



Universidad  
Nacional  
de Loja

## Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no  
Renovables

Carrera de Ingeniería en Sistemas


Módulo para el seguimiento y control de becas de la Unidad de  
Bienestar Universitario

Trabajo de Titulación previa  
a la obtención del título de  
Ingeniero en Sistemas

### AUTOR:

 Jackson Andrews Guzmán Tituana

### DIRECTOR:

 Ing. Pablo Fernando Ordoñez Ordoñez, Mg.Sc.

Loja - Ecuador

2023

## Certificación

Loja, 11 de septiembre de 2022

Ing. Pablo Fernando Ordoñez Ordoñez Mg. Sc.

**DIRECTOR DEL TRABAJO DE TITULACIÓN**

### **CERTIFICO:**

Que he revisado y orientado todo el proceso de la elaboración del Trabajo de Titulación denominado: **Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**, previa a la obtención del título de **Ingeniero en Sistemas** de autoría del estudiante: **Jackson Andrews Guzmán Tituana** con cédula de identidad Nro. **1105740979**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.



Ing. Pablo Ordoñez Ordoñez, Mg. Sc

**DIRECTOR DEL TRABAJO DE TITULACIÓN**

## **Autoría**

Yo, **Jackson Andrews Guzmán Tituana**, declaro ser autor del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mí Trabajo de Titulación en el Repositorio Digital Institucional – Biblioteca Virtual.

**Firma:**



**Cédula de Identidad:** 1105740979

**Fecha:** 24 de febrero de 2023

**Correo electrónico:** jackson.guzman@unl.edu.ec

**Teléfono:** +593 963 922 947

**Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo del Trabajo de Titulación.**

Yo, **Jackson Andrews Guzmán Tituana**, declaro ser autor del Trabajo de Titulación denominado: **Módulo para el seguimiento y control de Becas de la Unidad de Bienestar Universitario** como requisito para optar el título de **Ingeniero en Sistemas**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los veintitrés días del mes de febrero del dos mil veintitrés.

**Firma:**



**Autor:** Jackson Andrews Guzmán Tituana

**Cédula:** 1105740979

**Dirección:** Av. Paltas y Rusia, Loja - Ecuador

**Correo electrónico:** [jackson.guzman@UNL.edu.ec](mailto:jackson.guzman@UNL.edu.ec)

**Teléfono:** (+593) 963 922 947

**DATOS COMPLEMENTARIOS:**

**Director del Trabajo de Titulación:** Ing. Pablo Fernando Ordoñez Ordoñez, Mg. Sc.



## **Dedicatoria**

*Dedico este trabajo a mis padres y hermanos por ser el núcleo principal en mi formación profesional al depositar su confianza en que voy a alcanzar grandes cosas y llenar de orgullo a la familia, lo acepto con toda humildad y gratitud demostrando con este trabajo que he logrado alcanzar un peldaño más hacia la construcción de mis sueños y metas.*

**Jackson Guzmán**

## Agradecimientos

Quiero expresar mi gratitud primeramente a mis padres **Geovany Guzmán y Dilma Tituana** por ser el pilar fundamental en mi formación académica, por su esfuerzo, dedicación y constancia en hacer de mi una persona de bien con sus valiosos consejos y el apoyo incondicional, demostrándome que siempre se puede salir adelante sin importar los retos que la vida presente.

A mi hermano **Patricio Guzmán** con gran admiración, por ser el mentor y motivador para alcanzar cosas cada vez más grandes; con su ejemplo y valiosos consejos que me han incentivado a creer en mi, confiar en mis sueños y seguir siempre adelante con determinación demostrando que se puede llegar a ser grande manteniéndose humilde.

A mis demás hermanos y familiares que de muchas maneras han contribuido en formar mi carácter y moldear una persona sencilla, honesta y carismática, gracias a su demostración de afecto y consideración.

Agradezco también a todos aquellos docentes que han aportado significativamente en mi formación profesional, para que pueda contribuir a la sociedad de manera positiva aportando con el conocimiento adquirido y con ese espíritu curioso que me inculcaron de investigar y aprender siempre más, para entregar mejores resultados.

Finalmente agradezco de manera especial a los ingenieros **Máximo Álvarez y Pablo Fernando Ordoñez** por su apoyo incondicional para el desarrollo de mi trabajo de investigación.

**Jackson Guzmán**

## Índice de Contenidos

<b>Portada</b> .....	<b>i</b>
<b>Certificación</b> .....	<b>ii</b>
<b>Autoría</b> .....	<b>iii</b>
<b>Carta de autorización</b> .....	<b>iv</b>
<b>Dedicatoria</b> .....	<b>v</b>
<b>Agradecimientos</b> .....	<b>vi</b>
<b>Índice de Contenidos</b> .....	<b>vii</b>
Índice de Tablas .....	x
Índice de Figuras .....	xii
Índice de Anexos.....	xvii
Índice de Códigos.....	xix
<b>1. Título</b> .....	<b>1</b>
<b>2. Resumen</b> .....	<b>2</b>
2.1    Abstract .....	3
<b>3. Introducción</b> .....	<b>4</b>
<b>4. Marco Teórico</b> .....	<b>6</b>
4.1    Antecedentes de la Unidad de Bienestar Universitario.....	6
4.1.1    Generalidades .....	7
4.1.2    Misión.....	7
4.1.3    Visión.....	7
4.1.4    Historia .....	8
4.1.5    Servicios prestados.....	8

4.2	Modelo y Notación de Procesos de Negocio.....	9
4.2.1	Proceso de Negocio .....	9
4.2.2	Gestión de Procesos de Negocio (BPM).....	10
4.2.3	Modelo y Notación de Procesos de Negocio (BPMN) .....	10
4.2.4	Usos de BPMN .....	10
4.2.5	Elementos BPMN.....	10
4.3	Estándar IEEE 830 1998.....	13
4.3.1	Lenguaje Unificado de Modelado (UML) .....	14
4.3.2	Historias de Usuario .....	15
4.4	Metodología XP.....	16
4.4.1	Ciclo de Vida XP .....	16
4.4.2	Exploración.....	16
4.4.3	Planificación .....	16
4.4.4	Iteraciones por entregas.....	17
4.4.5	Producción.....	17
4.4.6	Mantenimiento.....	17
4.4.7	Muerte .....	17
4.5	Marco de Trabajo Scrum.....	18
4.5.1	Teoría de Scrum.....	19
4.5.2	Valores Scrum.....	19
4.5.3	Scrum Team.....	19
4.5.4	Developers .....	20
4.5.5	Product Owner .....	20

4.5.6	Scrum Master .....	20
4.5.7	Eventos Scrum .....	20
4.5.8	Sprint .....	21
4.5.9	Sprint Planning .....	21
4.5.10	Daily Scrum.....	21
4.5.11	Sprint Review.....	21
4.5.12	Scrum Retrospective.....	22
4.5.13	Product Backlog.....	22
4.5.14	Sprint Backlog.....	22
4.5.15	Increment.....	22
4.6	Tecnologías de Desarrollo Web .....	22
4.6.1	Python .....	22
4.6.2	Django .....	23
4.6.3	PostgreSQL .....	23
4.7	Pruebas de Software.....	23
4.7.1	Tipos de Pruebas .....	23
4.8	Trabajos Relacionados .....	25
<b>5.</b>	<b>Metodología.....</b>	<b>27</b>
5.1	Contexto .....	27
5.2	Procedimiento .....	27
5.3	Recursos.....	28
5.4	Participantes .....	29
<b>6.</b>	<b>Resultados .....</b>	<b>31</b>

6.1	Objetivo 1: Analizar los procesos que intervienen en la Gestión de Becas mediante el uso de BPMN .....	31
6.2	Objetivo 2: Desarrollar el módulo para el proceso de seguimiento y control de becas, utilizando el framework Django y la metodología XP.....	32
6.2.1	Especificación de los Requisitos de Software para el proceso de seguimiento y control de becas mediante el uso del estándar IEEE 830 .....	33
6.2.2	Diseño del Diagrama de Clases para el proceso de seguimiento y control de becas con el uso de UML.....	36
6.2.3	Historias de Usuario .....	38
6.2.4	Codificación de los procesos para el seguimiento y control de Becas .....	39
6.2.5	Estructura de la aplicación.....	41
6.3	Objetivo 3: Validar la implementación del módulo para el proceso de seguimiento y control de becas en un entorno de pruebas .....	47
6.3.1	Ejecución del Plan de Pruebas en un Entorno Simulado.....	47
6.3.2	Elaboración de la documentación necesaria para la implantación y uso del módulo desarrollado.....	63
6.3.3	Certificación de Control de Calidad emitido por la DTI.....	65
<b>7.</b>	<b>Discusión.....</b>	<b>67</b>
<b>8.</b>	<b>Conclusiones .....</b>	<b>71</b>
<b>9.</b>	<b>Recomendaciones .....</b>	<b>72</b>
9.1	Trabajos futuros.....	72
<b>10.</b>	<b>Bibliografía.....</b>	<b>74</b>
<b>11.</b>	<b>Anexos .....</b>	<b>76</b>

## Índice de Tablas:

<b>Tabla 1.</b> Elementos Básicos para el modelado usados en la notación BPMN. ....	11
<b>Tabla 2.</b> Resumen de los trabajos relacionados.....	25
<b>Tabla 3.</b> Resumen de los Requisitos Funcionales para el desarrollo del módulo informático. ....	35
<b>Tabla 4.</b> Historia de Usuario para notificar los resultados de la postulación a tesorería y a los postulantes.....	38
<b>Tabla 5.</b> Historia de Usuario para registrar el pago de Becas. ....	39
<b>Tabla 6.</b> Métrica recolectada durante la ejecución de las pruebas de carga y estabilidad. ....	49
<b>Tabla 7.</b> Reporte resumen con los tiempos de respuesta del grupo de hilos de consultas.....	50
<b>Tabla 8.</b> Métrica recolectada durante la ejecución de las pruebas de seguridad. ....	53
<b>Tabla 9.</b> Métrica recolectada durante la ejecución de las pruebas unitarias.....	54
<b>Tabla 10.</b> Resultados de la evaluación del MGB en base a los Requisitos Funcionales capturados durante la fase de análisis. ....	60

## Índice de Figuras:

<b>Figura 1.</b> Servicios prestados por la Unidad de Bienestar Universitario - UNL.....	9
<b>Figura 2.</b> Ciclo de vida XP (fuente Calabria, Luis [7]).....	18
<b>Figura 3.</b> Modelado General de Procesos que intervienen en la Gestión de Becas. ....	32
<b>Figura 4.</b> Diagrama de Casos de Uso General para el desarrollo del módulo informático MGB. .	34
<b>Figura 5.</b> Diagrama de Clases del MGB. ....	37
<b>Figura 6.</b> Diagrama de Despliegue. ....	40
<b>Figura 7.</b> Diagrama de Componentes donde se muestra los módulos del SIAAF con los que interactúa el MGB y la conexión con el MGT.....	41
<b>Figura 8.</b> Diagrama de Componentes del MGB.....	41
<b>Figura 9.</b> Estructura de la aplicación y api del MGB dentro del SIAAF. ....	42
<b>Figura 10.</b> Etapa de Revisión - MGB.....	46
<b>Figura 11.</b> Capturas de Pantalla a la Tabla que Certifica las Pruebas de Casos de Uso en base a la ERS del MGB (fuente DTI).....	47
<b>Figura 12.</b> Configuración del Plan de Pruebas con Apache JMeter 5.5.....	48
<b>Figura 13.</b> Ejecución de Pruebas de Carga y Estabilidad con JMeter. ....	49
<b>Figura 14.</b> Gráfica del tiempo de respuesta de las peticiones del hilo de consultas múltiples. ...	51
<b>Figura 15.</b> Configuración del grupo de hilos para las Pruebas de Estabilidad con Apache JMeter 5.5 .....	51
<b>Figura 16.</b> Gráfica del tiempo de respuesta para el grupo de hilos de las Pruebas de Estabilidad con Apache JMeter.....	52
<b>Figura 17.</b> Respuesta de redireccionamiento de la vista listar Estadísticas Becas al intentar acceder sin token de sesión. ....	53
<b>Figura 18.</b> Vista que muestra el acceso no autorizado cuando un usuario intenta acceder a un apartado al que no tiene permisos.....	54



<b>Figura 19.</b> Estructura del código de Pruebas Unitarias para el Módulo de Gestión de Becas.....	55
<b>Figura 20.</b> Respuesta por consola de la ejecución de las Pruebas Unitarias al método para obtener meses de acuerdo a un periodo académico dado. ....	55
<b>Figura 21.</b> Resultados del análisis de código por SonarQube para el MGB.....	56
<b>Figura 22.</b> Comparativa de cambios antes y después de aplicada la refactorización a la función de Finalizar Etapas para reducir su complejidad cognitiva. ....	57
<b>Figura 23.</b> Calificación A (mejor puntuación) en la mantenibilidad del código para el MGB. ....	58
<b>Figura 24.</b> Calificación A (mejor puntuación) en la seguridad del código para el MGB.....	58
<b>Figura 25.</b> Gráfico que muestra la percepción que tiene el usuario Administrador de Becas con respecto a la reducción de tiempo con el uso del MGB. ....	61
<b>Figura 26.</b> Gráfico que muestra la percepción que tiene el usuario de Contabilidad Financiera con respecto a la reducción de tiempo con el uso del MGB. ....	62
<b>Figura 27.</b> Gráfico que muestra la percepción que tiene el usuario de Tesorería Financiera con respecto a la reducción de tiempo con el uso del MGB. ....	62
<b>Figura 28.</b> Gráfico que muestra la calificación proporcionada por los Usuarios con respecto al uso del MGB.....	63
<b>Figura 29.</b> Portada del Manual de Usuario para el uso del MGB. ....	64
<b>Figura 30.</b> Portada del Manual Técnico para la configuración del MGB. ....	65
<b>Figura 31.</b> Portada del informe de certificación emitido por DTI.....	66
<b>Figura 32.</b> Dialogo que muestra que los diagramas BPMN desarrollados fueron validados exitosamente por Bizagi Studio. ....	78
<b>Figura 33.</b> Modelado General de Procesos que intervienen en la Gestión de Becas. ....	79
<b>Figura 34.</b> Diagrama BPMN - subprocesos para la Etapa de Postulación de Becas.....	80
<b>Figura 35.</b> Diagrama BPMN - subprocesos para la Etapa de Revisión de Becas.....	81
<b>Figura 36.</b> Diagrama BPMN - subproceso para la Etapa de Selección de Becas.....	81

<b>Figura 37.</b> Diagrama BPMN - subprocesos en la Etapa de Notificación de Becas. ....	82
<b>Figura 38.</b> Diagrama BPMN - subprocesos para Adjudicar Becas en la Etapa de Adjudicación....	83
<b>Figura 39.</b> Diagrama BPMN - subprocesos para Gestionar Becas en la Etapa de Adjudicación. ..	84
<b>Figura 40.</b> Diagrama BPMN - subprocesos para Concesionar Becas. ....	85
<b>Figura 41.</b> Caso de uso Revisar Becas. ....	86
<b>Figura 42.</b> Caso de uso Seleccionar Becas. ....	87
<b>Figura 43.</b> Caso de uso Adjudicar Becas .....	87
<b>Figura 44.</b> Caso de uso Concesionar Becas. ....	88
<b>Figura 45.</b> Casos de uso general para el Módulo de Gestión de Becas (MGB).....	89
<b>Figura 46.</b> Mapa de Impacto que define el alcance del proyecto. ....	91
<b>Figura 47.</b> Versionado de las Historias de Usuario para el desarrollo del módulo informático. ..	92
<b>Figura 48.</b> Diagrama de componentes del MGB .....	93
<b>Figura 49.</b> Vistas del MGB.....	94
<b>Figura 50.</b> Modelos del MGB .....	94
<b>Figura 51.</b> Plantillas del MGB.....	94
<b>Figura 52.</b> Bloques de código reutilizables del SIAAF.....	95
<b>Figura 53.</b> Tareas planificadas para el Sprint 1 en Jira. ....	96
<b>Figura 54.</b> Vista en el MGT que permite sincronizar las postulaciones de cada Periodo Académico hacia el MGB. ....	97
<b>Figura 55.</b> Ventana modal en la Etapa de Revisión que permite actualizar el estado de los Postulantes seleccionados.....	99
<b>Figura 56.</b> Input que permite asignar una calificación a la ficha socio-económica del Postulante seleccionado.....	100
<b>Figura 57.</b> Funcionalidad en la Etapa de Revisión que permite descargar la documentación de los	

Postulantes seleccionados.....	101
<b>Figura 58.</b> Funcionalidad en la Etapa de Revisión que permite ver el trámite de los Postulantes seleccionados en el MGT.....	102
<b>Figura 59.</b> Funcionalidad que permite actualizar el estado de los Becados seleccionados en la Etapa de Adjudicación.....	104
<b>Figura 60.</b> Lista de Postulantes en la Etapa de Revisión. ....	107
<b>Figura 61.</b> Lista de Postulantes en la Etapa de Selección.....	107
<b>Figura 62.</b> Lista de Becados en la Etapa de Adjudicación. ....	107
<b>Figura 63.</b> Lista de Becados en la Etapa de Concesión.....	108
<b>Figura 64.</b> Funcionalidad en la Etapa de Adjudicación que le permite al usuario de tesorería guardar la confirmación de registro en el ESIGEF.....	109
<b>Figura 65.</b> Tareas planificadas para el Sprint 2 en Jira. ....	110
<b>Figura 66.</b> Funcionalidad en la Etapa de Revisión que permite finalizar la Etapa para el Periodo Académico elegido. ....	111
<b>Figura 67.</b> Funcionalidad que permite generar reportes de acuerdo a los filtros aplicados con la información de cada Etapa.....	112
<b>Figura 68.</b> Sección de Configuraciones que le permite al Administrador de Becas actualizar los valores de pago para cada tipo de Beca. ....	114
<b>Figura 69.</b> Notificación recibida por un estudiante con los resultados de su postulación. ....	115
<b>Figura 70.</b> Template que permite importar los registros de Devengado o Pago de las Becas. ..	118
<b>Figura 71.</b> Ventana modal que permite actualizar manualmente el estado de Concesión de las Becas. ....	118
<b>Figura 72.</b> Tareas planificadas para el Sprint 3 en Jira. ....	119
<b>Figura 73.</b> Sección de Configuraciones para administrar los mensajes Frecuentes que se presentan en cada una de las Etapas. ....	121

**Figura 74.** Template que muestra estadísticas generales de Postulantes y Becados. .... 123

**Figura 75.** Captura de pantalla de la reunión de acercamiento con DTI. .... 143

## Índice de Anexos:

<b>Anexo 1.</b> Desarrollo del Trabajo de Titulación .....	76
<b>Anexo 2.</b> Análisis del Reglamento de Becas .....	126
<b>Anexo 3.</b> Especificación de Requisitos de Software en base a la norma del IEEE 830 para el MGB .....	127
<b>Anexo 4.</b> Manual de Usuario para el uso del MGB.....	128
<b>Anexo 5.</b> Manual Técnico de Requisitos Básicos para la configuración del MGB.....	129
<b>Anexo 6.</b> Plan de Pruebas para el MGB.....	130
<b>Anexo 7.</b> Ejecución del Plan de Pruebas para el MGB.....	131
<b>Anexo 8.</b> Certificación de Control de Calidad para el MGB .....	132
<b>Anexo 9.</b> Comparativa entre las Bitácoras en curso y finalizada para las Pruebas de Control de Calidad aplicadas por DTI en el MGB .....	133
<b>Anexo 10.</b> Aplicación de la técnica del User Story Mapping en el desarrollo ágil .....	134
<b>Anexo 11.</b> Modelo de Base de Datos generado usando el ORM de Django a partir del Diagrama de Clases para el MGB .....	135
<b>Anexo 12.</b> Análisis de Datos obtenidos a partir de las encuestas aplicadas a cada uno de los usuarios del MGB.....	136
<b>Anexo 13.</b> Acta de reunión para la validación de requisitos de software con DTI.....	137
<b>Anexo 14.</b> Acta de Reunión para la Validación de Requisitos con el representante de la UBU en calidad de cliente .....	138
<b>Anexo 15.</b> Acta de reunión con el equipo de desarrollo de la DTI para la validación del Diagrama de Clases a usarse en el MGB.....	139
<b>Anexo 16.</b> Certificado proporcionado por la DTI que expresa la necesidad de automatizar los procesos de becas.....	140
<b>Anexo 17.</b> Certificado que avala la realización de entrevistas para la captura de requisitos de	

software.....	141
<b>Anexo 18.</b> Certificado proporcionado por la DTI para el uso de Scrum en el proyecto.....	142
<b>Anexo 19.</b> Entrevista con miembros de la DTI para conocer el estado actual del proyecto...	143
<b>Anexo 20.</b> Certificado de traducción de resumen.....	144

## Índice de Códigos:

<b>Código 1.</b> Modelo Beca Periodo Académico Etapa definido en el archivo models.py de la app Becas.....	43
<b>Código 2.</b> Vista Etapa Revisión definida en el archivo Etapas.py de la app Becas.....	43
<b>Código 3.</b> Vista Etapa Revisión Paginador definida en el archivo etapas.py de la app Becas.....	44
<b>Código 4.</b> Template Etapa Revisión definido en el archivo Etaparevision.html de la app Becas..	45
<b>Código 5.</b> Fragmento de Código del endpoint para registrar postulantes .....	96
<b>Código 6.</b> Fragmento de Código del endpoint para actualizar estado de postulantes y becados	97
<b>Código 7.</b> Fragmento de Código de la función Ajax para actualizar estado de postulantes y becados asincrónamente .....	98
<b>Código 8.</b> Fragmento de Código del endpoint que permite actualizar la ficha socio-económica de los Postulantes en la Etapa de Revisión .....	99
<b>Código 9.</b> Fragmento de Código de la función Ajax que permite actualizar la ficha socio-económica de los postulantes en la Etapa de Revisión .....	100
<b>Código 10.</b> Fragmento de Código de la función backend que permite descargar los documentos del postulante seleccionado .....	101
<b>Código 11.</b> Fragmento de Código de la función que permite descargar los documentos del postulante seleccionado .....	101
<b>Código 12.</b> Fragmento de Código de la función ajax que permite abrir el trámite correspondiente al postulante seleccionado .....	102
<b>Código 13.</b> Fragmento de Código del endpoint para actualizar estado de becados .....	103
<b>Código 14.</b> Fragmento de Código de la función ajax que permite cargar datos de Postulantes en la Etapa de Revisión.....	104
<b>Código 15.</b> Fragmento de Código de la vista para cargar información de Postulantes en la Etapa de Revisión.....	105

<b>Código 16.</b> Fragmento de Código HTML con los filtros de las Etapas de Revisión, Selección y Adjudicación .....	106
<b>Código 17.</b> Fragmento de Código del endpoint para guardar la confirmación de registro del becado en el ESIGEF .....	108
<b>Código 18.</b> Fragmento de Código de la función ajax que permite guardar la confirmación de registro en el ESIGEF de los Becados.....	109
<b>Código 19.</b> Fragmento de Código de la Vista que permite finalizar las Etapas de Revisión y Selección.....	110
<b>Código 20.</b> Fragmento de Código de la Vista que permite generar Reportes para todas las Etapas .....	112
<b>Código 21.</b> Fragmento de Código de la Vista que permite actualizar los montos de pago de cada tipo de Beca .....	113
<b>Código 22.</b> Fragmento de Código de la función de la capa de aplicación que se encarga de guardar la notificación de los Estudiantes y Tesorería .....	114
<b>Código 23.</b> Fragmento de Código de la función en la capa de aplicación que se encarga de procesar el archivo a importar.....	116
<b>Código 24.</b> Fragmento de Código de la función en la capa de aplicación que permite descargar un reporte con los registros no procesados durante la importación .....	116
<b>Código 25.</b> Fragmento de Código del endpoint que permite actualizar manualmente el estado de Concesión de una Beca .....	117
<b>Código 26.</b> Fragmento de Código de la vista paginadora que lista los Mensajes Frecuentes ....	119
<b>Código 27.</b> Fragmento de Código de la Vista que permite crear, actualizar y eliminar Mensajes Frecuentes .....	119
<b>Código 28.</b> Fragmento de Código de la Vista que permite obtener estadísticas generales de Postulantes y Becados .....	122



## **1. Título**

Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario

## 2. Resumen

La Universidad Nacional de Loja (UNL) brinda la oportunidad a su comunidad estudiantil de acceder a becas económicas y sociales con el fin de ofrecer igualdad en las condiciones de estudio y reconocer el esfuerzo y participación de los estudiantes en la institución. Sin embargo, la gestión de becas actualmente se realiza a través de reportes y matrices de datos Excel, lo que resulta en información dispersa y desactualizada entre la Sección de Becas de la Unidad de Bienestar Universitario (UBU) y el Departamento Financiero, con sus respectivas subdirecciones de Contabilidad y Tesorería.

El objetivo principal de este Trabajo de Titulación (TT) es mejorar el proceso de seguimiento y control de becas para la UBU mediante la implementación de un módulo de software que permita la interacción simultánea de todas las entidades involucradas y evite la asimetría de información, dando contestación a la pregunta de investigación: ¿De qué manera se podría mejorar el proceso de seguimiento y control de becas para la UBU con el uso de las TIC?

Para lograr este objetivo, se dividió el trabajo en 3 fases: en la primera se modelaron los procesos mediante diagramas BPMN (del inglés, Business Process Model Notation), en la segunda se codificaron los procesos con la metodología XP y el framework Django, y en la tercera se implementó el módulo en el ambiente de aseguramiento de calidad del Sistema de Información Académico Administrativo Financiero (SIAAF) y se llevó a cabo una encuesta de satisfacción a los usuarios.

El aporte más significativo de este TT es la automatización de los procesos en la gestión de becas, lo que permitirá una gestión más eficiente y segura. Además, el módulo desarrollado garantiza la robustez, seguridad, mantenibilidad, escalabilidad y cumplimiento de los requisitos especificados.

En conclusión, la implementación de este módulo de software para la gestión de becas en la UNL mejorará la interacción y flujo de información entre las entidades involucradas, garantizando una gestión más eficiente y segura.

**Palabras claves:** desarrollo de software, seguimiento y control de becas, Unidad de Bienestar Universitario

## 2.1 Abstract

The Loja National University (UNL) offers to its student community the opportunity to access economic and social scholarships in order to offer equality in study conditions and recognize the student's effort and participation in the institution. However, scholarship management is currently carried out through Excel reports and data matrices, which results in dispersed and outdated information between the Scholarship Section of the University Welfare Unit (UBU) and the Financial Department, with their respective Accounting and Treasury departments.

The Degree Work (TT) main objective is to improve the monitoring and control process of scholarships for the UBU through the software module implementation that allows the simultaneous interaction of all the entities involved and avoids information asymmetry, giving answers to the research questions: How could the monitoring and control process of scholarships for the UBU be improved using the ICT?

To achieve this objective, the procedure was divided into 3 phases: in the first, the processes were modeled using BPMN diagrams (Business Process Model Notation), in the second, the processes were codified with the XP methodology and the Django framework, and in the third, the module was implemented in the quality assurance environment of the Financial Administrative Academic Information System (SIAAF) and a user satisfaction survey was conducted.

The most significant contribution of TT is the scholarship management processes automation, which will allow for more efficient and secure management. In addition, the developed module guarantees robustness, security, maintainability, scalability, and compliance with the specified requirements.

In conclusion, the software module implementation for the management of scholarships at UNL will improve the interaction and flow of the information between the entities involved, guaranteeing more efficient and secure management.

**Keywords:** software development, scholarships' monitoring and control, University Welfare Unit.

### 3. Introducción

El Estado ecuatoriano faculta a las universidades a crear fuentes complementarias de ingresos en beneficio de la academia, la inversión en investigación y el otorgamiento de becas a los estudiantes, para la subvención parcial de gastos en la culminación de sus estudios de acuerdo a los criterios de vulnerabilidad establecidos por la institución de educación superior [1].

La Universidad Nacional de Loja (UNL) a través de la Unidad de Bienestar Universitario (UBU), en cumplimiento de los objetivos que plantea la institución, y de conformidad a la ley de educación superior y reglamentos universitarios, concede un programa de becas y servicios sociales a los estudiantes durante sus procesos educativos [1] [2].

Actualmente, la UBU gestiona las becas a través del Módulo de Gestión de Trámites (MGT), el módulo permite al estudiante postulante a la beca, enviar la documentación requerida para que el encargado de la Sección de Becas descargue los documentos de postulación durante la etapa de revisión. Posteriormente, en la etapa de selección se reúne la comisión de becas en sesión privada para analizar la situación de vulnerabilidad de los estudiantes y seleccionar aquellos que cumplen con los requisitos establecidos en el Reglamento de Becas. Los postulantes seleccionados de cada periodo académico son registrados en una matriz de datos Excel para el seguimiento y control de sus becas. Periódicamente el responsable de la Sección de Becas envía reportes al Departamento Financiero para el registro y concesión de las Becas. Esto causa que la información se encuentre dispersa entre 2 entidades con información desactualizada. El MGT no cuenta con las funcionalidades necesarias para automatizar el seguimiento y control de Becas lo que su vez genera dificultades en el procesamiento de los pagos a los becados y principalmente la asimetría en la información dificulta el análisis para proporcionar estadísticas generales sobre el otorgamiento de las becas.

Con este antecedente, para dar solución a la deficiencia en la gestión de becas, el presente trabajo titulación propuso como objetivo general: “Implementar el módulo para el proceso de seguimiento y control de becas en la Unidad de Bienestar Universitario”. Para alcanzar el objetivo general fue necesario el cumplimiento de tres objetivos específicos. Empezando por “Analizar los procesos que intervienen en la gestión de Becas mediante el uso de la notación BPMN” para

comprender la lógica de negocio presente y con ello proceder a “Desarrollar el módulo para el proceso de seguimiento y control de becas utilizando el framework Django y la metodología XP”, al ser un módulo del Sistema de Información Académico Administrativo Financiero (SIAAF), se utilizaron las tecnologías establecidas por este sistema; y la metodología XP considerando el ambiente de desarrollo ágil necesario para el cumplimiento de este TT durante el periodo comprendido entre octubre de 2021 a septiembre de 2022. Finalmente, se implementó el Módulo de Gestión de Becas (MGB) en el ambiente de aseguramiento para la calidad del SIAAF, con el fin de “Validar la implementación del módulo para el proceso de seguimiento y control de becas en un entorno de pruebas” que garanticen que el software se encuentra listo para el paso a producción.

A continuación, se describe brevemente la organización del trabajo de titulación (TT).

En el Marco Teórico, se encuentran los conceptos fundamentales para la comprensión del documento y está compuesto por las secciones antecedentes, herramientas de modelado, metodología de desarrollo, marco de trabajo utilizado, tecnologías de desarrollo, pruebas de software y trabajos relacionados.

Seguidamente en el capítulo 6 se muestran los resultados alcanzados en el TT empezando por el análisis de los procesos que intervienen en la gestión de becas mediante el modelado BPMN para la consolidación de los mismos mediante la Especificación de Requisitos de Software (ERS), el diagrama de clases, el diagrama de componentes del módulo y su estructura, demostrando la codificación de uno de sus principales componentes. Para finalizar este capítulo, se presentan 6 tipos de pruebas aplicadas para validar la implementación del módulo, así como la documentación para su uso.

En el capítulo 7 se discute acerca del TT y los resultados obtenidos en comparación con los otros trabajos investigados y el aporte adicional que entrega este TT. Posteriormente se enlistan las conclusiones y recomendaciones que surgieron a partir del trabajo realizado.

Por último, se encuentra el capítulo 11 de Anexos que brinda la sustentación necesaria de los resultados obtenidos y proporciona material adicional y recursos generados en el proceso de investigación para garantizar la correcta culminación del TT.

## **4. Marco Teórico**

En esta sección se enmarca toda la teoría necesaria para el desarrollo del presente TT; comenzando por describir en la subsección 4.1, los antecedentes de la UBU que permiten comprender la importancia y relevancia del presente TT, así como el lugar en donde se llevó a cabo. Seguidamente como base teórica para el desarrollo del primer objetivo, en la subsección 4.2 se habla sobre el Modelo y Notación de Procesos de Negocio, del inglés BPMN con la cual se realizó el modelado de los procesos que intervienen en la Gestión de Becas; para hacer una distinción con aquellos procesos a los que se dio solución. Luego, en la subsección 4.3 se describe en que consiste el Estándar IEEE 830 1998, con el fin de abstraer los requerimientos del módulo informático a partir del modelado BPMN.

Para dar cumplimiento al segundo objetivo, el desarrollo de software se llevó a cabo mediante la Metodología XP que se detalla en la subsección 4.4. Las actividades que se planificaron con la DTI se coordinaron bajo el Marco de Trabajo SCRUM; teoría que se presenta en la subsección 4.5 y las tecnologías que se utilizaron para el desarrollo del MGB de la UBU se muestra en la subsección 4.6.

Finalmente, para alcanzar el último objetivo, en la subsección 4.7 se investigó acerca de algunas pruebas que se usan para validar un software antes de su implementación. Por último, a modo comparativo en la subsección 4.8, se presenta un cuadro resumen con algunos trabajos relacionados con el presente TT.

### **4.1 Antecedentes de la Unidad de Bienestar Universitario**

En esta subsección se presenta las generalidades a considerar acerca de la UBU, así como su misión, visión, historia y servicios prestados. La UBU es una unidad que, bajo la dirección del Rector, posee autonomía funcional y de gestión, para contribuir al acatamiento de los objetivos institucionales, proporcionando ciertos beneficios amparados en la ley de educación superior y reglamentos universitarios a los estudiantes durante sus procesos educativos, siendo acogidos en los programas de becas y servicios sociales. La UBU está conformada por un conjunto de recursos materiales y humanos que se encarga de proporcionar una atención integral, de calidad y de fácil acceso a los estudiantes de la UNL. Al interior de la UBU se encuentra la Sección de Becas. La cuál

posee una gran cantidad de información que no se gestiona de una manera eficiente y automatizada, por lo que se necesita mejorar la gestión de la información para no perder la oportunidad de brindar una atención personalizada a los estudiantes.

#### 4.1.1 Generalidades

El objetivo de la UBU es proporcionar a los estudiantes un ambiente basado en el respeto, salvaguardando siempre la integridad de todos los miembros que integran el campus universitario [1, p. 4]. Es por ello que la UBU pone a disposición [1, p. 4] “servicios de salud, de formación humana, lúdico-deportivos, socio-culturales y de apoyo institucional” que permiten la formación, integración y el desarrollo de la comunidad universitaria.

La finalidad de la UBU es [1, p. 6]: “ser un ente propiciador del desarrollo institucional, facilitador y propiciador del bienestar de sus integrantes, que prioriza las políticas de servicios en el cuidado de la salud integral y de asistencia al desarrollo humano”.

Entonces, se afirma que la UBU se encarga de impulsar las mejoras educativas pertinentes a la vez que vela por la integridad de toda la comunidad unelina, ofreciendo servicios para el cuidado de su salud y el desarrollo personal.

Para alcanzar dicho objetivo en [1, p. 4] se indica que crearon un Plan Estratégico con vigencia entre los años 2019 a 2023 con una Misión y Visión renovada.

#### 4.1.2 Misión

“La Unidad de Bienestar Universitario, es un grupo humano que contribuye al desarrollo integral de la comunidad universitaria y de la sociedad en general, mediante acciones de prevención y satisfacción de sus necesidades, en un marco humanístico y de servicio” [1].

#### 4.1.3 Visión

“En el año 2023 la Unidad de Bienestar Universitario de la Universidad Nacional de Loja, oferta servicios de excelencia con personal especializado, capacitado y con espíritu de servicio, cuya gestión se basa en un modelo que es referente a nivel regional y nacional enfocado en la satisfacción de las necesidades y el mejoramiento de la calidad de vida de quienes integran la comunidad universitaria y de la sociedad en general” [1].

#### 4.1.4 Historia

La UBU nace en virtud de cubrir la falta de servicios universitarios inexistentes en su momento. De acuerdo con [1] la UBU “entra en funcionamiento el 29 de septiembre de 1976, con el objetivo fundamental de brindar servicios en área médico-dental; peluquería” en primera instancia, para posteriormente ampliar a los servicios de “farmacia, laboratorio clínico y Trabajo Social” [1, p. 5]. En 1989 se crea el proyecto denominado “Bienestar Estudiantil Universitario” conformado en su momento por el “Departamento de Becas y Bolsas de trabajo, teniendo como objetivo la planificación, coordinación, ejecución y evaluación de los servicios de asistencia y ayuda para los estudiantes.” [1, p. 5].

Durante los años 1999 a 2003 se incluye el “Proyecto de Desarrollo del Bienestar Universitario”; con la siguiente subdivisión: “Mejoramiento de los Servicios de Bienestar Estudiantil, Concesión de Becas y Ayudas Económicas” [1, p. 5].

Mientras que, en 2002, se activaron los siguientes servicios: “Psicopedagógico, Deberes y Derechos Estudiantiles, Servicio de Salud y Servicio de Becas” [1, p. 5].

#### 4.1.5 Servicios prestados

En el año 2002 se aprobó el Proyecto de Bienestar Universitario, por el H. Junta Universitaria con los siguientes servicios que se exponen en la Figura 1:





Figura 1. Servicios prestados por la Unidad de Bienestar Universitario - UNL<sup>1</sup>

## 4.2 Modelo y Notación de Procesos de Negocio

En esta sección se habla acerca del modelado y notación de procesos de negocio que este compuesto por un conjunto de diagramas que sirven para modelar, visualizar y documentar los procesos de negocio y sus interacciones entre los diferentes participantes del negocio. BPMN se fundamenta en el uso de diagramas de flujo y en el uso de actores, enfoques y actividades, los cuales en conjunto conforman un proceso de negocio. En este documento se presentan los principales elementos que lo conforman.

### 4.2.1 Proceso de Negocio

Según se detalla en [2] un proceso de negocio es un conjunto de actividades que están impulsadas por eventos y ejecutándose en una cierta secuencia, crean valor para el cliente. El agrupamiento de diversas actividades que se forman por medio de los eventos genera procesos de negocio con

<sup>1</sup> Imagen tomada de: [Bienestar Universitario | Universidad Nacional de Loja \(unl.edu.ec\)](http://Bienestar%20Universitario%20|%20Universidad%20Nacional%20de%20Loja%20(unl.edu.ec))

los cuales el cliente interactúa para conseguir el fin deseado.

#### 4.2.2 Gestión de Procesos de Negocio (BPM)

Los procesos de negocio son administrados de forma empírica por los negocios donde la atención se centra en mejorar la manera en que se hacen las cosas para el beneficio de las partes interesadas, entonces entra en juego la gestión de los mismo, por eso en [2, p. 25] se define a BPM como la “Disciplina para la Gestión de Procesos de Negocio y de mejora continua apoyada fuertemente por las Tecnologías de la Información”.

#### 4.2.3 Modelo y Notación de Procesos de Negocio (BPMN)

El Modelo y Notación de Procesos de Negocio (BPMN) se elaboró con el fin de modelar procesos y ese es su principal enfoque. Es una notación que sirve para expresar los procesos de negocio en un diagrama (BPD). Los Modelos de BPMN pueden enlazarse con otros modelos de una arquitectura empresarial. BPMN está basado en el uso de diagramas de flujo ya que estos diagramas permiten el documentar cómo los eventos y los datos fluyen entre los diferentes actores, enfoques y actividades. BPMN también está basado en el uso de actores, enfoques y actividades, ya que estos elementos describen cómo los procesos se llevan a cabo en los diferentes participantes del negocio. Está restringido a soportar sólo los conceptos de modelado que son aplicables a los Procesos de Negocio. Esto significa que otros tipos de modelado realizados por las organizaciones con fines empresariales están fuera del alcance de BPMN [3].

#### 4.2.4 Usos de BPMN

Uno de los usos de BPMN es comunicar distintos tipos de información a una audiencia diversa. BPMN está diseñado para cubrir muchos tipos de modelado y permite la creación de Procesos de Negocio de principio a fin. Los elementos estructurales de BPMN permiten al espectador ser capaz de diferenciar fácilmente entre las secciones de un diagrama BPMN [3].

#### 4.2.5 Elementos BPMN




Uno de los mecanismos que impulsó la creación del modelado a través de BPMN es que fuera sencillo de comprender por la mayoría de personas, pero que al mismo tiempo fuera capaz de manejar la complejidad existente en las organizaciones independientemente de su tamaño. Estos









dos requisitos indispensables y a la vez contradictorios entre sí, ha conducido a agrupar elementos con el fin de inducir a su utilización empezando con aquellos elementos más básicos y por ende esenciales. Los cinco elementos básicos son:

1. Objetos de flujo: conformados por eventos, actividades, puertas de enlace.
2. Datos: representados por objetos, entradas, salidas y almacenes.
3. Objetos de conexión: pueden ser flujos de secuencia, flujos de mensaje, asociaciones y asociaciones de datos.
4. Swimlanes: se encuentran las piscinas y los carriles.
5. Artefactos: están los agrupamientos y anotaciones de texto.

En la Tabla 1 se describe brevemente cada uno de estos elementos y se muestra su respectiva notación.

*Tabla 1. Elementos Básicos para el modelado usados en la notación BPMN.*

Elemento	Descripción	Notación
Evento	Los eventos se dan en el transcurso de un proceso o una coreografía, pueden afectar el flujo del modelo, suceden debido a una causa (desencadenante) y siempre tienen un impacto o resultado. Estos se pueden diferenciar en eventos circulares con centros abiertos. En cuanto a la clasificación se constatan tres tipos de Eventos, basados en el momento en que afectan al flujo: Inicio, Intermedio y Final [3].	
<b>Actividad</b>	Se utiliza el término actividad para referirse al trabajo que la empresa efectúa en un Proceso. Estas actividades pueden ser de forma atómicas o no atómicas (compuestas). Mientras que, los tipos de Actividades que conforman un Modelo de Proceso son: Subproceso y Tarea, que son rectángulos redondeados. Las actividades se utilizan tanto en los Procesos estándar como en las Coreografías [3].	
<b>Puerta de Enlace</b>	Un Gateway se utiliza para establecer la "ramificación, bifurcación, fusión y unión de trayectorias" [3].	


<b>Flujo de secuencia</b>	<b>de</b> El Flujo de Secuencia se utiliza para representar el orden de las actividades de un proceso [3].	
<b>Flujo mensaje</b>	<b>de</b> Utilizado para representar el flujo de Mensajes entre dos Participantes que reciben y envían [3].	
<b>Asociación</b>	Este elemento se usa para representar la unión de 2 elementos. Las puntas de flecha en la asociación indican la dirección del flujo (por ejemplo, datos), si están disponibles [3].	
<b>Piscina</b>	Un pool es una representación gráfica de un participante en una colaboración para separar un conjunto de actividades de otros pools, a menudo dentro del contexto de escenarios B2B. Un pool puede tener componentes internos, en forma de un proceso para ejecutar. O un pool podría ser una “caja negra, es decir, puede no tener componentes internos [3].	
<b>Carril</b>	Representa la división de una Piscina, para contener todo el proceso, ya sea de manera vertical u horizontal. Los carriles se utilizan para ordenar y clasificar actividades [3].	
<b>Objeto Datos</b>	<b>de</b> Los objetos de datos facilitan información sobre qué actividades deben realizarse y/o qué producen. El objeto de datos puede representar un solo objeto o un conjunto de objetos. Los datos de entrada y salida proveen la misma información para los procesos [3].	
<b>Mensaje</b>	Se utiliza para representar lo incluido en la comunicación entre dos colaboradores (según lo determinado por un PartnerRole o PartnerEntity) [3].	
<b>Agrupamiento</b>	Un grupo es un conjunto de elementos gráficos correspondientes a una misma categoría. En este tipo de grupo no hay afectación a los flujos de secuencia en el grupo. Los nombres de categoría aparecen en el diagrama como etiquetas de grupo. Las categorías se pueden utilizar	

---

con fines de documentación o análisis. La agrupación es una forma de mostrar visualmente una lista de objetos en un diagrama [3].

---

**Anotación de texto** Se trata de un mecanismo mediante el cual el modelador provee información textual adicional al lector sobre el diagrama BPMN [3].



Descriptive Text  
Here

---

*Nota. Esta tabla fue adaptada y traducido de [3, p. 29] acerca de Basic BPMN Modeling Elements.*

### 4.3 Estándar IEEE 830 1998

El estándar IEEE 830 permite la ERS por medio de un conjunto de recomendaciones que se usan para recopilar los requisitos de un software entregando como producto final la documentación de los acuerdos entre el cliente y el equipo de desarrollo de software [4].

Seguir el estándar IEEE implica seguir un formato establecido con 3 secciones principales con sus respectivas subsecciones que se detallan a continuación:

**1. Introducción:** debe proporcionar una vista general de la ERS. Incluir el objetivo, el alcance, las definiciones y acrónimos, las referencias, y la vista general de la ERS. Este compuesto por las siguientes subsecciones de:

- Propósito: se explica el objetivo que se pretende conseguir y a que publico está dirigido el software.
- Alcance: se nombra al software a desarrollar, se explica de manera general lo que hará y no hará el sistema
- Personal Involucrado: se describe a todas las personas involucradas en el desarrollo del sistema, con información de contacto.
- Definiciones, Acrónimos y Abreviaturas: se proporciona el significado de las siglas a utilizar y conceptos que deben ser explicados.
- Referencias: se cita todos aquellos documentos de orden superior que se requieren para la elaboración del software.

**2. Descripción General:** Se compone de las siguientes subsecciones:

- Perspectiva: consiste en indicar si el producto es independiente o forma parte de un sistema mayor.
- Objetivos del Producto: se considera aquel objetivo general y específicos que se pretenden alcanzar con el sistema.
- Características de los actores: se presenta una descripción de las actividades que desempeñan los actores del sistema dentro del mismo.
- Restricciones: se describen todas aquellas limitantes a la hora de diseñar y desarrollar el sistema.
- Suposiciones y Dependencias: se detallan aquellos factores que al cambiar pueden afectar a los requisitos.

**3. Requisitos Específicos:** contiene una lista detallada y completa de los requisitos que debe cumplir el sistema a desarrollar. Es la sección más extensa e importante del documento y se componen de las siguientes subsecciones:

- Requisitos comunes de Interfaces: se presentan las interfaces comunes de usuario que incluirá el sistema, tal como botones, menús, headers, footers, etc.
- Requisitos Funcionales: definición de acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados.
- Requisitos no Funcionales: Se muestran requisitos que aseguran la calidad del software de manera medible con calificadores cuantitativos. Está conformado por requisitos de rendimiento, seguridad, fiabilidad, disponibilidad, mantenibilidad y portabilidad.
- Otros Requisitos: aquí se detallan todos aquellos que no correspondan con las subsecciones anteriores.

#### 4.3.1 Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. UML tiene principalmente 4 objetivos: Visualizar, Especificar, Construir y Documentar [5]. Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.).
- Relaciones: relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.

#### 4.3.2 Historias de Usuario

Las historias de usuario son compatibles con marcos ágiles como Scrum. Las historias de usuario representan el corazón de los Sprints organizando el trabajo necesario a realizarse, ayudando a estimar y planificar los Sprints. “Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo proporcionará una función de software valor al cliente” tal como se resume en [6].

En base a la definición anterior se puede afirmar que las Historias de Usuario sirven para analizar el trabajo a desarrollar desde la perspectiva del usuario, utilizando un lenguaje informal para que el equipo de desarrollo comprenda lo que se espera recibir.

Entre los beneficios de usar historias de usuario están:

- Poseer la visión del usuario.
- Permiten la colaboración con el fin de lograr un mejor resultado.
- Impulsan soluciones creativas.
- Motivan a completar el reto para obtener una pequeña victoria.

Las historias de usuario se describen de acuerdo a la siguiente estructura:

*“Como <tipo usuario> quiero ... para ...”*

El “Como” viene acompañado del perfil para quien se desarrolla la funcionalidad, el equipo de desarrollo debe comprender cómo trabaja, cómo piensa y vive el problema esa persona. No se trata del rol que desempeña dicha persona. El “Quiere” no se refiere a las funciones que usará sino más bien a la intención que busca alcanzar dicha persona. Se debe describir independientemente del trabajo a realizarse para alcanzar el propósito. y finalmente el “Para” hace referencia al gran problema que debe resolver, y el beneficio que conlleva.

Para la transcripción de las historias de usuario se debe tomar en cuenta las siguientes

consideraciones.

- Describir tareas y subtareas.
- Perfiles de usuario.
- Pasos ordenados para lograr terminar la historia de usuario.
- Obtener feedback de los usuarios para captar sus problemas o necesidades.
- Tiempo, dado que las historias deben culminarse en un Sprint, de requerirse más tiempo se debe subdividir en historias más pequeñas para trabajarlas en diferentes Sprints.
- La historia se puede dar por finalizada solo si el usuario puede cumplir con el objetivo de la tarea.

#### **4.4 Metodología XP**

Esta metodología es diferente a los procesos convencionales de desarrollo de software, puesto que, el cambio es asumido como un proceso natural. El cliente realiza un pedido con los requerimientos que necesita, con la opción de sugerir cambios en cualquier momento esta es una de las ventajas más representativas de la Metodología XP [7].

##### **4.4.1 Ciclo de Vida XP**

El ciclo de vida XP está compuesto por las siguientes 6 fases.

##### **4.4.2 Exploración**

Durante la exploración los clientes se encargan de crear las historias de usuario con la descripción de las funciones que se verán proyectadas en el sistema posteriormente, permitiendo que los desarrolladores inicien la interacción con las diferentes herramientas que requiere el proyecto. En algunos casos, varios aspectos que se necesitan en la implementación de la arquitectura son puestos a prueba mediante el uso de un prototipo. En cuanto a la duración de esta fase debería estar entre unas semanas o algunos meses, según sea el caso en la adaptación de los programadores [7, p. 11].

##### **4.4.3 Planificación**

En la planificación se construye la primera versión de entregables y se establece el orden de cada historia, según su prioridad. El equipo de desarrollo estima el esfuerzo requerido para cada



historia acordando un cronograma. En promedio, la duración de la planificación es de dos días y la entrega del primer release ronda aproximadamente dos meses [7, p. 11].

#### 4.4.4 Iteraciones por entregas

Previamente a la entrega del primer release se realizan varias iteraciones. Se planifica dividiendo al calendario de forma que, la implementación de cada iteración lleve el tiempo de una a cuatro semanas. En la primera iteración se toma en cuenta los aspectos más relevantes de la arquitectura global para la implementación del sistema [7, p. 11]. Esto es posible gracias a la selección de las historias que contienen referencias a la construcción de la estructura de todo el sistema. Cabe recalcar que es decisión del cliente qué historias son tomadas en cuenta para cada iteración. Posteriormente al finalizar las iteraciones los clientes validan las historias y “al final de la última repetición el software está listo para ser puesto en producción” [7, p. 11].

#### 4.4.5 Producción

En esta etapa se realizan varias pruebas para avalar y garantizar el correcto funcionamiento del sistema. Pueden existir nuevos cambios que estén sujetos a discusión sobre la realización de su incorporación en la actual versión o en la siguiente entrega. En la fase de producción las iteraciones tienden a acortarse, pasando de tres a una semana, mientras que las sugerencias y propuestas que surjan se tomarán en cuenta en fases posteriores como la de mantenimiento [7, p. 11]. Posteriormente a la implementación del primer release, “el proyecto debe mantener el sistema en producción corriendo mientras se trabaja en las nuevas iteraciones” [7, p. 11].

#### 4.4.6 Mantenimiento

Es la fase más compleja para los programadores en cuanto al cumplimiento de los requerimientos extras, solicitados por los clientes. Una vez que el sistema se encuentra en producción la velocidad de desarrollo disminuye, puesto que existen muchos requerimientos a los que dar respuesta. Como consecuencia de esto varias veces es necesario cambiar la estructura del equipo de desarrollo e incorporar nuevos integrantes en la medida de lo posible [7, p. 11].

#### 4.4.7 Muerte

Constituye la última fase, donde el cliente no presenta ninguna historia a implementarse, por lo

que no se realizan más cambios en la arquitectura, el diseño o el código, pero es necesario efectuar la documentación correspondiente y satisfacer los requerimientos no funcionales como el rendimiento o la confiabilidad del software.

Por otra parte, esta fase puede aparecer también una vez que el sistema no muestra los resultados deseados o el costo de su desarrollo no es viable [7, p. 11].

A continuación, se puede observar mediante una gráfica como se correlacionan las distintas fases del ciclo de vida XP.

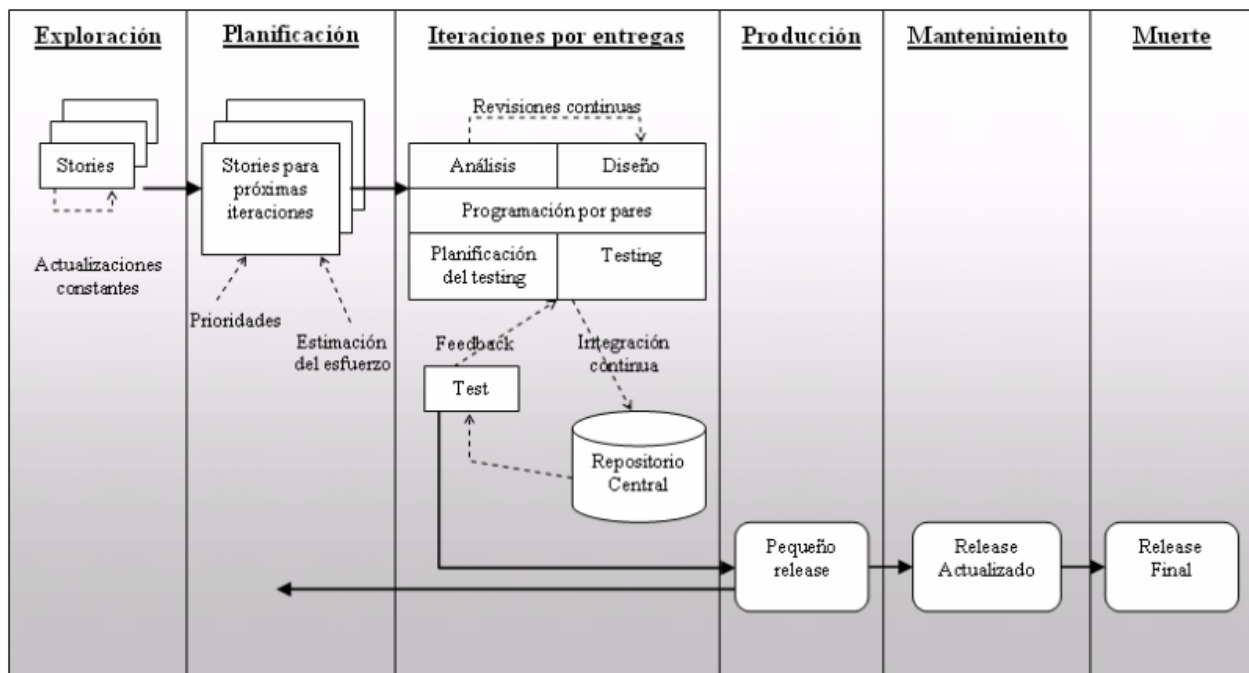


Figura 2. Ciclo de vida XP (fuente Calabria, Luis [7])

#### 4.5 Marco de Trabajo Scrum

En la edición 2020 K. Schwaber & J. Sutherland [8, p. 3] mencionan que “Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos”. Siendo un referente por el que las personas pueden abordar problemas y entregar productos de manera productiva y creativa. Este marco solo define la teoría necesaria para su implementación y permite que las personas lo adapten a su entorno y necesidades. Este marco de trabajo no establece instrucciones detalladas a seguirse, sino que guían el proceso de gestión [8, p. 3].

Revisando la guía de 2017 se menciona que Scrum no se trata de un proceso, método o técnica definitiva. En [9] se menciona que “engloba todo un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas”. De este modo se demuestra que aplicando diferentes técnicas de gestión al producto y trabajo se pueden ir mejorando continuamente. De acuerdo con [9, p. 4] Scrum se ha aplicado en “desarrollar software, hardware, software embebido, redes de funciones interactivas, vehículos autónomos, escuelas, gobiernos, también para gestionar la operación de organizaciones y casi todo lo que usamos en nuestra vida diaria, como individuo y como sociedad”.

#### 4.5.1 Teoría de Scrum

En [8] se menciona que “Scrum está basado en el empirismo y el pensamiento Lean”. El empirismo surge del conocimiento provisto por la experiencia en las decisiones tomadas a partir de la observación. Y por su parte el pensamiento Lean se centra en identificar y eliminar sistemáticamente los desperdicios de un proceso. Parafraseando acerca de su teoría, Scrum adopta un enfoque incremental e iterativo con el fin de optimizar tareas previsibles y controlar el riesgo. El trabajo con Scrum implica un grupo de personas dispuestas a compartir y adquirir habilidades para el desarrollo del trabajo.

#### 4.5.2 Valores Scrum

Para alcanzar el éxito en la implementación de Scrum en [8, p. 4] se recomienda practicar estos 5 valores fundamentales: “Compromiso, Foco, Franqueza, Respeto y Coraje”.

El equipo que incorpora Scrum debe comprometerse a alcanzar los objetivos planteados enfocándose en el trabajo por hacer siendo francos ante los desafíos que se presenten, teniendo el valor de afrontarlos con determinación [8, p. 4].

#### 4.5.3 Scrum Team

De acuerdo con Ken Schwaber & Jeff Sutherland [8] “El Scrum Team consta de un Scrum Master, un Product Owner y DevelopERS” En el equipo no existe una subdivisión, es decir, no está conformado por “sub-equipos ni jerarquías”. Los equipos presentan una gestión independiente, “lo que significa que deciden internamente quién hace qué, cuándo y cómo” [8, p. 5]; además habitualmente se utilizan no más de diez personas para completar un Sprint [8].

#### 4.5.4 Developers

Se refiere a las personas que se encargan de completar las Historias de Usuario que contiene un Sprint para obtener un Incremento del producto [8, p. 5].

Los developers o desarrolladores se encargan exclusivamente de:

- Crear el Sprint Backlog;
- Proporcionar entregables de calidad;
- Adecuarse diariamente al objetivo principal del Sprint; y,
- Tomar responsabilidad como profesionales.

#### 4.5.5 Product Owner

Es la persona encargada de dirigir el proyecto a nivel técnico en el equipo. Su principal responsabilidad es precautelar la realización del trabajo de los miembros del equipo están bajo su autoridad [8, p. 6]. Las tareas que realiza son:

- Completar y exponer el objetivo del sprint y producto.
- Gestionar la lista del Product Backlog.
- Garantizar que el equipo entienda el Product Backlog.

#### 4.5.6 Scrum Master

En la Guía de Scrum [8, p. 6], el término Scrum Master se define como el líder responsable de establecer Scrum. Apoya al equipo a entender “la teoría y la práctica de Scrum, tanto dentro del Scrum Team como de la organización” [8, p. 6]. Su responsabilidad radica principalmente en mejorar las prácticas de trabajo dentro del uso de Scrum [8, p. 6]. Algunas de las actividades del Scrum Master son:

- Dirigir de forma eficiente al equipo para lograr independencia en la gestión y multifunción.
- Liderar, enseñar e instruir en el uso de Scrum.

#### 4.5.7 Eventos Scrum

El propósito de la creación de los eventos Scrum es regular y reajustar el proceso que se lleva a cabo en el proyecto, éstos son una oportunidad para “inspeccionar y adaptar los

artefactos“ [8].

#### 4.5.8 Sprint

De acuerdo con [9] el Sprint representa la parte central de Scrum en el que en un mes o menos se entrega un incremento del producto, “utilizable y potencialmente desplegable” [9, p. 7]. En [9] se menciona que “es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Un Sprint siempre se inicia inmediatamente después de que ha finalizado el ciclo de Sprint anterior”.

#### 4.5.9 Sprint Planning

Para la ejecución de un Sprint Planning se establece el trabajo por medio de una reunión con el Scrum Team de no más de 8 horas. Durante el Sprint Planning se realiza asesoramiento, por lo que, se puede invitar a asistir a personas externas [9]. Los temas a tratar en la reunión, principalmente son: el por qué es valioso el sprint y cuál es el objetivo que se persigue, qué se puede hacer. Según Ken Schwaber & Jeff Sutherland [9] “los developer en conjunto con el product owner seleccionan elementos del Product Backlog para incluirlos en el Sprint actual y posteriormente refinarlos con el Scrum Team”. También se trata el cómo se realizará la planificación del trabajo necesario para crear un Increment, eventualmente esto se hace desmenuzando las actividades del product backlog diariamente o menos [9].

#### 4.5.10 Daily Scrum

Los Daily Scrum son reuniones de entre 15 a 30 minutos en la que los Developers se reúnen diariamente durante el transcurso del Sprint para elegir la estructura, centrando su atención en el progreso del trabajo hacia el objetivo de Sprint y continué la planificación para el siguiente día hábil [9].

#### 4.5.11 Sprint Review

En esta sesión, que tiene un marco de tiempo máximo de cuatro horas para el Sprint de un mes, su propósito es examinar los resultados del Sprint e identificar futuras medidas de adaptación. Los resultados del trabajo se presentan a los interesados clave para discutir sobre los avances y el objetivo a alcanzar con el producto [9].

#### 4.5.12 Scrum Retrospective

El equipo de Scrum analiza el último Sprint en términos de personas, “interacciones, procesos, herramientas y su definición de Terminado” [9]. Su propósito es planificar formas de aumentar la calidad y la eficiencia. Tiene un límite de hasta tres horas para un Sprint de un mes [9].

#### 4.5.13 Product Backlog

Se define al Product Backlog como una lista con las actividades para el desarrollo del producto, es decir, “son los requisitos conocidos y mejor entendidos al principio” [9]. Esta lista es dinámica e ira evolucionando conforme se desarrolla el producto.

Constantemente se realiza un refinamiento dividiendo en elementos más pequeños para añadir detalles, descripción, orden y tamaño.

#### 4.5.14 Sprint Backlog

Se llama así a la lista formada por elementos seleccionados en el Backlog para lograr el objetivo propuesto por el Sprint. Al igual que con el Product Backlog, se actualiza a medida que avanza el aprendizaje. Como se mencionó, incluye todos los elementos seleccionados para el Sprint, así como el objetivo del Sprint [9].

#### 4.5.15 Increment

Un Increment constituye la adición de todas las historias del último Sprint terminado, así como de anteriores Sprints [9]. “Representa un peldaño concreto hacia el objetivo del producto” [9].

### 4.6 Tecnologías de Desarrollo Web

#### 4.6.1 Python

Python es un lenguaje de muy alto nivel, que incorpora conceptos como la comprensión de listas y la programación funcional. Esto permite crear programas compactos que contienen mucha funcionalidad en poco código, mejorando la legibilidad y reduciendo el mantenimiento. Python viene con "pilas incluidas". Python se distribuye con una gran cantidad de bibliotecas (módulos) bien diseñadas y mantenidas que proporcionan cualquier cosa, desde el acceso a archivos .csv y el análisis sintáctico de XML, hasta la construcción de un servidor HTTP con un puñado de código, y estos módulos están al menos tan bien documentados como el propio lenguaje. Todo esto

significa que podemos reducir el tiempo de desarrollo ya que en muchos casos no tenemos que reinventar la rueda nosotros mismos [10].

#### 4.6.2 Django

Django es un marco de trabajo web (del inglés framework) para Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático. Creado por desarrolladores experimentados, se encarga de gran parte de la molestia del desarrollo web, por lo que los desarrolladores pueden concentrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto [11] Django ofrece la ventaja de ser robusto gracias a sus diversas librerías y complementos, seguro ya que toma en serio los riesgos de seguridad más comunes en el desarrollo web, y sobre todo escalable ya que permite incrementar rápida y fácilmente la complejidad del sitio web.

#### 4.6.3 PostgreSQL

En [12] se menciona que “PostgreSQL es un SGBD de tipo objeto-relacional con licencia de tipo libre”. Esta base de datos es del tipo SQL que a diferencia de otros SGBD’s libres como MySQL o Ingres es más potente acercándose a competir con bases de datos propietarias de la talla de Oracle o SQLServer. PostgreSQL es mantenido por la comunidad por lo que no pertenece a una compañía en concreto [12].

### 4.7 Pruebas de Software

Las pruebas de software (testing en inglés) son los procesos que permiten verificar y revelar la calidad de un producto software antes de su puesta en marcha. Básicamente, es una fase en el desarrollo de software que consiste en probar las aplicaciones construidas [13]. Las pruebas de software tienen diversas definiciones. Como referencia se toma una de las definiciones citada por IEEE y SWEBOK [14] Una prueba es una actividad ejecutada para evaluar y mejorar la calidad del producto a través de la identificación de defectos y problemas [14].

#### 4.7.1 Tipos de Pruebas

Existen diferentes pruebas que sirven para asegurar la calidad, robustez, escalabilidad y alcance del software, para la realización de este trabajo compete revisar las siguientes: Pruebas de Casos

de Uso, Pruebas de Carga y Estabilidad, Unitarias, Seguridad, Código y de Aceptación del Usuario.

**Pruebas de Casos de Uso:** este tipo de pruebas se encarga de asegurar que para diferentes entradas exista una salida deseada sin importar como se ejecuta el código. Con estas pruebas se pretende encontrar:

- Funcionalidades incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.

**Pruebas de Carga y Estabilidad:** permiten determinar que el sistema puede soportar el flujo de carga definido en las especificaciones definiendo la eficiencia en la realización de las operaciones, para un sistema web, se probará la capacidad para aceptar peticiones (capacidad de concurrencia).

**Pruebas de Seguridad:** consisten en validar que el sistema de autenticación del software, así como los privilegios y permisos asignados para cada rol de usuario. Permiten garantizar que la información se mantenga segura y que el software no sea presa de ataques que involucren la pérdida parcial o total de la información.

**Pruebas Unitarias:** Estas pruebas las ejecuta el desarrollador, cada vez que va probando fragmentos de código o scripts para ver si todo funciona como se desea. Estas pruebas son muy técnicas. Por ejemplo, probar una consulta, probar que un fragmento de código envíe a imprimir un documento, probar que una función devuelva algo, etc. Para obtener el mejor éxito en este tipo de pruebas se debe cumplir con:

- No debería existir intervención manual.
- Deben cubrir la mayor cantidad de código.
- Repetibles o Reutilizables.
- Independientes en su ejecución para no afectar a la ejecución de otra.

**Pruebas de Código:** consiste en verificar la calidad en la codificación de un software; para ellos se revisa que no existan duplicidad, que la comprensión cognitiva de las funcionalidades se encuentre por un rango inferior a 15 puntos, que no existan líneas de código muertas y advertir



sobre potenciales bugs que se podrían presentar debido a la organización del código o el llamado incorrecto de los recursos.

**Pruebas de Aceptación de Usuario:** tienen la finalidad de determinar si el software está listo para su implementación por parte del usuario operativo. Se divide en dos, las pruebas alfa y las pruebas beta. Las pruebas alfa se realizan por el usuario que hará uso del producto bajo la supervisión de los desarrolladores; las pruebas beta por su parte, las realiza el usuario cuando el equipo de desarrollo entregue una versión casi definitiva del producto.

#### 4.8 Trabajos Relacionados

Los trabajos relacionados que se encontraron durante la revisión de literatura fueron *“An Online Scholarship Application System”* [15], el *“Sistema informático para la gestión del programa de becas de la asociación para el desarrollo humano”* [16] y finalmente el *“Diseño, desarrollo e implementación de un sistema de información web para la gestión de créditos, subsidios y becas del fondo para la educación técnica tecnología y superior de la secretaría de educación del municipio de Girardot para el año 2021”* [17]. Estos trabajos están estrechamente relacionados con la propuesta de este TT. En la Tabla 2 se detalla un resumen de cada uno de los trabajos mencionados.

Tabla 2. Resumen de los trabajos relacionados.

Trabajo	Resumen
<b>An Online Scholarship Application System</b>	“Este manuscrito describe el desarrollo de un nuevo sistema de software para la solicitud y tramitación de becas mediante la reingeniería del proyecto piloto” [15]. “Algunas de las características importantes de este nuevo sistema son el procesamiento eficiente y la interactividad entre múltiples usuarios en un entorno de red altamente seguro” [15]. “El documento también describe algunos de los retos encontrados durante cada fase del ciclo de vida del desarrollo, al tiempo que añade más funcionalidades y técnicas introducidas para resolver los problemas antes mencionados” [15].
<b>Sistema informático para</b>	“El documento se divide en tres importantes capítulos: Estudio Preliminar,

---

**la gestión del programa de becas de la asociación para el desarrollo humano** Análisis y Diseño del Sistema, y Plan de Implementación” [16].

“El Estudio Preliminar muestra las condiciones actuales de procesos de gestión del programa de becas en ADHU por las cuales es necesario un sistema informático que ayude a mecanizar sus procesos y gestionar más eficientemente su información” [16].

“El capítulo de Análisis y Diseño del Sistema, explica detalladamente la solución propuesta para el sistema informático, que ayudará a ADHU en la gestión de la información de su programa de becas” [16].

“El último capítulo, Plan de Implementación, describe el proceso que se debe de seguir para echar a andar el sistema informático, de tal forma que no se presenten ningún tipo de dificultades durante la ejecución del proceso” [16].

---

**Diseño, desarrollo e implementación de un sistema de información web para la gestión de créditos, subsidios y becas del fondo para la educación técnica tecnología y superior de la secretaría de educación del municipio de Girardot para el año 2021** “El sistema de información web es un software desarrollado para la administración los diferentes créditos del Fondo para la Educación Técnica, Tecnológica y Superior del municipio de Girardot, el cual fue realizado debido a que la secretaría de educación del municipio de Girardot, ente gubernamental que administra este fondo, tenía problemas con el correcto tratamiento de la información, seguridad de la información, redundancia y no clasificación adecuada de la información” [17]

“Para la realización de este software se utilizó la metodología de desarrollo ágil XP, además, se utilizaron herramientas mayormente libres y gratis como PHP, Laravel, MYSQL, Bootstrap, JQuery, entre otros, todo esto implementado en un servidor capacitado para múltiple tráfico de usuarios, obteniendo así una mejora en los diferentes procesos del fondo además de cubrir la demanda actual y futura de beneficiarios y funcionarios que operan y operarán el sistema de información web” [17]

---

## 5. Metodología

### 5.1 Contexto

El Trabajo de Titulación (TT) de carácter académico se lleva a cabo en la Carrera de Ingeniería en Sistemas (CIS), perteneciente a la Facultad de las Energías, las Industrias y los Recursos Naturales no Renovables (FEIRNNR), de la UNL en colaboración con la DTI durante el periodo comprendido entre octubre de 2021 a septiembre de 2022. El desarrollo del presente TT se lleva a cabo en respuesta a la necesidad de automatización de los procesos de becas solicitado por la UBU [véase Anexo 16]. La propuesta es acogida por el autor mediante el acuerdo que mantiene la DTI con la CIS para aquellos estudiantes que optan por desarrollar el TT puedan vincularse con un proyecto de desarrollo de software en favor de la comunidad universitaria.

### 5.2 Procedimiento

A fin de alcanzar el objetivo de estudio del TT se definen 3 etapas conformadas por cada uno de los objetivos específicos con sus actividades:

#### 1. Analizar los procesos que intervienen en la Gestión de Becas mediante el uso de diagramas BPMN

- Se identificaron los procesos y subprocesos que intervienen en la Gestión de Becas a través de entrevistas con los involucrados en estos procesos [véase Anexo 2, Anexo 17, Anexo 19].
- Se modeló los Procesos que intervienen en la Gestión de Becas con el uso de un software modelador de BPMN [véase Anexo 1 - sección 1.2].
- Se validó el modelado BPMN de los procesos que intervienen en la gestión de becas, con los interesados en su desarrollo mediante un acta de validación [véase Anexo 13 y Anexo 14].

#### 2. Desarrollar el módulo para el proceso de seguimiento y control de becas, utilizando el framework Django y la metodología XP

- Se especificaron los requisitos para el proceso de seguimiento y control de becas mediante el uso del estándar IEEE 830 [véase Anexo 3].
- Se diseñó el diagrama de clases para el proceso de seguimiento y control de becas con

el uso de diagramas UML 14 [véase Anexo 1 – sección 2.2].

- Se elaboró el Product Backlog mediante la técnica Users Story Mapping y se validó con el Product Owner [véase Anexo 10].
- Se codificaron los Procesos para el Seguimiento y Control de Becas [véase Anexo 1 - sección 6.2.4 y Anexo 18].

### **3. Validar la implementación del módulo para el proceso de seguimiento y control de becas en un entorno de pruebas**

- Se planificaron las Pruebas en un Entorno Simulado para la validación de funcionamiento del módulo desarrollado [véase Anexo 6].
- Se ejecutó el Plan de Pruebas en un Entorno Simulado [véase Anexo 7].
- Se elaboró la documentación necesaria para la implantación y uso del módulo desarrollado [véase Anexo 4 y Anexo 5].

### **5.3 Recursos**

Con la finalidad de lograr el cumplimiento de los objetivos planteados, se hizo uso de los siguientes recursos:

- **Investigación exploratoria:** esta técnica fue utilizada para esclarecer los conceptos teóricos relacionados al campo de estudio del TT logrando la comprensión y entendimiento de estos.
- **Entrevistas:** esta herramienta cualitativa permitió recopilar información de la mano de los interesados en el desarrollo de este TT, para comprender los procesos y subprocesos que se llevan a cabo durante la gestión de becas, con el fin de abstraer todos los requerimientos necesarios para la construcción del sistema.
- **Estudios de caso:** esta técnica en conjunto con la investigación exploratoria permitió analizar casos similares a la propuesta de este TT en la implementación de un módulo para la gestión de becas.
- **Investigación Bibliográfica:** esta técnica fue útil para investigar acerca de la teoría que sustenta el presente trabajo, por medio de la búsqueda y recopilación de diversas fuentes de consulta como: libros, artículos, revistas y otras fuentes secundarias. Mediante esta técnica se hizo una discriminación para destacar los conceptos esenciales evaluando su

confiabilidad y sacando las conclusiones correspondientes.

- **Software de Diseño:** se hizo uso del software modelador de BPMN denominado Bizagi Modeler como herramienta principal, y como herramienta de respaldo se hizo uso de la herramienta en línea BPMN.io, esto con la finalidad de tener una base de respaldo ante posibles problemas de funcionamiento de alguna de las herramientas mencionadas. Se eligió estas herramientas por ser de uso libre y gratuito para evitar incurrir en costos en la realización de el modelado de los procesos para la gestión de becas. Para el uso de diagramas UML y se hizo uso de la herramienta en línea Draw.io y para el prototipo de interfaces frontend del módulo se hizo uso de Adobe XD en su versión gratuita.
- **Software de Desarrollo:** el editor de código fuente Visual Studio Code, El software de Control de Versiones Git, GitHub como plataforma de almacenamiento del repositorio, El lenguaje de programación Python, la Base de Datos PostgreSQL.
- **Otras herramientas:** se hizo uso de diversos tipos de software que aportaron de manera directa o indirecta a la realización de este TT tales como: Zoom para la comunicación en entrevistas y para coordinar el trabajo con DTI, Jira ya que integra Scrum y permite coordinar el trabajo de manera adecuada.
- **Marco de Trabajo SCRUM:** el desarrollo de software se llevó a cabo bajo este marco de trabajo escogido por DTI, ya que permite coordinar el trabajo en equipo y organizar las actividades en Sprints para ir entregando continuamente Increments que conduzcan el trabajo hacia la dirección deseada, así como corregir a tiempo eventuales inconvenientes previstos o no.
- **Metodología XP:** se hizo uso de esta metodología de desarrollo de software en base al segundo objetivo para implementar todo el ciclo de vida del software.

#### 5.4 Participantes

El presente Trabajo de Titulación fue realizado por Jackson Guzmán estudiante de la Carrera de Ingeniería en Sistemas (CIS), orientado por el equipo de trabajo designado por la DTI encabezado por el Ing. Danny Muñoz y los ingenieros Máximo Álvarez y Johana Cueva como soporte en la coordinación de las actividades. La colaboración de la Lcda. Marlene Valdez como responsable de la comisión de becas, el Lic. Juan Sebastián Varela contador financiero y la Lcda. Mónica Mora

tesorera financiera. El trabajo estuvo bajo la dirección del Ing. Pablo Fernando Ordoñez Ordoñez Mg. Sc., director docente de la CIS - UNL.

## **6. Resultados**

En esta sección se detallan los resultados más relevantes obtenidos en el transcurso del desarrollo del TT, organizado por cada uno de los objetivos específicos planteados. Como resultado del primer objetivo específico se presenta el modelado de los procesos que intervienen en la gestión de becas, el segundo objetivo describe desde la especificación de requisitos, el diseño del diagrama de clases y ejemplifica a través de 2 historias de usuario y la descripción de un componente como se llevó a cabo el proceso de codificación, presentando la estructura del módulo informático. El capítulo finaliza describiendo los 6 tipos de pruebas aplicadas para validar la implementación del módulo en un ambiente de pruebas.

### **6.1 Objetivo 1: Analizar los procesos que intervienen en la Gestión de Becas mediante el uso de BPMN**

#### **Procesos que intervienen en la Gestión de Becas**

Una vez establecido e identificado cada uno de los procesos que se llevan a cabo en las Etapas para la Gestión de Becas mediante las entrevistas aplicadas a los involucrados y el análisis del reglamento de becas [véase Anexo 1 – apartado 1.1], se procedió a modelar los procesos a través de diagramas BPMN que permitieron visualizar el flujo y las relaciones existentes entre estos, así como el personal que interviene y las interacciones necesarias. Para el modelado de los procesos se hizo uso del software Bizagi Studio que además permitió validar el diseño elaborado [véase Figura 32].

#### **Modelo BPMN general de procesos**

El modelado de estos procesos fue validado por parte de los interesados en el desarrollo (DTI y la UBU) mediante una reunión de trabajo con sus representantes [véase Anexo 13 y Anexo 14]. Primeramente, se creó el diagrama general de Procesos para la Gestión de Becas en la que se muestra todas las Etapas involucradas y los Procesos que intervienen [véase Figura 3].

En la Gestión de Becas intervienen 6 etapas. La primera etapa es la de postulación, donde los solicitantes presentan sus solicitudes y cumplen con los requisitos necesarios. Luego, la etapa de revisión, donde el responsable de la sección de becas evalúa cada solicitud y la documentación adjuntada asignando una calificación a la ficha socioeconómica. La tercera etapa es la de





### 6.2.1 Especificación de los Requisitos de Software para el proceso de seguimiento y control de becas mediante el uso del estándar IEEE 830

Esta actividad inició luego del análisis de los procesos que intervienen en cada una de las Etapas para la Gestión de Becas y tuvo por objetivo definir y establecer los requisitos de software que fueron cubiertos en el desarrollo del módulo informático. La elaboración de la Especificación de Requisitos de Software (ERS) precisó el propósito, alcance, personal involucrado, perspectiva, restricciones suposiciones y dependencias, así como los requisitos y detalles necesarios para garantizar un desarrollo exitoso del MGB. En particular, se incluyó el diagrama de casos de uso, que permite visualizar de manera clara y concisa los diferentes casos de uso o escenarios en los que el sistema será utilizado [véase Figura 4]. Además, se incluyeron los requisitos funcionales y no funcionales, los cuales fueron propuestos propusieron en base al Reglamento de Becas vigente [1], las consideraciones obtenidas a partir de las entrevistas generadas con los interesados en su desarrollo [véase Anexo 17] y los modelos BPMN diseñados en la actividad anterior. La relación entre cada caso de uso y los requisitos funcionales y no funcionales es clara y detallada en el ERS. Finalmente, se incluyó el prototipado y las historias de usuario, lo que permitió tener una visión completa de las funcionalidades esperadas y las necesidades de los usuarios en el MGB. Para acceder a una revisión detallada véase Anexo 3.

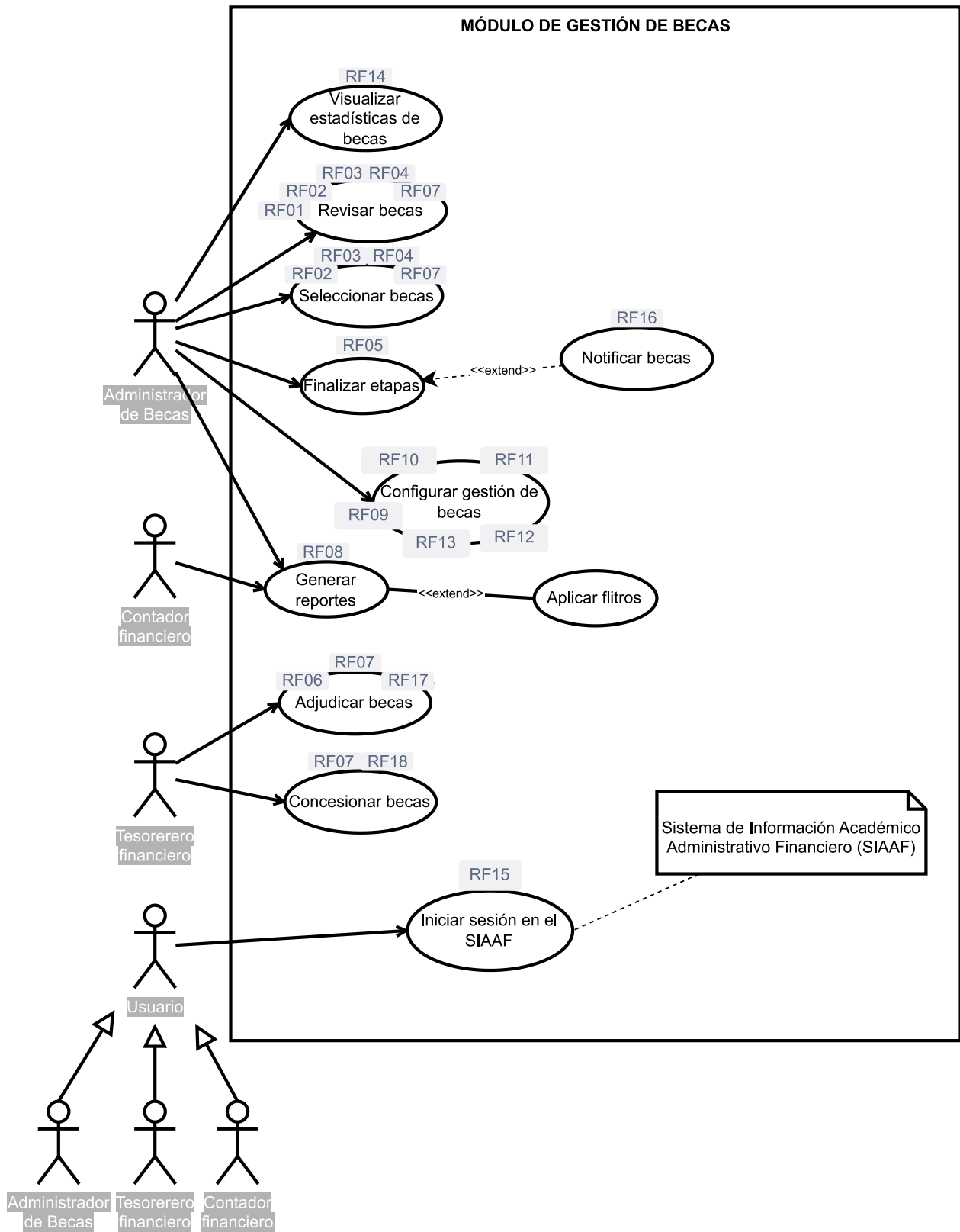


Figura 4. Diagrama de Casos de Uso General para el desarrollo del módulo informático MGB.

A continuación, se muestran los Requisitos Funcionales que se establecieron para alcanzar el Objetivo del Proyecto [véase Tabla 3]. Los requisitos de software fueron validados por DTI y la UBU como cliente [véase Anexo 13 y Anexo 14]

*Tabla 3. Resumen de los Requisitos Funcionales para el desarrollo del módulo informático.*

<b>Identificador</b>	<b>Nombre</b>
<b>RF01</b>	Sincronizar datos de postulación desde el MGT hacia el MGB
<b>RF02</b>	Actualizar estados de postulaciones
<b>RF03</b>	Descargar documentos de postulación
<b>RF04</b>	Ver trámite en el MGT
<b>RF05</b>	Finalizar Etapas
<b>RF06</b>	Actualizar estados de Becas
<b>RF07</b>	Visualizar Postulaciones y Becas
<b>RF08</b>	Generar Reportes
<b>RF09</b>	Agregar mensaje frecuente
<b>RF10</b>	Actualizar mensaje frecuente
<b>RF11</b>	Eliminar mensaje frecuente
<b>RF12</b>	Marcar como favorito mensaje frecuente
<b>RF13</b>	Actualizar montos de pago para los tipos de beca
<b>RF14</b>	Visualizar estadísticas generales de becados
<b>RF15</b>	Iniciar Sesión
<b>RF16</b>	Notificar resultados de postulación
<b>RF17</b>	Guardar registro de becado en el sistema financiero
<b>RF18</b>	Registrar Pago de Becas

### 6.2.2 Diseño del Diagrama de Clases para el proceso de seguimiento y control de becas con el uso de UML

Esta actividad consistió en el diseño del diagrama de clases para estructurar los datos del módulo informático y las relaciones necesarias entre estos. Además, se prestó especial cuidado para evitar la existencia de relaciones ambiguas, inconsistentes, redundantes o información duplicada en distintas clases. Primeramente, se diseñó una propuesta del diagrama para estructurar los datos, y las relaciones existentes de acuerdo al levantamiento de requisitos descritos en la actividad anterior. Esta propuesta fue presentada al equipo de desarrollo de DTI, quienes revisaron y aportaron con cambios significativos en el correcto uso de relaciones que permita soportar la infraestructura del módulo informático [véase Anexo 15].

Cabe indicar que el módulo informático formará parte del SIAAF, por lo que, algunos de los modelos que se diseñaron como propuesta, fueron sustituidos por otros preexistentes en distintos módulos del sistema; es por ello que el diagrama de clases tuvo que ser modificado, para incluir las relaciones con estos modelos. En la Figura 5 se los destaca en color amarillo.

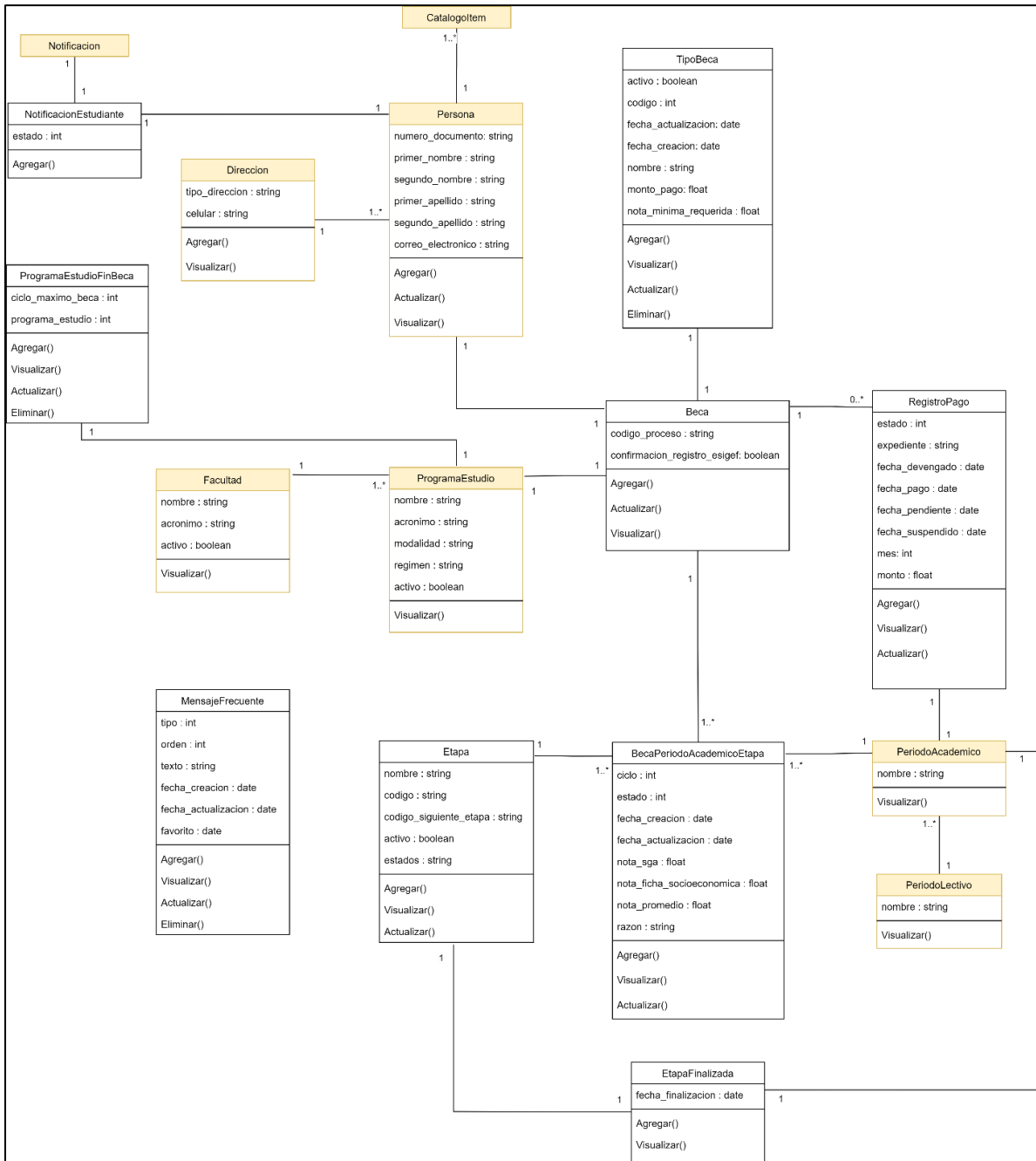


Figura 5. Diagrama de Clases del MGB.

El diseño del diagrama de clases permitió representar cada clase en Django con sus respectivos atributos y relaciones con otras clases. Estos elementos son importantes para el ORM (del inglés Object Relational Model) de Django, ya que este se encarga de transformar los modelos de Django en modelos de base de datos para la base de datos PostgreSQL [véase Anexo 11]. Por lo tanto, el diagrama de clases fue una herramienta útil para visualizar las relaciones y los atributos de los

modelos de Django, y para garantizar que el ORM de Django genere correctamente los modelos de base de datos a partir de los modelos de Django.

### 6.2.3 Historias de Usuario

A partir de cada requisito de software se elaboraron sus correspondientes historias de usuarios con la descripción, los escenarios de prueba y los criterios de aceptación de cada una de ellas. Además, en cada una se especificaron la prioridad y los puntos estimados para completar el trabajo. Se agregaron variantes considerando los distintos roles o las funcionalidades adicionales que se deberían tomar en cuenta.

A continuación, se presentan 2 historias de usuario tomadas para ejemplificar este resultado [véase Tabla 4 y Tabla 5], para una descripción completa véase Anexo 3 sección Historias de Usuario.

*Tabla 4. Historia de Usuario para notificar los resultados de la postulación a tesorería y a los postulantes.*

<b>ID: H016</b>		<b>Notificar resultados de postulación</b>	
<b>Usuario / Rol</b>	Usuario Administrador de Becas	<b>Prioridad</b>	Media
		<b>Puntos estimados</b>	40
<b>Descripción</b>	Como Usuario Administrador de Becas, quiero notificar a los postulantes los resultados de su postulación, y al Usuario de Tesorería notificar la existencia de nuevos becados, para que los estudiantes conozcan los resultados de su postulación y el Usuario de Tesorería sepa que puede empezar el proceso de registro de nuevos becados.		
<b>Nro</b>	<b>Escenario</b>	<b>Criterio de aceptación</b>	
1	<p>En el correo de un postulante llegará la notificación de los resultados de su postulación.</p> <p>En el correo del Usuario de Tesorería llegará una notificación de nuevos becados.</p> <p>En el MGB aparecerá una notificación para el Usuario de Tesorería.</p>	<p>Al finalizar la Etapa de Selección de cada Periodo Académico de Postulación, los postulantes deben recibir una notificación en el correo electrónico institucional usado en la Etapa de Postulación, con los resultados obtenidos.</p> <p>Al finalizar la Etapa de Selección de cada Periodo Académico de Postulación. El Usuario de Tesorería recibirá una notificación vía correo electrónico.</p> <p>El usuario de tesorería debe recibir una notificación dentro del MGB minutos después de</p>	

---

finalizada la etapa de selección.

---

*Tabla 5. Historia de Usuario para registrar el pago de Becas.*

---

<b>ID: H018</b>	<b>Registro de Pago de Becas</b>		
<b>Usuario / Rol</b>	Usuario de Tesorería, Usuario de Departamento Financiero	<b>Prioridad</b>	Alta
		<b>Puntos estimados</b>	50
<b>Descripción</b>	Como Usuario de Tesorería, quiero importar los registros de pago de becas en formato Excel generado previamente por el administrador de becas en el MGB como parte de la solicitud de pago recurrente y completado manualmente con el código de devengado correspondiente por mí, para llevar el control de los becados a los que ya se ha realizado el pago.		
<b>Nro</b>	<b>Escenario</b>	<b>Criterio de aceptación</b>	
1	Al iniciar sesión con el rol Usuario Tesorería, podrá visualizar el listado de registros de pago pendiente.  Al importar un archivo de pago, se actualizarán los registros de pago.  Si al importar se detectan errores, el módulo mostrará en una tabla la lista de errores.	El Usuario de Tesorería debe ser capaz de visualizar el listado de registros de pago pendiente.  El usuario de Tesorería debe poder importar los registros de pago de acuerdo al formato acordado.  El usuario de Tesorería debe poder actualizar individualmente los registros de pago seleccionando un registro desde el listado.  Si el archivo a importar contiene errores el módulo debe ser capaz de indicar dichos errores.	
<b>Descripción</b>	Como Usuario de Contabilidad Financiera necesito visualizar el estado actual de pagos de los Becados para armar el presupuesto anual y emitir órdenes de pago.		
<b>Nro</b>	<b>Escenario</b>	<b>Criterio de aceptación</b>	
2	Al iniciar sesión con el rol de Usuario Contabilidad Financiera, se podrá visualizar el listado de registros de pago pendiente.  Se podrá generar reportes del estado actual de pagos.	El Usuario de Departamento Financiero debe ser capaz de visualizar el listado de registros de pago pendiente.  El usuario del Departamento Financiero debe poder generar un reporte del estado actual de pagos.	

#### 6.2.4 Codificación de los procesos para el seguimiento y control de Becas

El MGB forma parte del SIAAF e interactúa con el MGT como se muestra en el diagrama de

despliegue de la Figura 6, el MGB fue diseñado en base al diagrama de componentes que se muestra en la Figura 8. Se compone principalmente de 2 directorios denominados app y api (del inglés, Application Programming Interface) [véase Figura 9]. El api posee una View que se encarga de proveer una interfaz de comunicación al MGT para el registro de postulantes. La app2 Becas por su parte, posee todos los componentes del módulo siguiendo el patrón de arquitectura de software MVT (del inglés Model, View, Template) definido por Django, los cuales de acuerdo con la arquitectura establecida por DTI, las Views se orquestan a nivel de capas que se encargan de obtener, crear y actualizar información a través de los Models y estos a su vez utilizando el driver PostgreSQL para conectarse a la base de datos. Además, el MGB hace uso de los modelos de otros módulos del SIAAF; como la app Seguridad que se utiliza para la identificación de los usuarios, la app Core de la que se usa el modelo Persona para registrar los datos de los postulantes, la app Configuración que se usa para almacenar valores parametrizables, y la app Académico para obtener datos acerca del periodo académico [véase Figura 7].

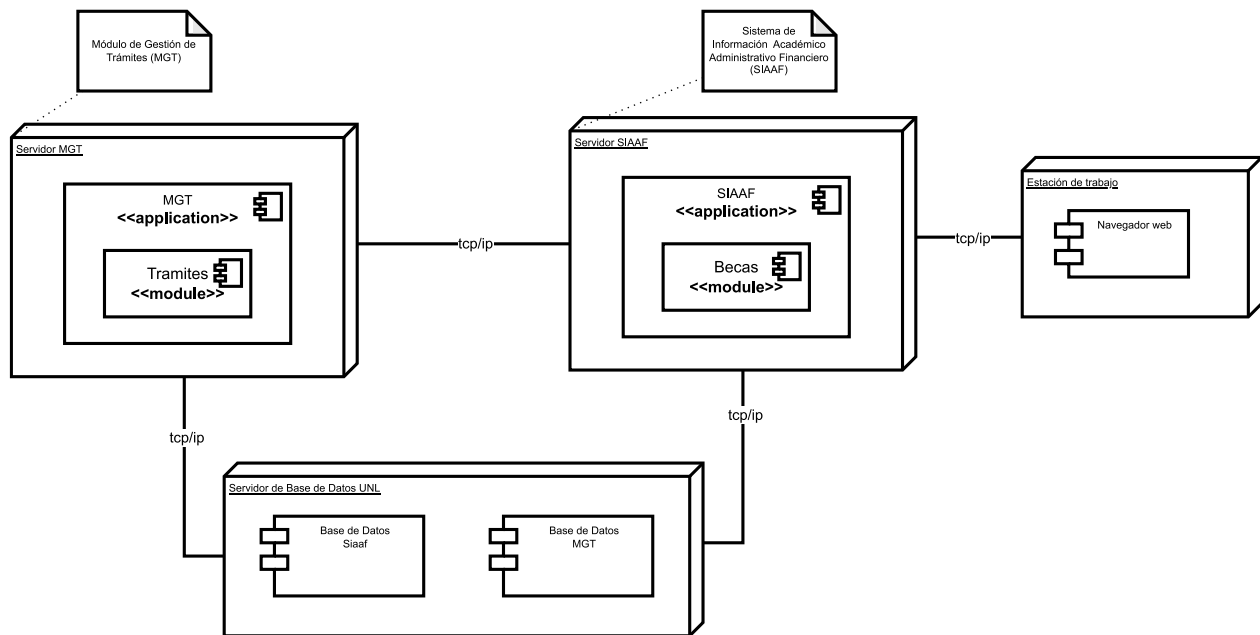


Figura 6. Diagrama de Despliegue.

<sup>2</sup> Django denomina apps a todos los módulos del sistema, no confundir con una aplicación o sistema que interactúa independientemente.



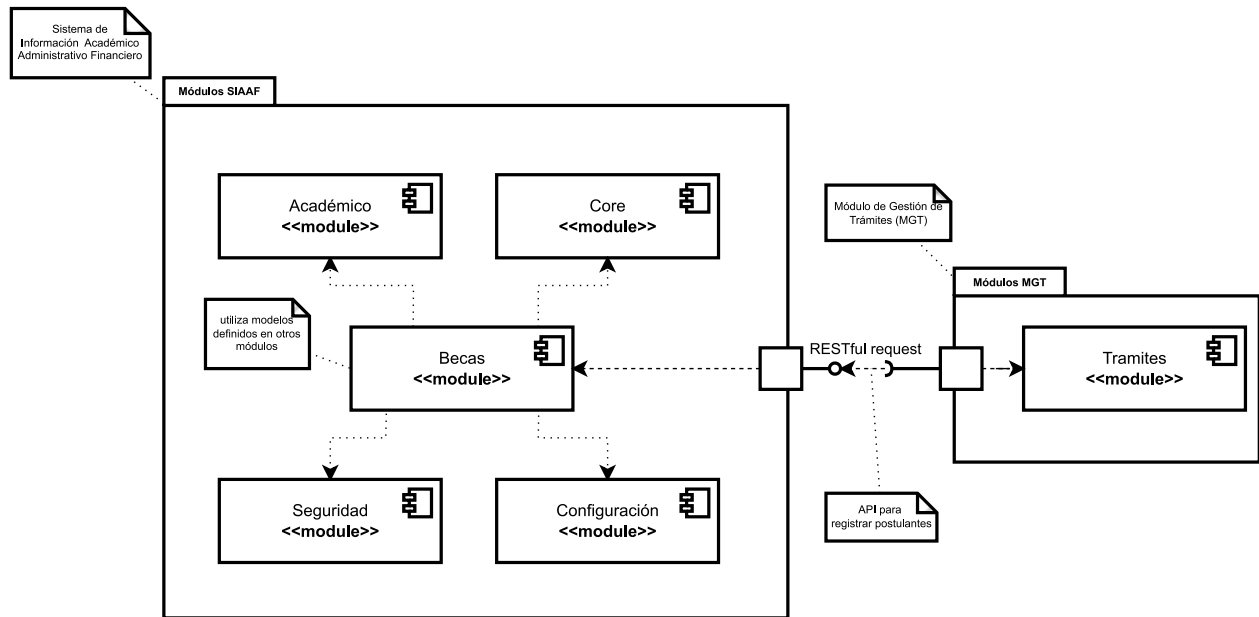


Figura 7. Diagrama de Componentes donde se muestra los módulos del SIAAF con los que interactúa el MGB y la conexión con el MGT.

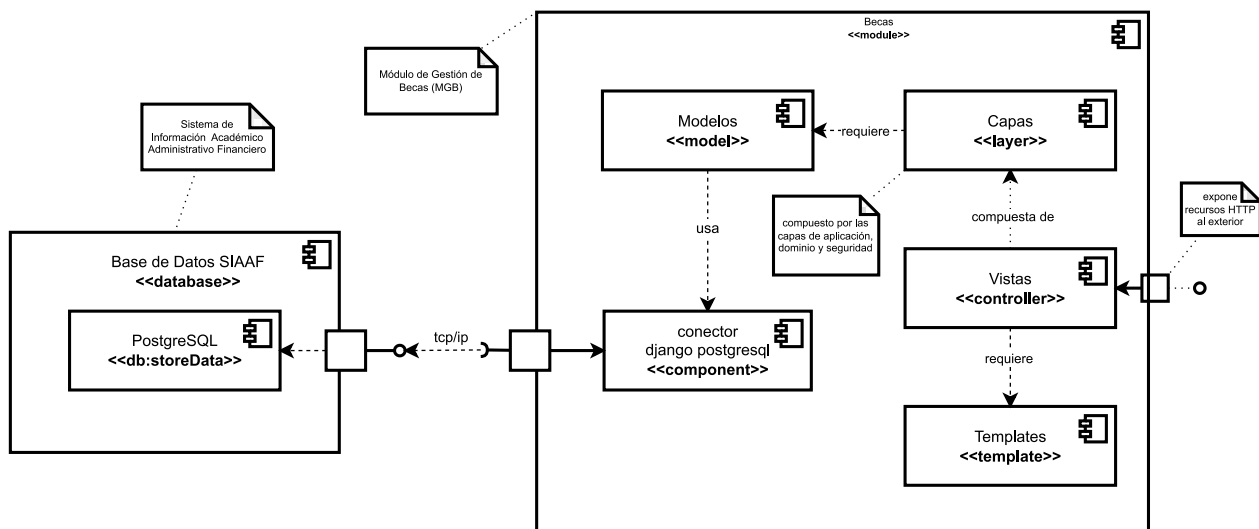


Figura 8. Diagrama de Componentes del MGB.

Para una vista más detallada de los elementos contenidos dentro de cada componente del Módulo de Gestión de Becas (MGB), [véase Anexo 1 – Sección 2.4.1]

### 6.2.5 Estructura de la aplicación

El módulo informático sigue la estructura recomendada por Django en base al patrón MVT: models, views, templates. En base a un patrón de estructura tradicional Modelo, Vista

Controlador (MVC), es importante indicar que Django denomina a un Controlador como Vista y a una Vista como Template.

Además, cabe mencionar que existe un directorio llamado **layer** en el que se establecen 3 tipos de capas: "*application*", "*domain*" y "*security*". En la capa *application* se almacena toda la lógica del código que sirva a las vistas con información. En la capa *domain* se guarda el código que se encarga de las validaciones de datos. Y finalmente la capa de *security* es la encargada de definir la seguridad del módulo.

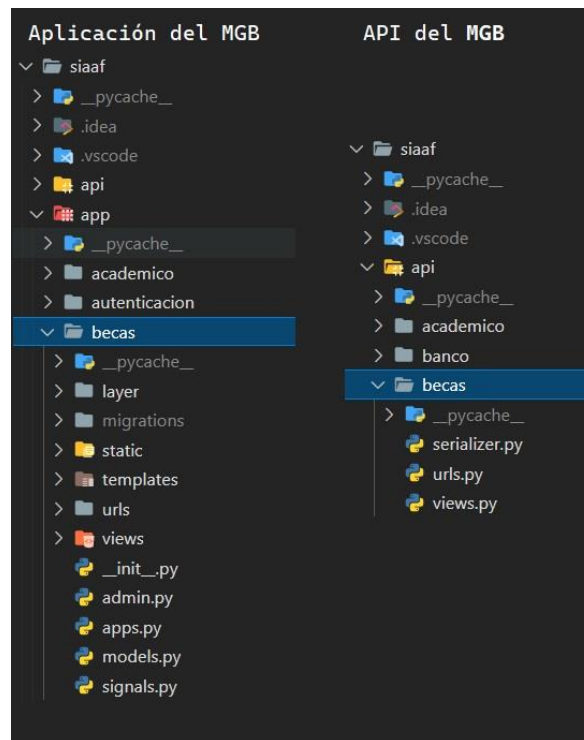


Figura 9. Estructura de la aplicación y api del MGB dentro del SIAAF.

En base al patrón de diseño MVT se describe el proceso de codificación de los componentes mediante cada una de las clases involucradas dentro de los componentes, para ejemplificar se considera al modelo **BecaPeriodoAcademicoEtapa** el cual se encarga de gestionar la información de las becas por cada periodo académico, en cada una de sus etapas. Para ello, se relaciona con otros modelos como el modelo *Beca*, *Etapa* y *PeriodoAcademico* desde los cuales se obtiene la información complementaria [véase Código 1].

Código 1. Modelo Beca Periodo Académico Etapa definido en el archivo `models.py` de la app `Becas`.

```
class BecaPeriodoAcademicoEtapa(models.Model):

    ciclo = SmallIntegerField(default=2, choices=Ciclo.CHOICES)
    estado = PositiveSmallIntegerField(
        default=BecaPeriodoAcademicoEtapaEstados.PENDIENTE,
        choices=BecaPeriodoAcademicoEtapaEstados.CHOICES, null=True, blank=True)
    fecha_creacion = DateTimeField(auto_now_add=True)
    fecha_actualizacion = DateTimeField(auto_now=True)
    nota_sga = FloatField(default=0.0)
    nota_ficha_socioeconomica = FloatField(default=0.0)
    nota_promedio = FloatField(default=0.0)
    razon = CharField(null=True, blank=True, max_length=500,
        help_text="Proporcione una razón por la cual se da el cambio de estado")

    beca = ForeignKey(Beca, on_delete=models.CASCADE)
    etapa = ForeignKey(Etapa, on_delete=models.CASCADE)
    periodo_academico = ForeignKey(PeriodoAcademico, on_delete=models.CASCADE)

    objects = models.Manager()
    becas_objects = BecasObjectsManager()

    def __str__(self):
        return ("%s - %s / %s" % (self.beca, self.etapa, self.periodo_academico))
```

La vista que se muestra en el Código 2 se encarga de renderizar el **Template** de la Etapa de Revisión al proveer información consultando a la Base de Datos a través de la capa de Aplicación y esta a su vez mediante la interfaz de comunicación ofrecida por los modelos.

Código 2. Vista Etapa Revisión definida en el archivo `Etapas.py` de la app `Becas`.

```
@login_required
@permission_required('becas.view_etapa_revision', raise_exception=True, )
def etapa_revision(request):
    """
    Etapa de Revisión Gestión de becas
    :param request:
    :return:
    """
    if request.method == 'GET':
        etapa = Etapa.objects.get(codigo=Constantes.ETAPA_REVISION.value)
        (facultades, programas_estudio, tipos_beca, estados_etapa,
         periodos_academicos) = EtapasAppService.obtener_filtros_etapas(etapa)
        (url_tramite_postulante, url_MGT) = EtapasAppService.obtener_urls()
        etapas_finalizadas = EtapasAppService.obtener_etapas_finalizadas(
```

```

    etapa, periodos_academicos)
cabecera_etapa = 'Listado de Postulaciones - Revisión de la Documentación'
navegacion = ('Etapa de Revisión',
              [('Becas', reverse('becas:index')), ('Etapa Revisión', None)])
return render(request, "becas/etapas/etapa_revision.html", locals())

```

En el Código 3 se muestra la vista complementaria que es responsable de proveer información a la tabla del Template bajo demanda gracias a peticiones POST y al uso de AJAX.

*Código 3. Vista Etapa Revisión Paginador definida en el archivo etapas.py de la app Becas.*

```

@login_required
@require_http_methods(["POST"])
@permission_required('becas.view_etapa_revision', raise_exception=True, )
def etapa_revision_paginador(request):
    """
    Lista de postulantes en la etapa de revision con la paginación de datatable
    :param request:
    :return:
    """
    try:
        etapa_params = DataTableParams(request, **request.POST)
        etapa = Etapa.objects.get(
            codigo=Constantes.ETAPA_REVISION.value)
        EtapasAppService.get_datatable(
            etapa_params=etapa_params, etapa_codigo=etapa.codigo)
        data = [{
            'id': item.id,
            'beca_persona': item.beca.persona.get_nombres_completos_inverso(),
            'beca_persona_correo_electronico': item.beca.persona.correo_electronico,
            'beca_persona_numero_documento': item.beca.persona.numero_documento,
            'beca_persona_direccion_set_celular':
item.beca.persona.direccion_set.first().celular,
            'beca_carrera_facultad': item.beca.carrera.facultad.siglas,
            'beca_carrera': item.beca.carrera.nombre,
            'beca_tipo_beca': item.beca.tipo_beca.nombre,
            'beca_codigo_proceso': item.beca.codigo_proceso,
            # 'periodo_academico': item.periodo_academico.nombre,
            'ciclo': item.ciclo,
            'nota_sga': item.nota_sga,
            'nota_ficha_socioeconomica': item.nota_ficha_socioeconomica,
            'estado': item.get_estado_display(),
            'indice_estado': item.estado,
            'razon': item.razon,
        } for item in etapa_params.items]
        result = etapa_params.result(data)
    
```

```

return JsonResponse(result)

except Exception as e:
return HttpResponseServerError(e)

```

En el Código 4 se observa la estructura principal del Template en la que se hace uso de varios fragmentos de código más pequeños y reutilizables. Se debe señalar que un Template está compuesto de fragmentos, archivos *javascript*, hojas de estilo (*css*) e imágenes.

*Código 4. Template Etapa Revisión definido en el archivo Etaparevision.html de la app Becas.*

```

{% extends './etapa.html' %}
{% block card_head %}
<header>Listado de Postulaciones - Revisión de la Documentación</header>
{% include 'fragmentos/botones_postulantes.html' with opciones=estados_etapa %}
{% endblock %}
{% block urls_etapa %}
const url_etapa_revision_paginador = "{% url 'becas:etapa_revision_paginador' %}";
const url_calificar_ficha_socioeconomica = "{% url
'api.becas:calificar_ficha_socioeconomica' %}";
const url_actualizar_estado = "{% url 'api.becas:etapas_actualizar_estado' %}";
const url_finalizar_etapa = "{% url 'becas:finalizar_etapa' %}";
const url_MGT = "{{url_MGT}}";
const url_tramite_postulante = "{{url_tramite_postulante}}";
const url_zip_documentos_postulante = "{% url 'becas:descargar_documentos_postulantes'
%}";
{% endblock urls_etapa %}
{% block script_etapa %}
{% load staticfiles %}
<script src="{% static 'js/becas/etapa/actualizar_estado.js' %}?v=0.1"></script>
<script src="{% static 'js/becas/etapa/revision.js' %}?v=0.2"></script>
<script src="{% static 'js/becas/etapa/calificar_ficha_socio_economica.js'
%}?v=0.1"></script>
{% endblock script_etapa %}

```

En la Figura 10 se visualiza el *Template* construido por la View que muestra la información obtenida del *Model* previamente descrito en el *Componente Etapa Revisión*. Para una descripción completa del proceso de codificación iterativa véase Anexo 1 – apartado 2.4.3.

**Siaaf** Módulo de Gestión de Becas → ENTORNO DE PRUEBAS Foto JACKSON ANDREWS

← Menú Principal

Tablero

Etapas >

- Revisión
- Selección
- Adjudicación
- Concesión

Configuraciones

### Etapa de Revisión

# Becas > Etapa Revisión

#### Listado de Postulaciones - Revisión de la Documentación

[Actualizar Estado](#)
[Descargar documentos](#)
[Ver Trámite](#)

Facultad: 
 Programa de estudio: 
 Tipo de Beca: 
 Estado: 
 Período Académico:

**Etapa Finalizada**

[Ayuda](#)
[Deseleccionar todo](#)

Mostrar  registros

Buscar:

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómico
<input type="checkbox"/> ALVARADO GONZALEZ MAYURY SILVANA	19007987	mayuryalvaradogonzalez@unl.edu.ec	090420047	FJSA	ADMINISTRACION PUBLICA	Beca Tipo A	3	9.04	7.78
<input type="checkbox"/> COYAGO TUQUERES MAYBETH ARACELY	17917887	maybethcoyagotiquer@unl.edu.ec	090400449	FJSA	ADMINISTRACION PUBLICA	Beca Tipo B	3	8.97	9.5
<input type="checkbox"/> CUEVA YOMARA MERCEDES	19007986	yomara.cueva@unl.edu.ec	090400006	FJSA	ADMINISTRACION PUBLICA	Beca Tipo A	3	8.59	9.2
<input type="checkbox"/> GIRON TORRES GIAMPIER	19007988	giampiergiron@unl.edu.ec	090400447	FJSA	ADMINISTRACION PUBLICA	Beca Tipo B	3	8.75	9
<input type="checkbox"/> GUAMAN DIAZ GABRIELA ANABELL	19007989	gabrielaguaman@unl.edu.ec	090420440	FJSA	ADMINISTRACION PUBLICA	Beca Tipo C	3	9.39	9.7
<input type="checkbox"/> HERRERA JIMENEZ MARIA VANESSA	19007985	vanessaherrerajimenez@unl.edu.ec	090400340	FJSA	ADMINISTRACION PUBLICA	Beca Tipo A	3	9.22	9.8
<input type="checkbox"/> HURTADO ALVERCA DIANA ALEJANDRA	19007982	dianahurtadoalverca@unl.edu.ec	090400007	FJSA	ADMINISTRACION PUBLICA	Beca Tipo C	3	9.26	8.9
<input type="checkbox"/> INIGUEZ VINCES ANAHY ALEJANDRA	19007984	anahy.iniguez@unl.edu.ec	090400339	FJSA	ADMINISTRACION PUBLICA	Beca Tipo B	3	8.61	9.39
<input type="checkbox"/> MAURAT VELEZ JHON MICHAEL	19007980	jhonmaurativalez@unl.edu.ec	090400308	FJSA	ADMINISTRACION PUBLICA	Beca Tipo A	3	8.52	9
<input type="checkbox"/> PINTA AÑAZCO JUAN ANDRES	19007983	juanpintaanzaco@unl.edu.ec	090420007	FJSA	ADMINISTRACION PUBLICA	Beca Tipo A	3	8.58	8.9

Mostrando registros del 1 al 10 de un total de 11 registros (filtrado de un total de 87 registros) [Clic en una fila para seleccionarla](#)

[Anterior](#)
1
2
[Siguinte](#)

Estados disponibles:

No revisar
Revisar
Aprobado
Rechazado

[Generar Reporte](#)
[Finalizar Etapa](#)

2021 © Sistema de Información Académico, Administrativo y Financiero Universidad Nacional de Loja

Figura 10. Etapa de Revisión - MGB.

### 6.3 Objetivo 3: Validar la implementación del módulo para el proceso de seguimiento y control de becas en un entorno de pruebas

#### 6.3.1 Ejecución del Plan de Pruebas en un Entorno Simulado

##### 6.3.1.1 Pruebas de Caso de Uso

Estas pruebas consistieron en diseñar casos de prueba por cada caso de uso con los distintos caminos que pueden llegar a presentarse y validar si se cumplen o se requiere aplicar cambios y correctivos. Durante la ejecución de estas pruebas se presentaron varios inconvenientes detectados tanto por el equipo de calidad de software de DTI como por parte del autor. Las incidencias reportadas se trabajaron en una bitácora diseñada para reportar y analizar incidentes presentes durante la fase de pruebas [véase Anexo 9]. Esta fue una de las fases más largas ya que se llevó un estricto control de calidad para asegurar que el MGB funcione de lo mejor posible y reducir la tasa de incidentes que puedan llegar a presentarse.

A continuación, se presenta la captura abstraída del certificado de Control de Calidad proporcionado por el equipo de QA de la DTI [véase Figura 11].

Proyecto 2021-007 - Implementación del Módulo de Gestión de Becas para Bienestar Universitario - SIAAF							
CU Casos de uso	DESCRIPCIÓN	CP Casos de prueba	CPNE CP NO ejecutados	IR Incidencias reportadas	EI Estado de incidencia	A Aprobado	%
<b>CASOS DE USO FUNCIONALES</b>							
CU001	Sincronizar datos de postulación desde el MGT hacia el MGB	1	0	2	Resuelta	Si	100
CU002	Actualizar estados de postulaciones	9	0	13	Resuelta	Si	100
CU003	Descargar documentos de postulación	2	0	2	Resuelta	Si	100
CU004	Ver trámite en el MGT	2	0	2	Resuelta	Si	100
CU005	Finalizar Etapas	3	0	3	Resuelta	Si	100
CU006	Actualizar estados de Becas	7	0	10	Resuelta	Si	100
CU007	Visualizar Postulaciones y Becas	3	0	5	Resuelta	Si	100
CU008	Generar Reportes	4	0	5	Resuelta	Si	100
CU009	Agregar mensaje frecuente	1	0	2	Resuelta	Si	100
CU010	Actualizar mensaje frecuente	1	0	2	Resuelta	Si	100
CU011	Eliminar mensaje frecuente	1	0	2	Resuelta	Si	100
CU012	Marcar como favorito mensaje frecuente	1	0	2	Resuelta	Si	100
CU013	Actualizar montos de pago para los tipos de becas	1	0	2	Resuelta	Si	100
CU014	Visualizar estadísticas generales de becados	1	0	1	Resuelta	Si	100
CU015	Iniciar Sesión	1	0	1	Resuelta	Si	100
CU016	Notificar resultados de postulación	2	0	3	Resuelta	Si	100
CU017	Guardar confirmación de registro de beca del sistema financiero	2	0	3	Resuelta	Si	100
CU018	Registrar Pago de Becas	8	0	12	Resuelta	Si	100

Figura 11. Capturas de Pantalla a la Tabla que Certifica las Pruebas de Casos de Uso en base a la ERS del MGB (fuente DTI)

### 6.3.1.2 Pruebas de Carga y Estabilidad

Para la ejecución de las Pruebas de Carga y Estabilidad se usó Apache JMeter, configurado con una población de 3 usuarios, ya que corresponde al número máximo de personas simultáneas que interactuaran con el módulo. Las pruebas se dividieron en 2 grupos de hilos, en el primer hilo están asociadas todas las peticiones que se pueden realizar consultas múltiples, y en el segundo grupo de hilos están peticiones atómicas, es decir, aquellas que se pueden realizar una única vez [véase Figura 12].

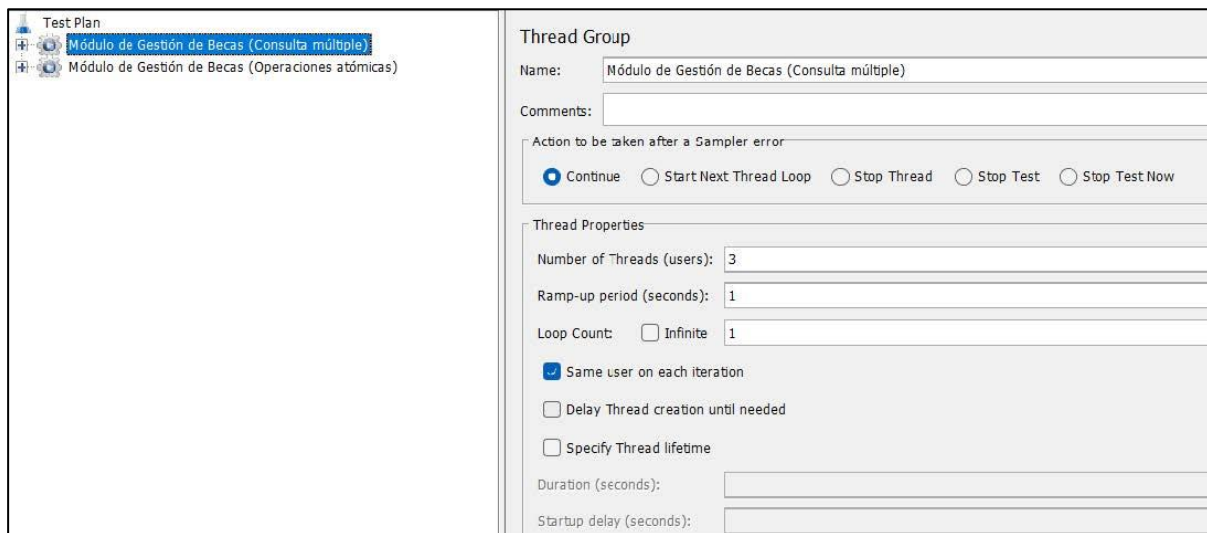


Figura 12. Configuración del Plan de Pruebas con Apache JMeter 5.5

Dentro del grupo de hilos de consultas múltiples se encuentran peticiones de lectura GET como la carga de cada uno de los apartados del módulo, así como sus correspondientes peticiones paginadoras. Además están peticiones POST para la descarga de reportes de cada una de las etapas y la petición de importación de archivos.

Para iniciar la solicitud de las peticiones desde Apache JMeter fue necesario configurar las cookies y las cabeceras, para que el servidor no interprete como una amenaza y permita realizar las consultas, ya que los recursos del módulo se encuentran protegidos y es necesario identificarse con el token de sesión [Figura 17]. Tanto las cookies como las cabeceras se obtuvieron desde una sesión de usuario del navegador.

Se ejecutaron un total de 21 pruebas de carga y estabilidad como se muestra en la Tabla 6.





*Tabla 7. Reporte resumen con los tiempos de respuesta del grupo de hilos de consultas*

<b>Recursos</b>	<b># Muestras</b>	<b>Promedio</b>	<b>Min</b>	<b>Max</b>	<b>Desv. Std</b>
<b>Listar Estadísticas de Becas</b>	3	417	413	422	3.74
<b>Etapa de Revisión</b>	3	292	269	313	18.06
<b>Etapa de Revisión Paginador</b>	3	68	54	76	9.93
<b>Etapa de Selección</b>	3	372	272	484	86.92
<b>Etapa de Selección Paginador</b>	3	54	51	59	3.4
<b>Etapa de Adjudicación</b>	3	291	254	349	41.36
<b>Etapa de Adjudicación Paginador</b>	3	63	52	81	12.83
<b>Etapa de Concesión</b>	3	259	258	261	1.25
<b>Etapa de Concesión Paginador</b>	3	51	49	53	1.63
<b>Configuraciones</b>	3	197	164	223	24.8
<b>Importar Registros</b>	3	181	175	194	8.96
<b>Descargar documentos postulantes</b>	3	774	721	847	53.11
<b>Descargar reporte revisión (completo)</b>	3	90	75	111	15.17
<b>Descargar reporte selección (filtros)</b>	3	89	80	106	11.81
<b>Descargar reporte adjudicación</b>	3	142	132	157	10.66
<b>Descargar reporte concesión</b>	3	74	66	89	10.21
<b>Descargar reporte para importar concesión</b>	3	169	92	323	108.66
<b>Descargar reporte errores al importar concesión</b>	3	36	26	52	11.43
<b>TOTAL</b>	54	201	26	847	183.25

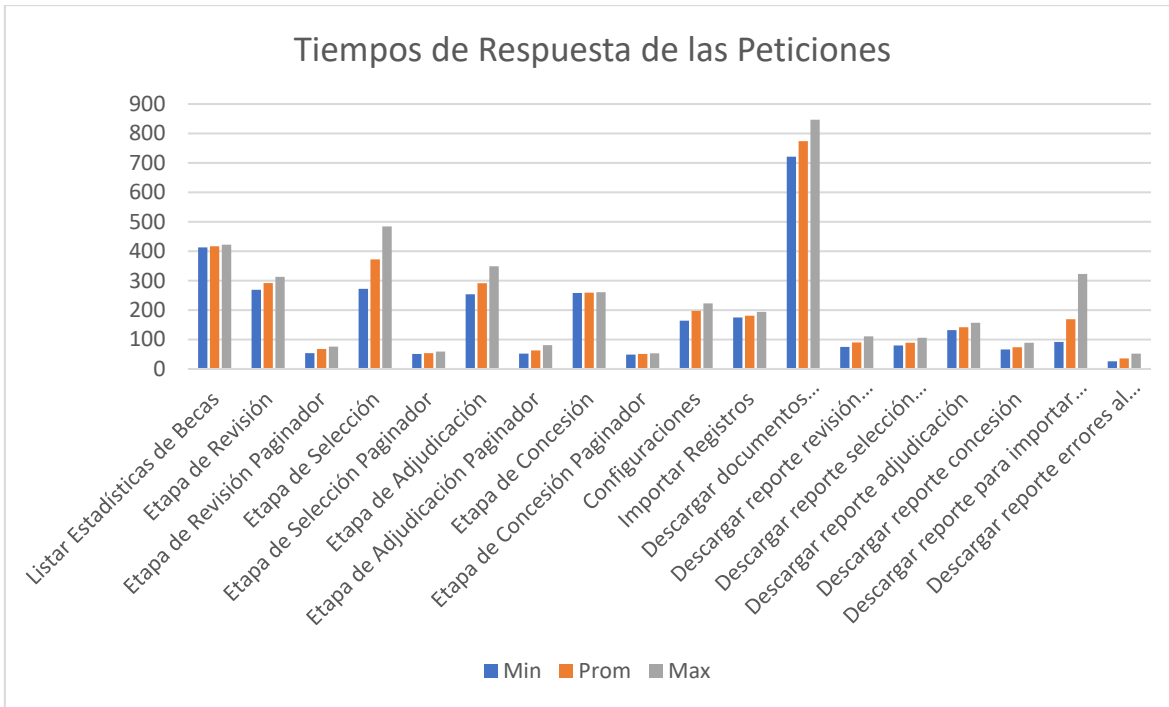


Figura 14. Gráfica del tiempo de respuesta de las peticiones del hilo de consultas múltiples.

En cuanto a las Pruebas de Estabilidad se creó un grupo de hilos específico con todas las peticiones GET manteniendo los parámetros de configuración de 3 usuarios simultáneamente cada segundo, habilitando el tiempo de ejecución de pruebas indefinido y dejando activa la ejecución por 5 minutos exactos [véase Figura 15].

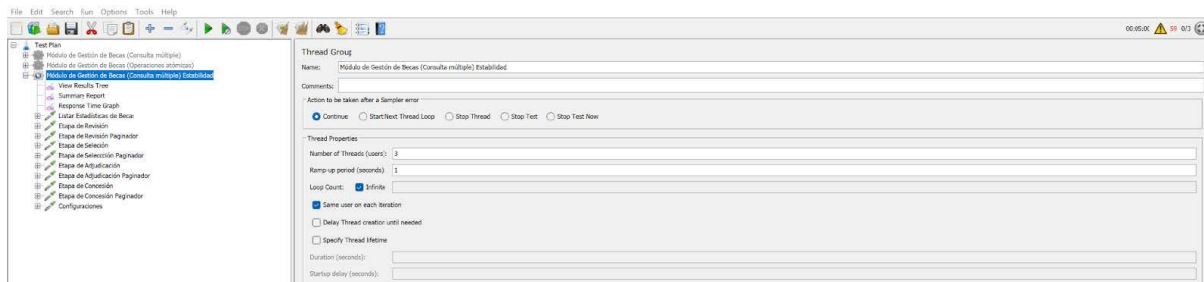


Figura 15. Configuración del grupo de hilos para las Pruebas de Estabilidad con Apache JMeter 5.5

Durante este tiempo se ejecutaron 425 muestras en promedio para cada recurso dando un total de 4257 peticiones a los recursos del módulo, de los cuales se obtuvo un tiempo de respuesta mínimo de 44 ms, un tiempo máximo de 1108 ms, tiempo promedio de ejecución de 211 ms con una desviación estándar de 122 ms y un porcentaje de error del 0% para todos los casos [véase

Anexo 7 - Pruebas de Carga y Estabilidad]. Analizando la gráfica arrojada por Apache JMeter en las pruebas de estabilidad [véase Figura 16], se puede asegurar que, de 10 peticiones en total, el 40% se mantienen en un umbral inferior a los 120 ms, el 50% no supera los 210 ms y un 90% se mantiene inferior a los 220 ms.

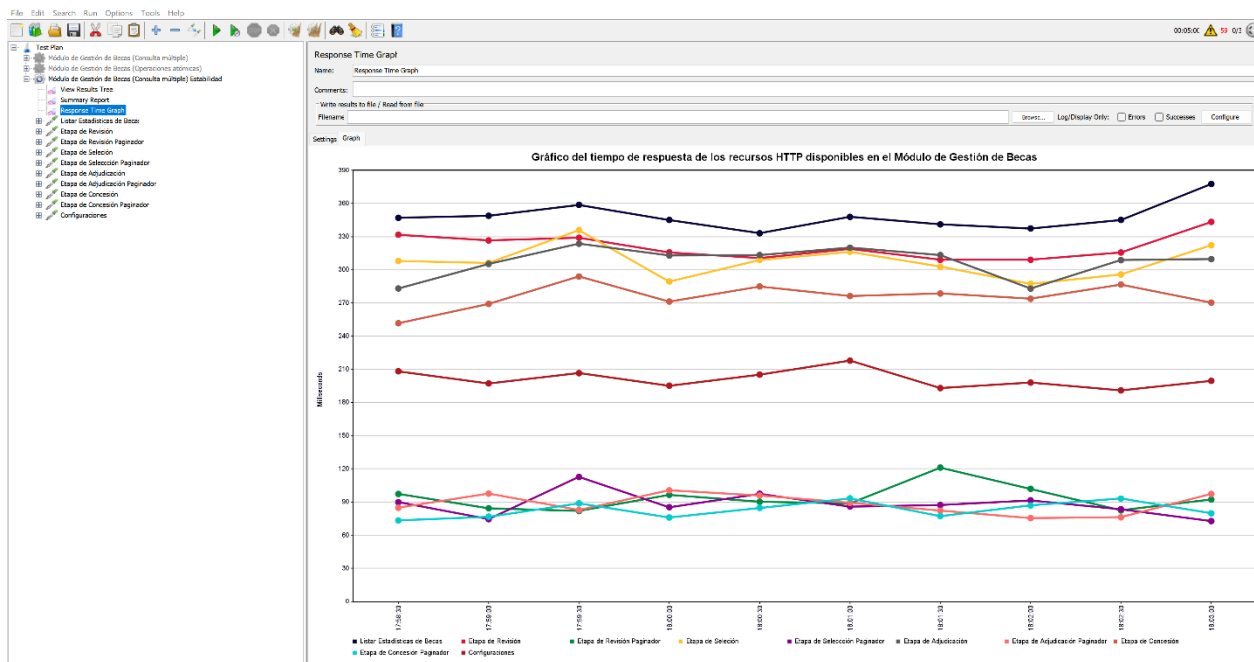


Figura 16. Gráfica del tiempo de respuesta para el grupo de hilos de las Pruebas de Estabilidad con Apache JMeter.

### 6.3.1.3 Pruebas de Seguridad

Para la ejecución de estas pruebas se utilizó el grupo de hilos que contiene las peticiones de las 6 vistas del MGB, desactivando las cookies para que no se envíe el token de sesión y poder validar si es posible acceder a alguna de las vistas del módulo sin haber iniciado sesión. Todas las peticiones responden correctamente con el código 301 que indica el proceso de redirección a la vista de inicio de sesión en 3 fases, siendo el último la respuesta de la vista de inicio de sesión del SIAAF [véase Figura 17].

Se ejecutaron un total de 13 pruebas de seguridad como se muestra en la Tabla 8.

Tabla 8. Métrica recolectada durante la ejecución de las pruebas de seguridad.

Número de casos de prueba ejecutados:	13
Número de incidentes:	0
Número de casos de prueba re-ejecutados:	6
Número de incidentes resueltos:	0

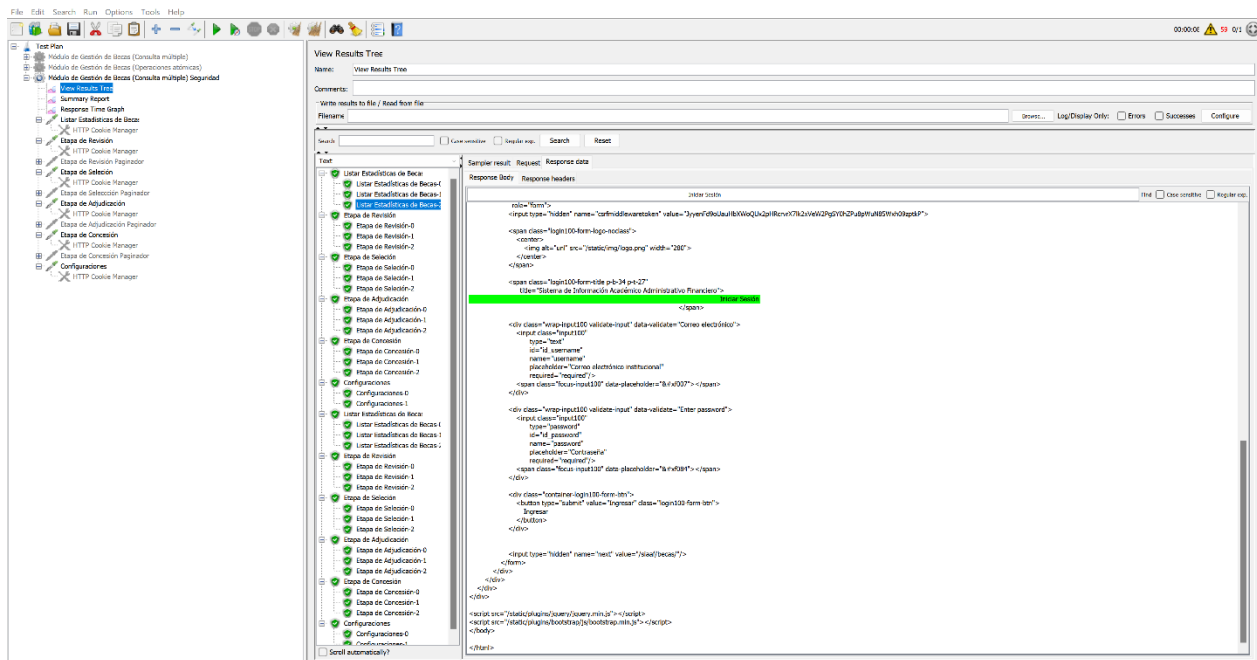


Figura 17. Respuesta de redireccionamiento de la vista listar Estadísticas Becas al intentar acceder sin token de sesión.

Para validar los permisos de acceso que tienen los diferentes grupos del módulo se intentó acceder a vistas en las que cada uno de los usuarios no debería poder acceder ya que no tienen los permisos necesarios [véase Figura 18]. Para una revisión completa en la ejecución de estas pruebas, véase Ejecución del Plan de Pruebas para el MGB - Pruebas de Seguridad.

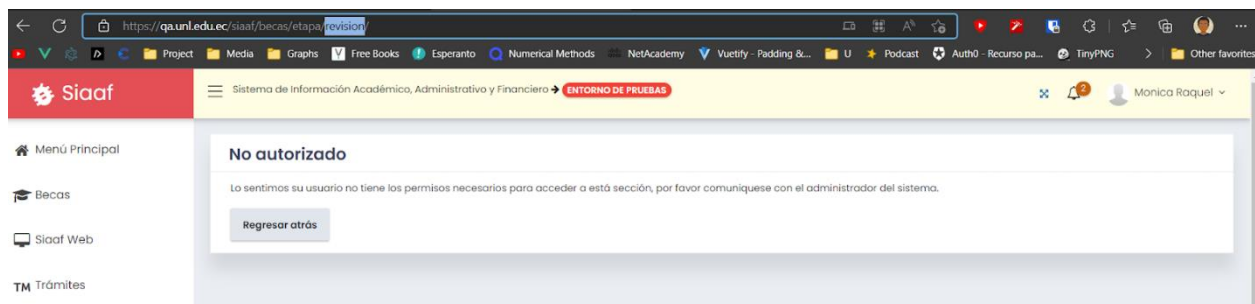


Figura 18. Vista que muestra el acceso no autorizado cuando un usuario intenta acceder a un apartado al que no tiene permisos.

### 6.3.1.4 Pruebas Unitarias

En la ejecución de las Pruebas Unitarias se utilizó la librería **SimpleTestCase** incluida en Django que permite crear escenarios de prueba del código, para simular que las distintas combinaciones de entrada produzcan una salida deseada y corregir errores en caso de ser necesario. En esta validación se consideraron aquellas funcionalidades que son más críticas para el módulo, es decir, que de fallar podría producir errores del uso de varias características. Se ejecutaron un total de 30 pruebas unitarias como se muestra en la Tabla 9.

Tabla 9. Métrica recolectada durante la ejecución de las pruebas unitarias.

<b>Número de casos de prueba ejecutados:</b>	<b>30</b>
<b>Número de incidentes:</b>	5
<b>Número de casos de prueba re-ejecutados:</b>	19
<b>Número de incidentes resueltos:</b>	5

El código para las Pruebas Unitarias del módulo de Gestión de Becas fue organizado de acuerdo a las capas más importantes del módulo, dentro de un directorio tests en la aplicación de becas [véase Figura 19].

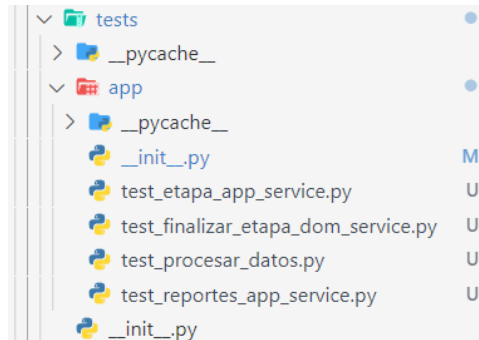


Figura 19. Estructura del código de Pruebas Unitarias para el Módulo de Gestión de Becas.

Para validar las distintas funciones que tiene el MGB se crearon Pruebas Unitarias específicas para cada función con algunas de las variaciones que se pueden presentar; por ejemplo, para el caso de la función que obtiene los meses de un periodo académico se crearon 2 funciones test, la primera valida que para los periodos ABR-SEP obtenga siempre los meses de mayo a agosto y la segunda valida que para los periodos OCT-ABR se obtengan los meses de octubre a marzo [véase Figura 20].

```
(siaafenv) jack@Legion:~/UNL/siaaf$ python manage.py test app.becas.tests.app
.TestEtapaAppService
System check identified no issues (0 silenced).
Debería retornar los meses de Mayo a Agosto:
meses obtenidos: ['may', 'jun', 'jul', 'ago']
meses esperados: ['may', 'jun', 'jul', 'ago']
.Debería retornar los meses de Octubre a Marzo:
meses obtenidos: ['oct', 'nov', 'dic', 'ene', 'feb', 'mar']
meses esperados: ['oct', 'nov', 'dic', 'ene', 'feb', 'mar']
-----
Ran 2 tests in 0.001s

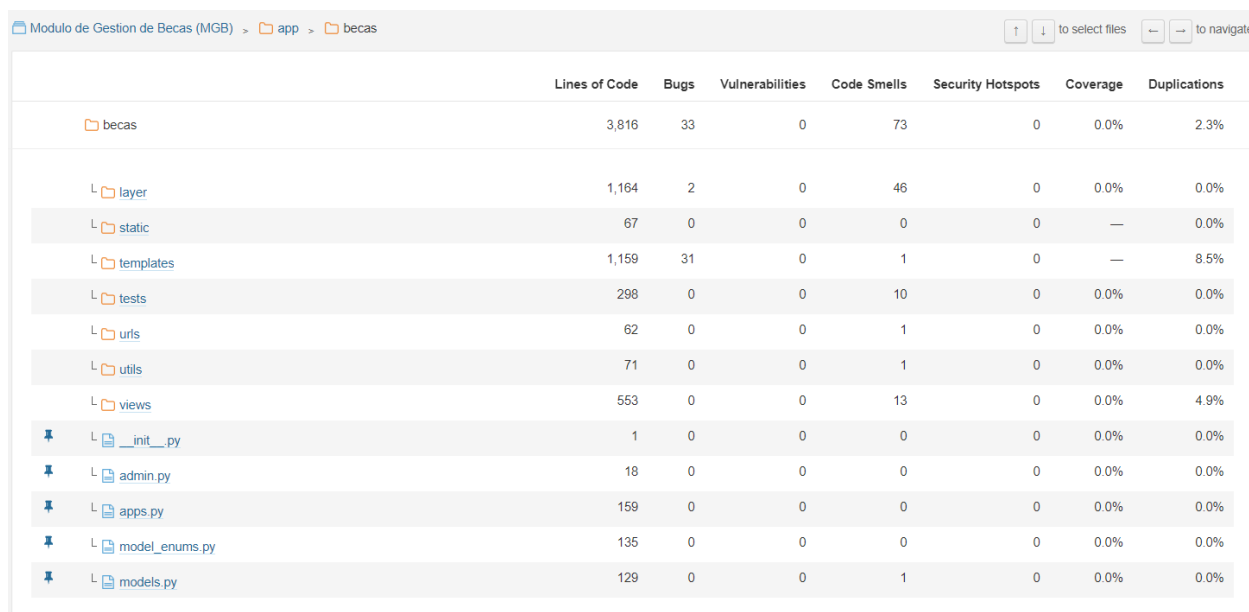
OK
(siaafenv) jack@Legion:~/UNL/siaaf$
```

Figura 20. Respuesta por consola de la ejecución de las Pruebas Unitarias al método para obtener meses de acuerdo a un periodo académico dado.

Cada Prueba Unitaria contiene impresiones por consola con una descripción del caso de prueba, así como del resultado obtenido y esperado como se aprecia en la Figura 20. Para una vista más detallada en el desarrollo de estas pruebas y los resultados obtenidos, véase Ejecución del Plan de Pruebas para el MGB sección Pruebas Unitarias.

### 6.3.1.5 Pruebas de Código

Para realizar estas pruebas se hizo uso de la herramienta SonarQube que permite analizar el código de un software. SonarQube se encarga de la búsqueda de errores complicados para evitar un comportamiento indefinido que afecten a los usuarios finales, detectar vulnerabilidades que comprometan a la aplicación, asegurar que el código se mantenga limpio y mantenible. El análisis arrojado por esta herramienta muestra que el MGB cuenta con cero vulnerabilidades detectadas y cero puntos de acceso que comprometan la seguridad. En cuanto a *code smells* que es un término del inglés para referirse a esas porciones de código que funcionan pero que impactan a primera vista; el MGB tiene 73 de ellos. Por último, en cuanto a duplicidad de código el MGB representa un 2.3% de código repetido de entre todos los módulos existentes en el SIAAF [véase Figura 21].



	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
becas	3,816	33	0	73	0	0.0%	2.3%
layer	1,164	2	0	46	0	0.0%	0.0%
static	67	0	0	0	0	—	0.0%
templates	1,159	31	0	1	0	—	8.5%
tests	298	0	0	10	0	0.0%	0.0%
urls	62	0	0	1	0	0.0%	0.0%
utils	71	0	0	1	0	0.0%	0.0%
views	553	0	0	13	0	0.0%	4.9%
__init__.py	1	0	0	0	0	0.0%	0.0%
admin.py	18	0	0	0	0	0.0%	0.0%
apps.py	159	0	0	0	0	0.0%	0.0%
model_enums.py	135	0	0	0	0	0.0%	0.0%
models.py	129	0	0	1	0	0.0%	0.0%

Figura 21. Resultados del análisis de código por SonarQube para el MGB.

Una de las recomendaciones sugeridas por la herramienta fue refactorizar el código de la finalización de Etapas para reducir la complejidad cognitiva [véase Figura 22], con el fin de hacer el código más legible y comprensible para los desarrolladores. Para ello se trasladó toda la lógica de la vista a la capa de aplicación y dominio según corresponda, creando métodos más pequeños y que se puedan entender con facilidad gracias a la descripción de cada uno de ellos. Con esto se



crearon funciones específicas para cada acción haciendo que el código pase estar interconectado entre 3 archivos.

Se utilizó herencia de clases entre las capas de dominio y aplicación para heredar todos los atributos y métodos de la clase dominio (padre) a la clase aplicación (hijo) esto con la finalidad de acceder desde la capa aplicación a la capa de dominio y a su vez hacer un llamado desde la vista a la capa de aplicación que se encargará de gestionar toda la lógica para la Finalización de Etapas.

```

297 @login_required
298 @require_http_methods(["POST"])
299 def finalizar_etapa(request: HttpRequest):
300     try:
301         if request.method == 'POST':
302             # inicializar variables
303             data = json.load(request)['json_data']
304             etapa_codigo = data['etapa_codigo']
305             periodo_academico_id = data['periodo_academico']
306             # obtener etapa actual y siguiente etapa
307             etapa = Etapa.objects.get(
308                 codigo=etapa_codigo)
309             siguiente_etapa = Etapa.objects.get(
310                 codigo=etapa_codigo_siguiente_etapa)
311             (finalizada, mensaje, estado) = FinalizarEtapaDomService.
312             comprobar_estado_etapa(
313                 etapa, periodo_academico_id)
314             if finalizada:
315                 return get_response(mensaje=mensaje, status=estado)
316             # obtener becas del presente periodo academico etapa
317             becas_periodo_academico_etapa = BecaPeriodoAcademicoEtapa.objects.filter(
318                 etapa_id=etapa.id, periodo_academico_id=int(periodo_academico_id))
319             # crear registros en la siguiente etapa
320             listado_becas_siguiente_etapa = []
321             for beca_etapa in becas_periodo_academico_etapa:
322                 (no_permitido, mensaje, estado) = FinalizarEtapaDomService.
323                 comprobar_estado_permitido(
324                     beca_etapa.estado)
325                 if no_permitido:
326                     return get_response(mensaje=mensaje, status=estado)
327                 esta_aprobada = FinalizarEtapaDomService.beca_esta_aprobada(
328                     beca_etapa, etapa)
329                 if esta_aprobada:
330                     listado_becas_siguiente_etapa.append(
331                         FinalizarEtapaDomService.crear_beca_siguiente_etapa
332                         (beca_etapa, siguiente_etapa))
333             with transaction.atomic():
334                 # guardar registros en la siguiente etapa
335                 lista_becas_periodo_academico_nueva_etapa =
336                 BecaPeriodoAcademicoEtapa.objects.bulk_create(
337                     listado_becas_siguiente_etapa)
338                 # guardar notificaciones si se finalizo la etapa de seleccion
339                 if siguiente_etapa.codigo == Constantes.ETAPA_ADJUDICACION.value:
340                     EtapasAppService.notificar_nuevos_becados(
341                         lista_becas_periodo_academico_nueva_etapa)
342             (finalizada, mensaje, estado) = FinalizarEtapaDomService.
343             finalizar_etapa(
344                 etapa.id, periodo_academico_id)
345             if finalizada:
346                 return get_response(mensaje=mensaje, status=estado)
347             return JsonResponse("No se pudo finalizar la Etapa")
348         except Exception as e:
349             return get_response(mensaje=e.args[0], status=400)
350
298 @login_required
299 @require_http_methods(["POST"])
300 def finalizar_etapa(request: HttpRequest):
301     try:
302         if request.method == 'POST':
303             data = json.load(request)['json_data']
304             etapa_codigo = data['etapa_codigo']
305             etapa = Etapa.objects.get(codigo=etapa_codigo)
306             periodo_academico_id = int(data['periodo_academico'])
307             finalizar_etapa_app_service = FinalizarEtapasAppService(
308                 etapa, periodo_academico_id)
309             finalizar_etapa_app_service.set_siguiente_etapa()
310             finalizar_etapa_app_service.comprobar_estado_etapa()
311             finalizar_etapa_app_service.set_becas_periodo_academico_etapa()
312             finalizar_etapa_app_service.crear_becas_siguiente_etapa()
313             with transaction.atomic():
314                 finalizar_etapas_app_service.guardar_becas_siguiente_etapa()
315                 etapa_finalizada_ok = finalizar_etapas_app_service.finalizar_etapa()
316                 if etapa_finalizada_ok:
317                     return get_response(
318                         mensaje=FinalizarEtapasAppService.mensaje_etapa_finalizada,
319                         status=status.HTTP_200_OK)
320
321             return JsonResponse(FinalizarEtapasAppService.mensaje_etapa_no_finalizada)
322         except Exception as e:
323             return get_response(mensaje=e.args[0], status=status.HTTP_400_BAD_REQUEST)
324

```

Figura 22. Comparativa de cambios antes y después de aplicada la refactorización a la función de Finalizar Etapas para reducir su complejidad cognitiva.

Por último, cabe destacar que la herramienta SonarQube puntúa con una calificación A (mejor puntuación) en cuanto a mantenibilidad del código y a la seguridad evidenciando que no se han detectado vulnerabilidades graves que afecten a este apartado [véase Figura 23 y Figura 24].

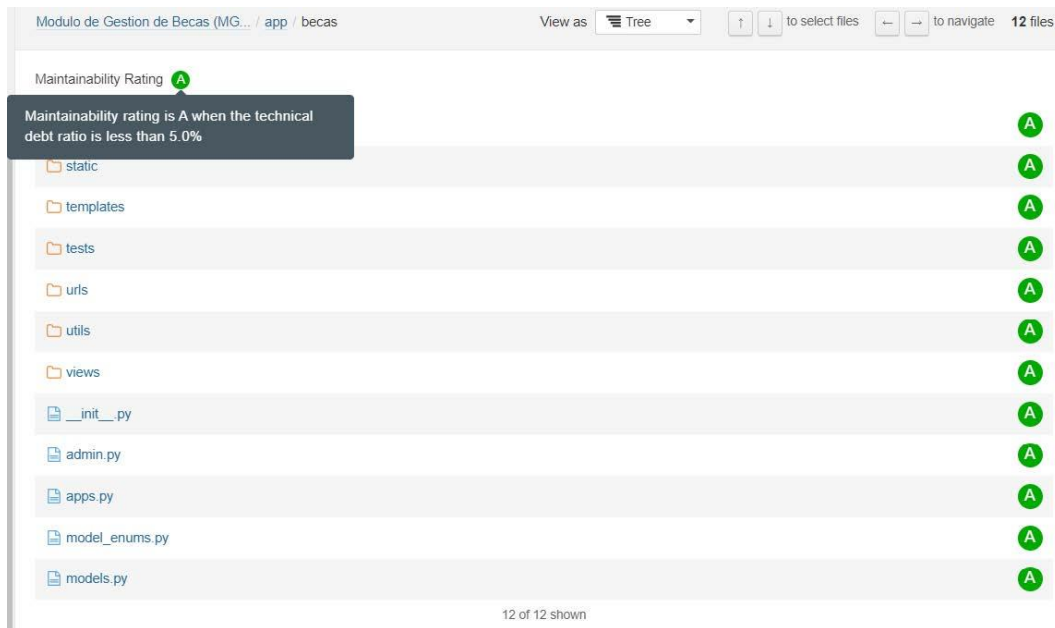


Figura 23. Calificación A (mejor puntuación) en la mantenibilidad del código para el MGB.

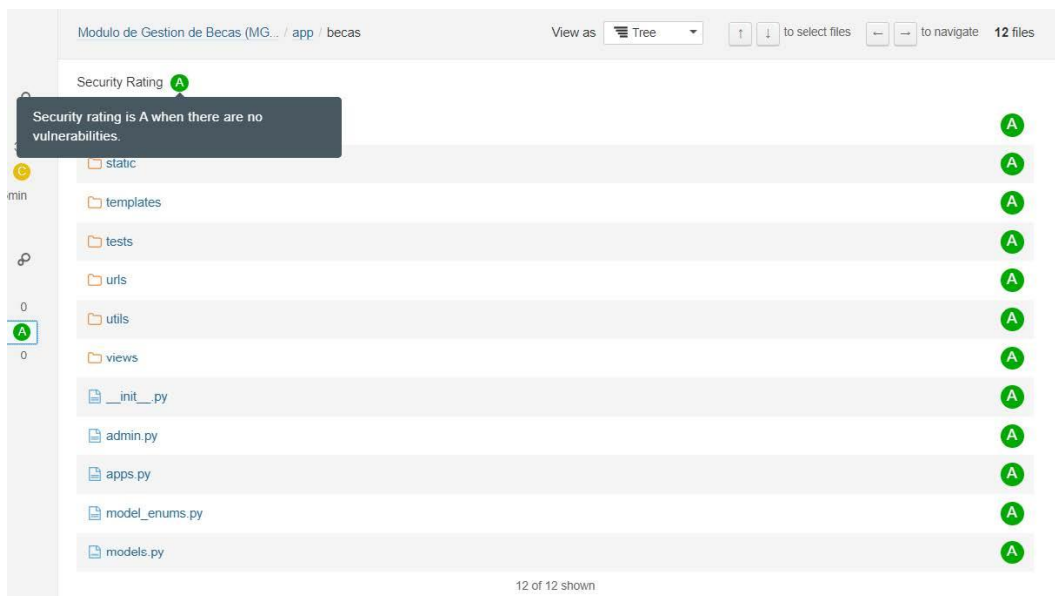


Figura 24. Calificación A (mejor puntuación) en la seguridad del código para el MGB.

### 6.3.1.6 Pruebas de Aceptación de Usuario

Las Pruebas de Aceptación del Usuario se llevaron a cabo en 3 fases:

1. Fase de Socialización del MGB con los Clientes
2. Fase de Pilotaje para los Clientes Directos

### 3. Fase de Aceptación del MGB por parte de los Clientes

**Fase de Socialización del MGB con los Clientes.** - En la Fase de Socialización del MGB se aplicó una encuesta a cada uno de los Clientes Directos para obtener información acerca del tiempo que les toma realizar algunos de los procesos sin el uso del módulo informático. Luego se procedió con la capacitación del personal indicando las funcionalidades y responsabilidades que tendría cada rol de usuario. Durante la capacitación se resolvieron las dudas que fueron surgiendo con respecto al uso del módulo informático y posteriormente se abrió un foro de preguntas y respuestas para resolver dudas específicas, una vez cumplido con la capacitación se firmó el acta de asistencia y se conversó sobre la Fase de Pilotaje.

**Fase de Pilotaje para los Clientes Directos.** - La Fase de Pilotaje consistió en un periodo de tiempo en el que los Clientes Directos pudieron probar y validar las funcionalidades desarrolladas del módulo, fase que estuvo habilitada durante 10 días. Esta fase fue necesaria para comprobar que cada usuario del módulo pueda realizar las acciones de que le competen con el uso del MGB. Durante esta fase se solicitó ayuda en el manejo del MGB por parte del usuario de Tesorería Financiera y el Administrador de Becas. Para brindar soporte se organizaron reuniones de asistencia virtual guiada a cada usuario en la Gestión de Becas según las funciones correspondientes.

**Fase de Aceptación del MGB por parte de los Clientes.** - La Fase de Aceptación consistió en una reunión con los Clientes, y de acuerdo a la agenda prevista se aplicó 2 encuestas, la primera acerca del tiempo que les toma realizar los mismos procesos que se consultaron durante la Socialización, pero esta vez enfocada al uso del MGB, que sirvió para comparar los resultados obtenidos antes y después del uso del MGB. La segunda encuesta por su parte, estuvo enfocada en determinar el cumplimiento de los Requisitos de Software tanto funcionales como no funcionales para conocer el grado de satisfacción del MGB por parte de los usuarios.

El cumplimiento de los requisitos funcionales fue evaluado por parte de los Clientes Directos, mientras que los requisitos no funcionales fueron evaluados por los Clientes Indirectos ya que involucra conocimiento técnico para su correcta evaluación. Para finalizar la fase todos los Clientes firmaron el acta de Aceptación del MGB con los acuerdos y compromisos a los que se

llegaron [véase Anexo 7 – Pruebas de Aceptación]. Los resultados recopilados durante esta fase son visibles en la tabla de resultados de evaluación basados en los Requisitos Funcionales [véase Tabla 10]. La tabla de resultados muestra el número identificador de Requisito Funcional para su comparación con el documento de la ERS.

*Tabla 10. Resultados de la evaluación del MGB en base a los Requisitos Funcionales capturados durante la fase de análisis.*

Identificador	Nombre	Evaluación		
		No Cumple	En parte	Cumple
RF01	Sincronizar datos de postulación desde el MGT hacia el MGB			X
RF02	Actualizar estados de postulaciones			X
RF03	Descargar documentos de postulación			X
RF04	Ver trámite en el MGT			X
RF05	Finalizar Etapas			X
RF06	Actualizar estados de Becas			X
RF07	Visualizar Postulaciones y Becas			X
RF08	Generar Reportes			X
RF09	Agregar mensaje frecuente			X
RF10	Actualizar mensaje frecuente			X
RF11	Eliminar mensaje frecuente			X
RF12	Marcar como favorito mensaje frecuente			X
RF13	Actualizar montos de pago para los tipos de beca			X
RF14	Visualizar estadísticas generales de becados			X
RF15	Iniciar Sesión			X
RF16	Notificar resultados de postulación			X
RF17	Guardar registro de becado en el sistema financiero			X

Como parte de la evaluación para determinar si el MGB representa una mejora significativa en cuanto al desempeño de las actividades para los usuarios del módulo, se consultó acerca de determinadas tareas que cada usuario debe realizar durante la Gestión de Becas, para poder medir la diferencia de tiempo existente en desarrollar las mismas actividades sin el uso del módulo y con el uso del mismo. Los gráficos que se muestran a continuación forman parte del análisis de las encuestas aplicadas a cada uno de los usuarios del MGB [véase Anexo 12]. En el caso del usuario Administrador de Becas, principal interesado en el desarrollo del módulo, la percepción de reducción de tiempo es considerablemente alta [véase Figura 25], esto se debe a que el MGB está enfocado en cubrir gran parte de la Gestión de Becas para el Administrador de Becas como actor principal. Es verdad, que en un escenario real con información verdadera las expectativas puedan llegar a reducirse, sin embargo, esto indica que el módulo desarrollado representa positivamente una mejora significativa al desarrollo de los procesos en contraste con prescindir de su uso.

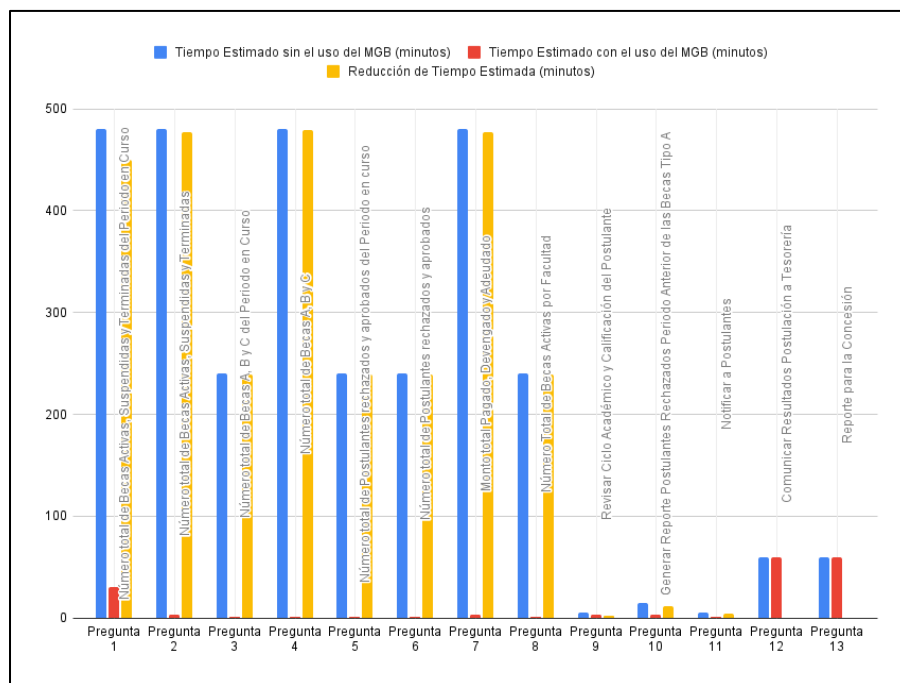


Figura 25. Gráfico que muestra la percepción que tiene el usuario Administrador de Becas con respecto a la reducción de tiempo con el uso del MGB.

En cuanto al usuario de Contabilidad Financiera, los resultados indican que existe una gran expectativa en cuanto a la reducción de tiempo empleado para los procesos consultados en un escenario real [véase Figura 26]. De igual manera para el usuario de Tesorería Financiera se puede evidenciar que el módulo representa una mejora sustancial como ayuda en el desarrollo de sus actividades para la Gestión de Becas [véase Figura 27].

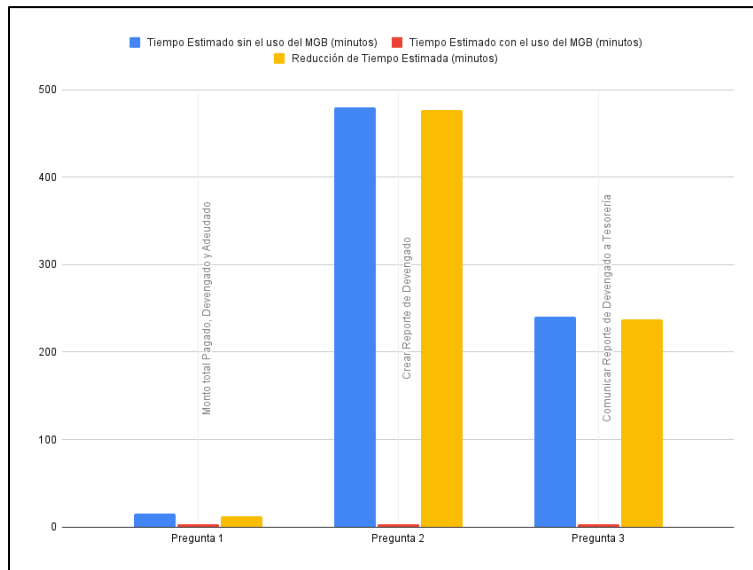


Figura 26. Gráfico que muestra la percepción que tiene el usuario de Contabilidad Financiera con respecto a la reducción de tiempo con el uso del MGB.

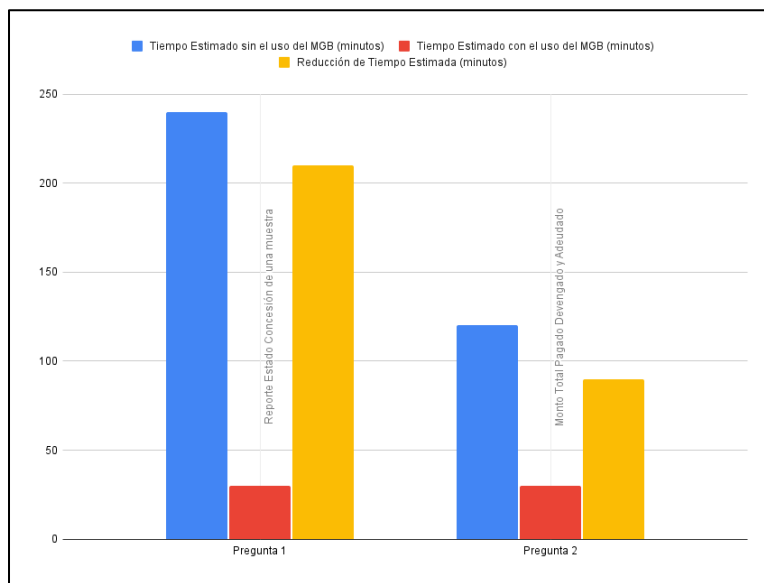


Figura 27. Gráfico que muestra la percepción que tiene el usuario de Tesorería Financiera con respecto a la reducción de tiempo con el uso del MGB.

Para finalizar la encuesta de Aceptación del Usuario se solicitó una calificación general de las funcionalidades más importantes para cada usuario con el fin de conocer la percepción que los usuarios tenían con respecto al uso del MGB. Los resultados muestran que los 3 usuarios del módulo se encuentran muy satisfechos con el trabajo realizado [véase Figura 28].

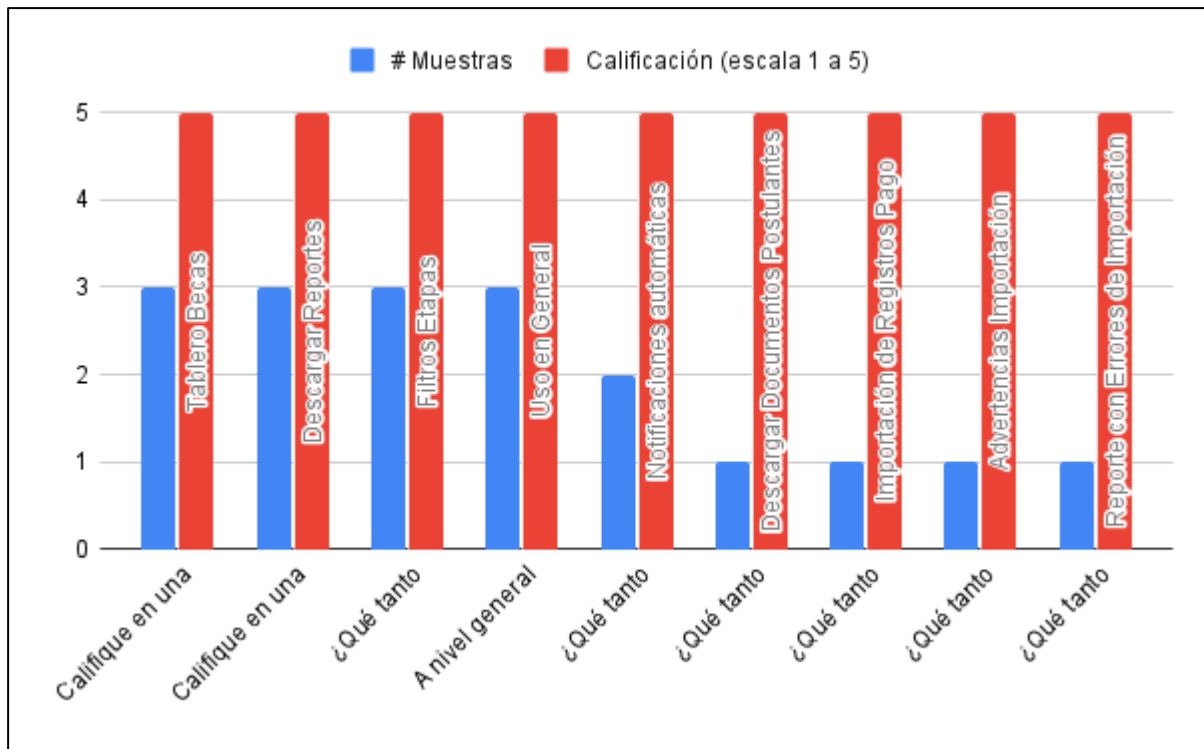


Figura 28. Gráfico que muestra la calificación proporcionada por los Usuarios con respecto al uso del MGB.

### 6.3.2 Elaboración de la documentación necesaria para la implantación y uso del módulo desarrollado

#### 6.3.2.1 Manual de Usuario

Se elaboró un Manual de Usuario en base al formato establecido por DTI siguiendo el esquema de organización propuesto. Este documento se elaboró con la finalidad de que cada uno de los usuarios finales tenga una fuente de consulta primaria en caso de dudas al ejecutar determinadas acciones dentro del MGB. Además, es útil en un futuro para nuevos funcionarios que asumen el rol en cualquiera de los procesos que intervienen en la Gestión de Becas. Para conocer acerca del

manual de usuario, véase el Anexo 4.



Dirección de  
Tecnologías de Información

[MANUAL DE USUARIO]

<<DIRECCIÓN DE BIENESTAR UNIVERSITARIO>>  
<<SUBDIRECCIÓN DE CONTABILIDAD FINANCIERA>>  
<<SUBDIRECCIÓN DE TESORERÍA FINANCIERA>>

## MÓDULO DE GESTIÓN DE BECAS - SIAAF

Versión 22.08

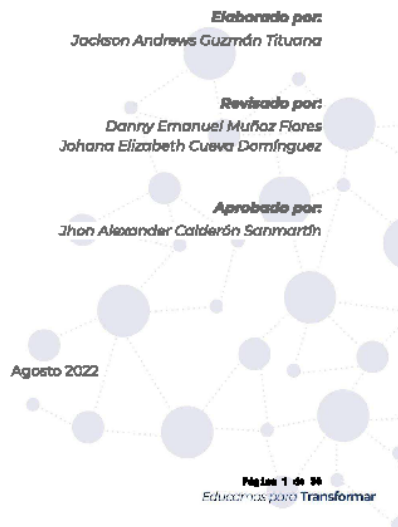


Figura 29. Portada del Manual de Usuario para el uso del MGB.

### 6.3.2.2 Manual Técnico

Como parte de la documentación necesaria para la implantación del módulo se elaboró un Manual Técnico para centralizar la información relacionada con la configuración del módulo para su puesta en marcha en el ambiente de producción. Este documento contempla las inserciones necesarias de registros a la base de datos, la asignación de roles de usuario y la administración de los ciclos máximos de concesión de becas para cada programa de estudio [véase Anexo 5].



[MANUAL TÉCNICO]

<<DIRECCIÓN DE TECNOLOGÍAS DE INFORMACIÓN>>

## MÓDULO DE GESTIÓN DE BECAS - SIAAF

Versión 22.08

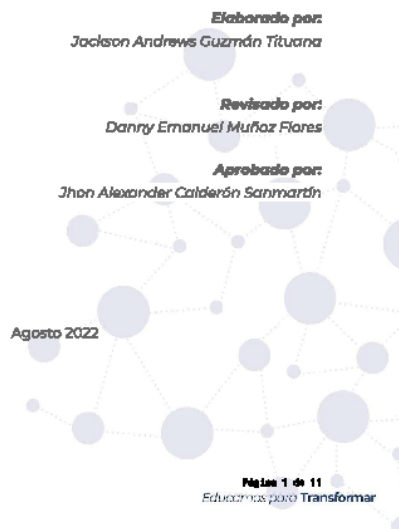


Figura 30. Portada del Manual Técnico para la configuración del MGB.

### 6.3.3 Certificación de Control de Calidad emitido por la DTI

Toda la documentación presentada que incluye al Plan de Pruebas y la evidencia presentada en la Ejecución del Plan de Pruebas, así como el Manual de Usuario y Técnico fueron revisados por la DTI para validar que se dio cumplimiento con la ejecución del Proyecto certificándose que el MGB queda listo para su puesta en producción por parte de DTI [véase Anexo 8].



**Informe Nro. UNL-DTI-2022-126**

<b>Para:</b>	Ing. Jhon Calderón <b>DIRECTOR DE TECNOLOGÍAS DE INFORMACIÓN</b> email: <a href="mailto:direccion.ti@unl.edu.ec">direccion.ti@unl.edu.ec</a>
<b>De:</b>	Ing. Johana Cueva <b>ANALISTA DE SISTEMAS INFORMÁTICOS I</b>
<b>Asunto:</b>	<b>Certificación de Control de Calidad:</b> Proyecto 2020-007 - Implementación del Módulo de Gestión de Becas para Bienestar Universitario - SIAAF
<b>Referencia:</b>	<a href="#">27282</a> - SE-No.02-ROCS-No-05-08-02-2021 (Reglamento de Becas)
<b>Tickets - Oficina No</b>	<a href="#">33588, 33579</a> - Certificado para realizar el módulo de Becas <a href="#">37282</a> - Revisión del documento de ERS para el MGB <a href="#">38888</a> - Registro de pago de becas <a href="#">42168</a> - Conexión MGT con MGB
<b>Fecha:</b>	Loja, 02 de septiembre del 2022

**ANTECEDENTES**

La implementación de los requerimientos del Módulo de Gestión de Becas para Bienestar Universitario - SIAAF, se lleva a cabo considerando los siguientes antecedentes:

- Mediante ticket de la Mesa de Servicios Nro. 27626 con Of. Nro. 09-SG-OCS-UNL se hace referencia a la Resolución SE-No.02-ROCS-No-05-08-02-2021, adoptada en la Segunda Sesión Extraordinaria del Órgano Colegiado Superior de la Universidad Nacional de Loja (OCS), desarrollada el 8 de Febrero de 2021, mediante la cual resolvió en lo principal aprobar el "REGLAMENTO DE LA SECCIÓN DE BECAS E INCENTIVOS DE BIENESTAR UNIVERSITARIO DE LA UNIVERSIDAD NACIONAL DE LOJA".
- Mediante ticket de la Mesa de Servicios Nro. 33588 y 33579 en el que consta, de fecha 20 de septiembre del 2021, suscrito por el Sr. Jackson Guzmán estudiante de la Carrera de Ingeniería en Sistemas, donde solicita:
 

si "¿... la emisión de un certificado que avale la solicitud por mi parte de la Unidad de Bienestar Universitario del módulo de becas.¿...?"
- Mediante ticket de la Mesa de Servicios Nro. 37202 en el que consta, de fecha 23 de febrero del 2022, suscrito por el Sr. Jackson Guzmán estudiante de la Carrera de Ingeniería en Sistemas, donde solicita:

Figura 31. Portada del informe de certificación emitido por DTI.

## 7. Discusión

El presente TT basa su justificación en la necesidad de poseer una plataforma tecnológica adaptada a los requerimientos de la UBU para la Gestión de Becas de los estudiantes de la UNL tal como lo manifiesta el Órgano Colegiado Superior y la Dirección de Bienestar Universitario en el Reglamento de Becas vigente, de acuerdo al certificado emitido por la DTI [véase Anexo 16] con el propósito de facilitar el manejo de la información de Postulantes y Becados a través de las distintas Etapas por parte de los Gestores de Becas permitiendo organizar y clasificar la información de manera más eficiente. El resultado final de este TT es el módulo de software para la Gestión de Becas de la UBU según los requisitos establecidos [véase Anexo 3].

Mientras que en el TT *“Sistema Informático para la Gestión del Programa de Becas de ADHU para la Universidad de El Salvador”* se utiliza una adaptación del diagrama de actividades para el mapeo de los procesos de gestión de becas [17, p. 42]; y en el TT *“An Online Scholarship Application System”* se utiliza un diagrama libre de secuencia de estados [16, p. 2] para explicar las operaciones que se llevan a cabo en la gestión de becas. Por su parte en el TT *“Diseño, desarrollo e implementación de un sistema de información web para la gestión de créditos, subsidios y becas del fondo para la educación técnica tecnología y superior de la secretaría de educación del municipio de Girardot para el año 2021”* [18] se presenta una descripción textual de los procesos que se llevan a cabo a través de la descripción del problema [18, p. 15]. En este TT se utiliza el estándar BPMN que está diseñado con el propósito de proveer toda la nomenclatura necesaria para el mapeo de la lógica de negocios [véase **¡Error! No se encuentra el origen de la referencia.**] a l poseer elementos específicos para representar distintos propósitos que suceden durante las interacciones de los involucrados en los procesos. La ventaja de usar BPMN con respecto a los diagramas de actividades, es que pueden ser fácilmente comprensibles por cualquier persona. A diferencia de los otros trabajos relacionados consultados [16] [17] [18], en este TT se presenta un diagrama general de los procesos para la Gestión de Becas [véase Figura 3] que vincula a cada una de las etapas. Además, partir desde un modelado de procesos como BPMN facilita identificar y agrupar los Requisitos de Software, así como comprender las diferentes actividades que intervienen en la lógica de negocio.

Los trabajos relacionados consultados [16] [18], no usaron el estándar IEEE 830 ERS para la

especificación de requisitos de software, y por su parte el TT [17], se enfocó únicamente en definir los requisitos de software. El documento de ERS constituye el documento formal en el que se establece el propósito, alcance, personal involucrado, así como los objetivos, las restricciones, suposiciones, dependencias y los requerimientos del producto de software; lo cual resulta imprescindible considerando el ámbito académico en el que se encuentra inmerso este TT en todo software a desarrollarse ya que establece la planificación del proyecto y sienta las reglas contractuales para alcanzar las expectativas acordadas sobre el software.

Los trabajos de titulación *“Sistema Informático para la Gestión del Programa de Becas de ADHU para la Universidad de El Salvador”* [17] y *“An Online Scholarship Application System”* [16] también presentan como resultado al Diagrama de Casos de Uso General, sin embargo, en este TT se utiliza la nomenclatura recomendada de manera más efectiva al utilizar varios de sus elementos y no solo aquellos elementos básicos.

En contraste con el TT *“Sistema Informático para la Gestión del Programa de Becas de ADHU para la Universidad de El Salvador”* [17] en donde por cada requisito funcional se presenta el código, el nombre, la descripción y el actor que interviene, en este TT se vincula su identificación con el caso de uso, se describen las precondiciones, el flujo principal y los flujos alternos. Además, se ofrece una relación directa con el prototipo visual de cada apartado del módulo que cumple con cada requisito de software [véase Anexo 3 – Requerimientos de alto nivel].

Durante la fase de Exploración y Planificación que establece la Metodología XP [revisar sección Marco Teórico, apartado 4.4.2] fue necesario crear y priorizar las Historias de Usuario, para ello aplicar la técnica del User Story Mapping [véase Anexo 10] permitió ganar tiempo al organizar cada historia de usuario en la épica correspondiente priorizando y versionando cada Sprint. En el TT *“Diseño, desarrollo e implementación de un sistema de información web para la gestión de créditos, subsidios y becas del fondo para la educación técnica tecnología y superior de la secretaría de educación del municipio de Girardot para el año 2021”* [18] se evidencia la utilización de la herramienta User Story Mapping [18, p. 63] al igual que en este TT con lo que se demuestra que representa una ventaja en la organización de las tareas de desarrollo que deben llevarse a cabo, permitiendo ordenar, priorizar y versionar el software. Esto facilita por su parte a una mejor descripción de las historias de usuario al dejar el panorama claro de las funcionalidades que se

requieren alcanzar.

La codificación del módulo informático se realizó siguiendo los patrones definidos para el SIAAF en cuanto a la manera de estructurar la aplicación, se siguió el esquema de organización Modelo, Vista, Plantilla (MVT) de acuerdo con la documentación de Django. Con respecto a los estándares de codificación se siguieron muchas de las recomendaciones descritas por Python y aplicadas por el equipo de desarrollo de DTI, como la nomenclatura y organización [véase sección Resultados, apartado 6.2.4].

En el TT *“Diseño, desarrollo e implementación de un sistema de información web para la gestión de créditos, subsidios y becas del fondo para la educación técnica tecnología y superior de la secretaría de educación del municipio de Girardot para el año 2021”* [18] se presentan pruebas unitarias, de recuperación y tolerancia a fallas, de integridad de datos y bases de datos, de seguridad y de usabilidad. Sin embargo, en las pruebas unitarias se evidencia que se hizo uso del software Postman, el cual no está diseñado para aplicar pruebas unitarias. Ya que las pruebas unitarias consisten en la validación de pequeñas porciones de código para garantizar que pese a las diferentes entradas siempre se obtenga una salida y no se interrumpa la ejecución del software [véase 0].

En el TT *“Sistema Informático para la Gestión del Programa de Becas de ADHU para la Universidad de El Salvador”* [17] se diseñaron pruebas unitarias, de integración, de sistema y de aceptación. Sin embargo, no se muestra evidencia de cómo se llevaron a cabo estas pruebas, ni que resultados se obtuvieron de cada una. Lo mismo se evidenció en el TT *“An Online Scholarship Application System”* [16]. Se presume que la evidencia no se adjuntó al no ser un requisito para las universidades en las que se presentaron estos TT.

A diferencia de los trabajos [16], [17] y [18], en este TT se ofrecen 6 tipos de pruebas de software para garantizar la calidad del mismo, cada una con su enfoque. En este TT se presentan Pruebas de Casos de Uso que validaron el cumplimiento de cada requisito de software [véase 6.3.1.1]. Pruebas de Carga y Estabilidad que tuvieron por objetivo garantizar que el módulo informático seguirá funcionando sin mayores complicaciones incluso en escenarios de sobre estrés. Además, se realizaron Pruebas de Seguridad que permitieron comprobar que los permisos de cada usuario

están correctamente descritos, evitando que los usuarios puedan realizar acciones que no les son permitidas para su rol de usuario [véase 6.3.1.3]. Pruebas Unitarias para asegurar que el software no detenga su funcionamiento en sus funcionalidades más críticas. Pruebas de Código que garantizan la calidad de codificación realizada y que el proyecto de software entregado a DTI puede ser mantenible y escalable para su continuo desarrollo en la integración de nuevas características. Y por último se presentaron también, Pruebas de Aceptación de Usuario que afirman la conformidad de los usuarios con el software desarrollado. Esto queda demostrado de igual forma con la calificación brindada por los usuarios a distintos apartados del módulo informático [véase Figura 28].

El MGB al ser un módulo integrado del SIAAF está sujeto a las políticas de privacidad de la DTI por ende el código fuente es de carácter privado y se mantiene seguro en el repositorio del software de control de versiones de software GitLab de la UNL.

## 8. Conclusiones

- El módulo de gestión de becas propuesto para su implementación, permitió automatizar los procesos manuales que se llevan a cabo en la Gestión de Becas de la Unidad de Bienestar Universitario de la Universidad Nacional de Loja.
- El modelado de los procesos que intervienen en la gestión de becas mediante diagramas BPMN, permitió identificar las etapas, los roles y los casos de uso necesarios a considerarse para el desarrollo del módulo.
- Con la implementación del módulo de gestión de becas propuesto, la información se mantiene siempre actualizada para todos los involucrados en la gestión de becas.
- El proceso aplicado para la codificación del módulo informático siguiendo la metodología XP bajo el marco de trabajo SCRUM, permitió combinar los procesos iterativos del software con la planificación de los entregables de cada sprint.
- La implementación del módulo para los procesos de seguimiento y control de Becas, fue validado en un entorno simulado con la ejecución de 18 pruebas de casos de uso que están avaladas por la certificación de la Dirección de Tecnologías de Información.
- Las pruebas de software aplicadas al Módulo de Gestión de Becas, demostraron que se mantiene robusto, seguro, mantenible, escalable y que cumple con los requerimientos establecidos en la Especificación de Requisitos de Software.
- El Módulo de Gestión de Becas representa una mejora significativa en cuanto al desempeño de las actividades para los usuarios del módulo, ya que se evidencia una sustancial reducción de tiempo en contraste con prescindir de su uso.
- La arquitectura del Módulo de Gestión de Becas, permite el desarrollo de mejoras y conexiones con otros módulos y sistemas externos que se requieran en los procesos para la gestión de las becas.

## 9. Recomendaciones

Las recomendaciones que se dan a continuación son generales y se basan en la experiencia del autor en el desarrollo de software.

- En la implementación de modelos de negocio que involucren la gestión de diversas Etapas, se recomienda el uso de diagramas BPMN para el modelado de procesos y subprocesos.
- Elaborar historias de usuario en entornos de desarrollo ágil como el que define la metodología XP, ya que permiten corroborar la relación entre los criterios de éxito, escenarios de prueba y la especificación de cada requisito para su validación.
- Aplicar la técnica User Story Mapping en conjunto con la metodología XP, para la organización de las Historias de Usuario, esto permitirá ganar tiempo al priorizar y versionar cada Sprint.
- Establecer patrones para la codificación en cuanto a la manera de estructurar la aplicación, nomenclatura y organización permitirá una mejor lectura y comprensión del código fuente para continuar con el desarrollo a futuro.
- Mantener una comunicación constante con el cliente para una evaluación continua del producto, así como celebrar dailys con el equipo de desarrollo para el seguimiento de las tareas.
- En todo tipo de software desarrollado convendría siempre contemplar un Plan de Pruebas, con el fin de asegurar el cumplimiento de los Requisitos de Software previstos durante la fase de análisis y asegurar la calidad del producto de software desarrollado.
- El montaje de un entorno de pruebas debería ser similar a la arquitectura del entorno de producción, para la Ejecución del Plan de Pruebas ya que esto permitirá validar correctamente la calidad del software desarrollado.

### 9.1 Trabajos futuros

- Implementar una interfaz de comunicación en el Sistema de Gestión Académica (SGA), para obtener los datos actualizados de los postulantes como: ciclo actual y pérdida de la gratuidad directamente en el Módulo de Gestión de Becas (MGB), para crear controles automatizados de suspensión y terminación de becas.



- Diseñar y desarrollar la funcionalidad para gestionar en el Módulo de Gestión de Becas (MGB), el presupuesto anual asignado para becas por cada carrera, para automatizar la asignación de becas.
- Para una comunicación directa en el marco de las becas, es necesario desarrollar un módulo que permita a los becados conocer el estado de su beca y administrar el pago de sus beneficios, así como recibir notificaciones y enviar la documentación que les sea requerida.

## 10. Bibliografía

- [1] I. Figueroa, P. Palacios y C. Ruiz, «Informe de Gestión DEL AVANCE DE LA UNIDAD DE BIENESTAR UNIVERSITARIO,» Loja, 2019.
- [2] B. Hitpass, «BPM: Business Process Management: Fundamentos y Conceptos de Implementación,» Disponible en <https://bit.ly/3DnUeCB>, 2017 (Accedido el 12/10/2021).
- [3] «Business Process Model and Notation (BPMN), Version 2.0,» Object Management Group, Disponible en <https://bit.ly/3S4OYrx>, 2011. (Accedido el 25/10/2021).
- [4] IEEE, «Especificación de Requisitos según el estándar de IEEE 830,» Disponible en <https://bit.ly/3ePYkJH>, octubre 2008. (Accedido el 07/11/2021).
- [5] J. Rumbaugh, I. Jacobson y G. Booch, «El Lenguaje Unificado de Modelado (UML) Manual de Referencia,» Disponible en <https://bit.ly/3QBpLEh>. (Accedido el 11/10/2021).
- [6] M. Rehkopf, «Historias de usuario | Ejemplos y plantilla | Atlassian,» Disponible en <https://bit.ly/3La25Wo>. (Accedido el 15/08/2021).
- [7] L. Calabria y P. Píriz, «Metodología XP,» *Universidad ORT Uruguay*, pp. 5-15, octubre 2003.
- [8] K. Schwaber y J. Sutherland, «*La Guía de Scrum La Guía Definitiva de Scrum: Las Reglas del Juego,*» Disponible en <https://bit.ly/3qyMFBF>, noviembre 2020. (Accedido el 08/15/2021).
- [9] K. Schwaber y J. Sutherland, «*La Guía de Scrum La Guía Definitiva de Scrum: Las Reglas del Juego,*» Disponible en <https://bit.ly/3qARlKV>, noviembre 2017. (Accedido el 08/15/2021).
- [10] I. Challenger, Y. Díaz y R. Becerra, «El lenguaje de programación Python,» Disponible en <https://bit.ly/3BBCh26>, junio 2014. (Accedido el 08/15/2021).
- [11] StackOverflow, Comunidad, «Aprendizaje Django eBook,» Disponible en <https://bit.ly/3UanMJM>. (Accedido el 11/19/2021).

- [12] C. J. M. Pérez, «POSTGRESQL SGBD LA ALTERNATIVA A LOS PROPIETARIOS,» Disponible en <https://bit.ly/3BEhafN>, octubre 2011. pp. 3-8. (Accedido el 11/22/2021).
- [13] C. B. Bueno, «Ingeniería del Software II - Construcción y Pruebas de Software,» *Universidad de Cantabria*, pp. 25-85.
- [14] A. Alain, J. Moore, P. Bourque y L. Tripp, «SWEBOK un proyecto del Comité de la Práctica profesional del IEEE computer society,» pp. 23-56, 2004.
- [15] W.-K. Shen, «An Online Scholarship Application System, » Disponible en <https://bit.ly/3V98O64>, 19 de enero 2011. (Accedido el 11/02/2021).
- [16] E. Orellana, F. Guardado, M. Orellana y J. Zelaya, «*Sistema Informático para la gestión del programa* de becas de la asociación para el desarrollo humano,» Disponible en <https://bit.ly/3eQ0s48>, 2019. (Accedido el 11/02/2021).
- [17] J. S. Triana Diaz y B. Z. Sánchez Martínez, «*Diseño, desarrollo e implementación de un sistema de información web para la Gestión de Créditos, Subsidios y Becas del Fondo para la Educación Técnica* Tecnología y Superior de educación del municipio de Girardot para el año 2021, 2021,» Disponible en <https://bit.ly/3G2omV1>, 2021. (Accedido el 29/05/2022).
- [18] Ó. C. Superior, «Reglamento de Becas, Loja, Ecuador: Universidad Nacional de Loja,» 2021.

## **11. Anexos**

### **Anexo 1. Desarrollo del Trabajo de Titulación**

## **1. Objetivo 1: Analizar los procesos que intervienen en la Gestión de Becas mediante el uso de BPMN**

### **1.1 Identificación de los procesos y subprocesos que intervienen en la Gestión de Becas**

En la UBU de la UNL mediante entrevistas con la Lcda. Marlene Valdez [véase Anexo 17] Responsable de la Sección de Becas e Incentivos de la UBU - UNL y el análisis del Reglamento de Becas vigente [véase Anexo 2], se identificaron las Etapas necesarias para la obtención de Becas por parte de los estudiantes.

La primer Etapa que consiste en la Postulación de Becas se realiza a través del MGT, en este módulo informático desarrollado por la DTI en la primera fase se receptan todos los requisitos necesarios para la postulación de acuerdo al Capítulo VI del Reglamento de Becas que habla sobre los requisitos para la Adjudicación de Becas [véase Anexo 2]. Una vez los postulantes envían toda la información requerida, la siguiente fase se lleva a cabo por la secretaria de Carrera a la que pertenece el Postulante, el secretario se encarga de adjuntar los certificados correspondientes de acuerdo al Reglamento de Becas [véase Anexo 2] y en la tercera fase interviene el secretario Abogado quien se encarga de legalizar los certificados de la fase anterior. Finalmente existe una última fase en la que se descarga el consolidado de documentos adjuntados por los Postulantes y el listado de los mismos mediante una hoja de cálculo para su Revisión y Validación.

En conjunto con la DTI de igual manera mediante una entrevista, se realizó el análisis de lo requerido para el MGB [véase Anexo 19]. Para el módulo informático propuesto se debería empezar por la Etapa de Revisión mostrando todos los postulantes que llegaron a fase 4 del MGT para lo cual se puede realizar la comunicación mediante Apis o la importación del listado de postulante por medio de una hoja de cálculo. En esta Etapa se evalúa y se asigna una calificación a la ficha socio-económica y se debe poder descargar la documentación adjuntada por cada uno de los Postulantes para que puedan ser aprobados o rechazados según sea el caso, indicando los motivos de dicha actualización.

Una vez finalizado el periodo de Postulación y Revisión de documentos, él/la Responsable de la Sección de Becas e Incentivos junto a él/la Trabajadora Social, como miembros del Comité realizan el análisis de la documentación, la situación socio-económica, la situación académica y criterios

de aplicación como la condición de vulnerabilidad y proximidad territorial, además se toma en consideración el presupuesto asignado y la disponibilidad de cupos por carrera, siendo actualmente del 10% de estudiantes matriculados en carreras presenciales [1, p. 10].

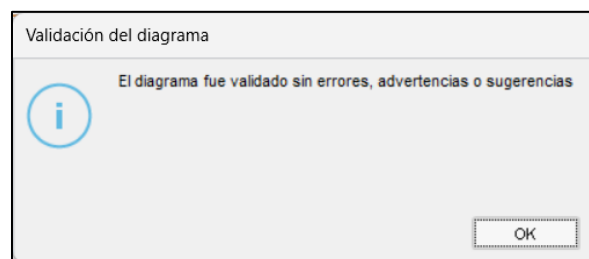
El resultado del proceso de Revisión, cumplimiento de requisitos y Selección, será notificado a los/las Postulantes, a través de su correo institucional. En la Etapa de Adjudicación el responsable de la Sección de Becas se encarga de Gestionar las Becas pudiendo Suspender y Terminar Becas, así como descargar los reportes que se consideren necesarios.

**(Véase Análisis del Reglamento de Becas)**

A continuación, se detalla los subprocesos que corresponden a cada una de las Etapas presentes en el diagrama BPMN general de Procesos para la Gestión de Becas.

### **1.2 Modelado de los procesos que intervienen en la Gestión de Becas con el uso de un software modelador de BPMN**

Una vez establecido e identificado cada uno de los procesos que se llevan a cabo en las Etapas para la Gestión de Becas, se procedió a modelarlos a través de diagramas BPMN que permitan visualizar el flujo y las relaciones existentes entre estos, así como el personal que interviene y las interacciones necesarias. Para el modelado de los procesos se hizo uso del software Bizagi Studio que además permitió validar el diseño elaborado garantizando que se usen de manera correcta cada uno de sus elementos.



*Figura 32. Dialogo que muestra que los diagramas BPMN desarrollados fueron validados exitosamente por Bizagi Studio.*

#### **1.2.1 Modelo BPMN general de procesos**

El modelado de estos procesos fue validado por parte de los interesados en el desarrollo (DTI y la

UBU) mediante una reunión de trabajo con sus representantes [véase Anexo 13 y Anexo 14]. Primeramente, se creó el diagrama general de Procesos para la Gestión de Becas en la que se muestra todas las Etapas involucradas y los Procesos que intervienen [véase Figura 33].

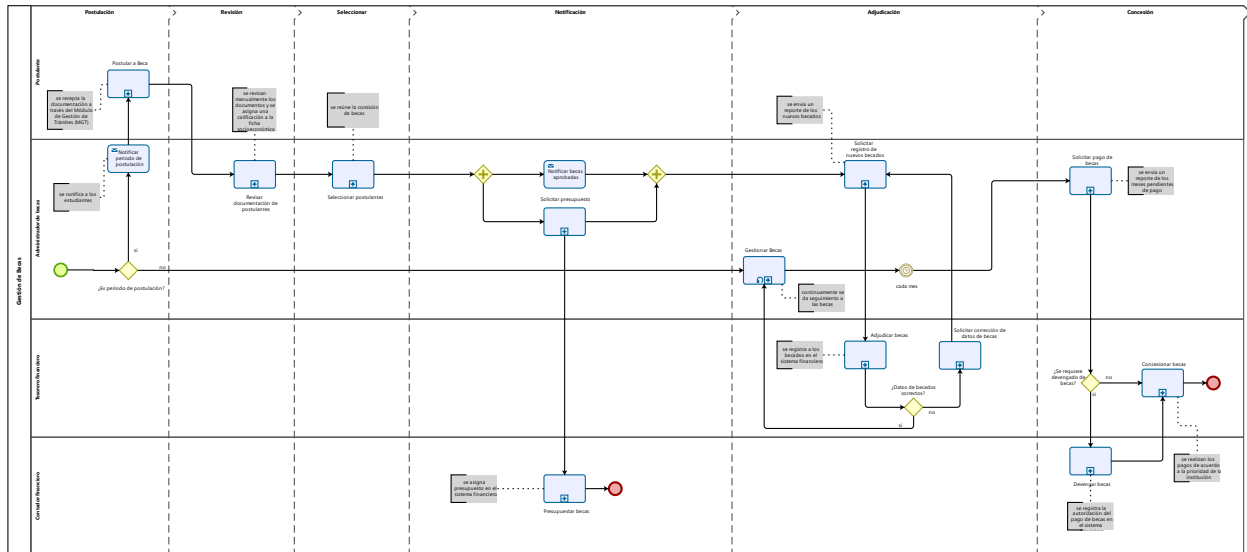


Figura 33. Modelado General de Procesos que intervienen en la Gestión de Becas.

### 1.2.2 Modelo BPMN de los subprocesos para la Etapa de Postulación

Los subprocesos que intervienen en esta Etapa son cubiertos por el MGT que actualmente se encuentra desplegado; ya se ha hecho uso de este módulo informático para la recepción de documentos por primera vez en el Periodo Lectivo de Postulación a Becas octubre de 2021. En la Figura 34 se muestra los subprocesos que intervienen en esta Etapa.

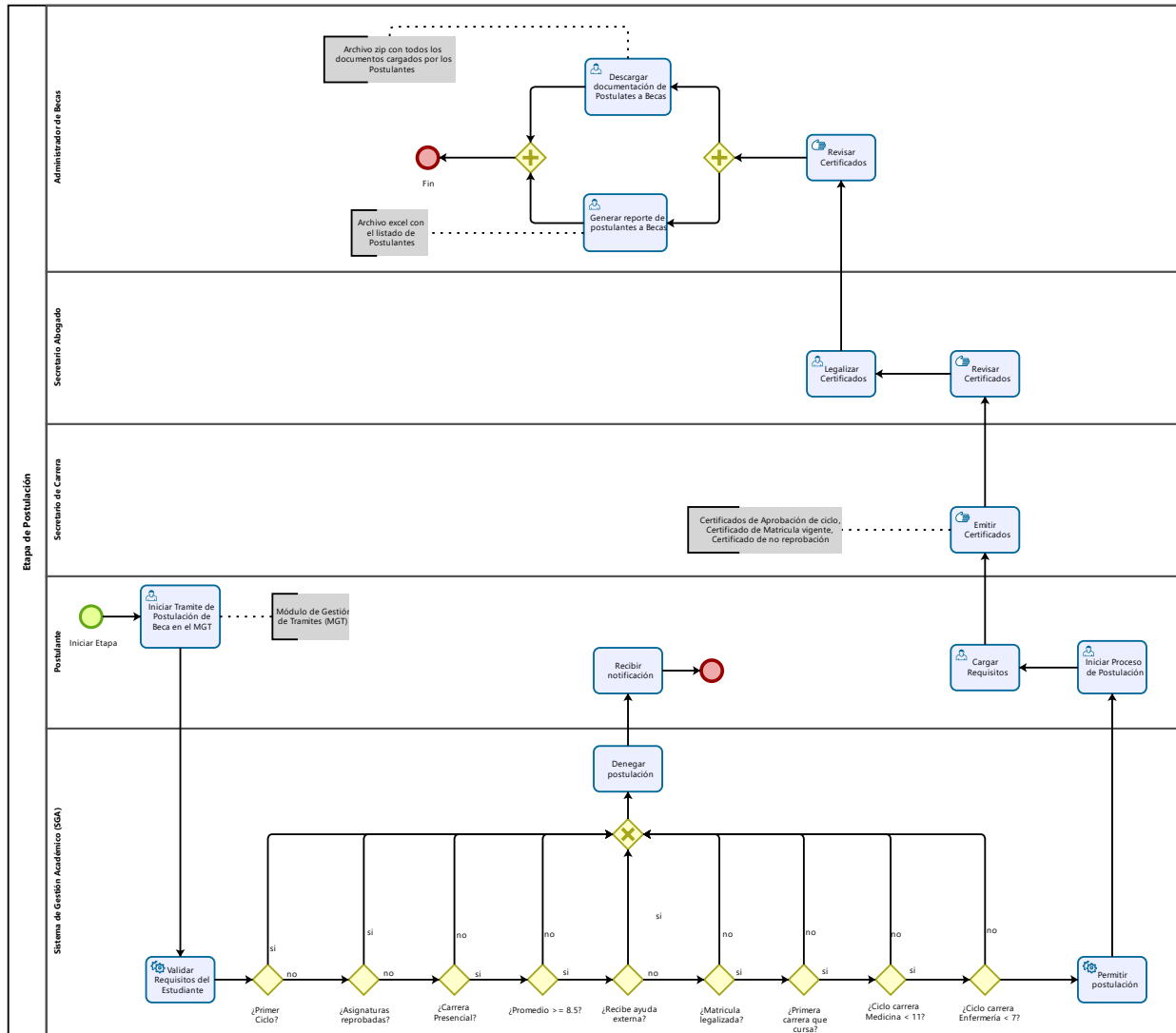


Figura 34. Diagrama BPMN - subprocesos para la Etapa de Postulación de Becas.

En adelante se describen los subprocesos de las Etapas que son cubiertas con el desarrollo del MGB.

### 1.2.3 Modelo BPMN de los subprocesos para la Etapa de Revisión

En esta Etapa el encargado de la Sección de Becas tiene la responsabilidad de revisar y validar la documentación, así como asignar una calificación a la ficha socio-económica de aquellos postulantes que previamente pasaron la Etapa de Postulación a través del MGT. En la *Figura 35* se muestran los subprocesos y actividades que se llevan a cabo en esta Etapa.



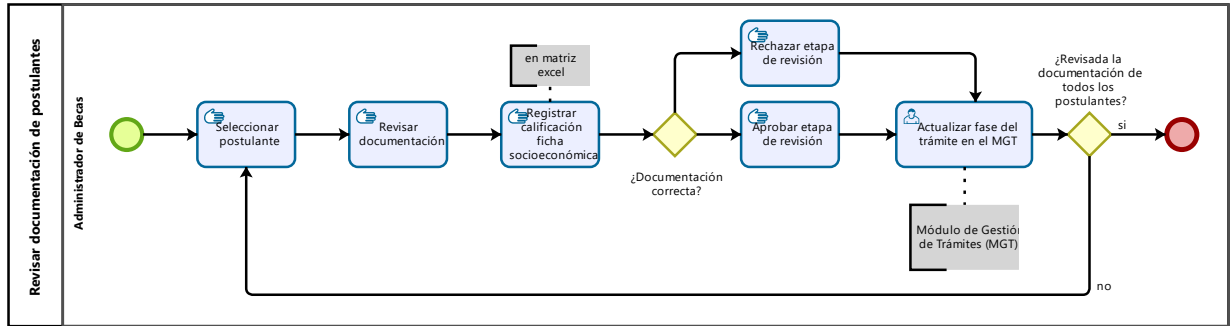


Figura 35. Diagrama BPMN - subprocesos para la Etapa de Revisión de Becas.

### 1.2.4 Modelo BPMN de los subprocesos para la Etapa de Selección

La etapa empieza por la selección de los postulantes que han aprobado la Etapa de Revisión, en esta etapa se realizan diversos tipos de análisis por parte de la Comisión de Becas, incluyendo el análisis de la documentación entregada, situación socio-económica, académica y criterios de aplicabilidad como la vulnerabilidad y proximidad territorial del postulante. En la Figura 36 se presenta el modelado de los procesos y actividades que se llevan a cabo en esta etapa.

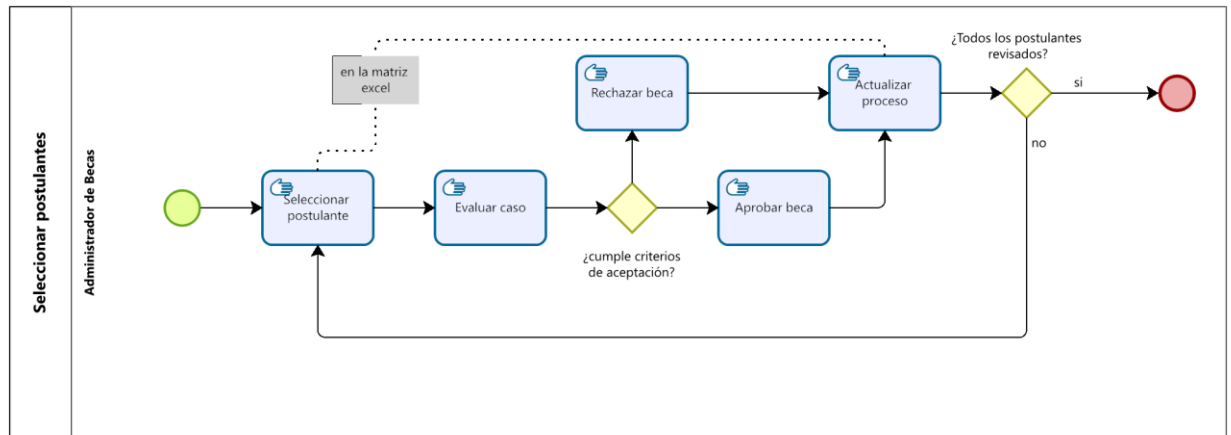


Figura 36. Diagrama BPMN - subproceso para la Etapa de Selección de Becas.

### 1.2.5 Modelo BPMN de los subprocesos para la Etapa de Notificación

Esta etapa consiste en notificar los resultados de postulación a los estudiantes para que conozcan si fueron o no aprobados, a Tesorería Financiera para el registro de nuevos becados y a Contabilidad Financiera para que pueda presupuestar becas para el año en curso [véase Figura 37].

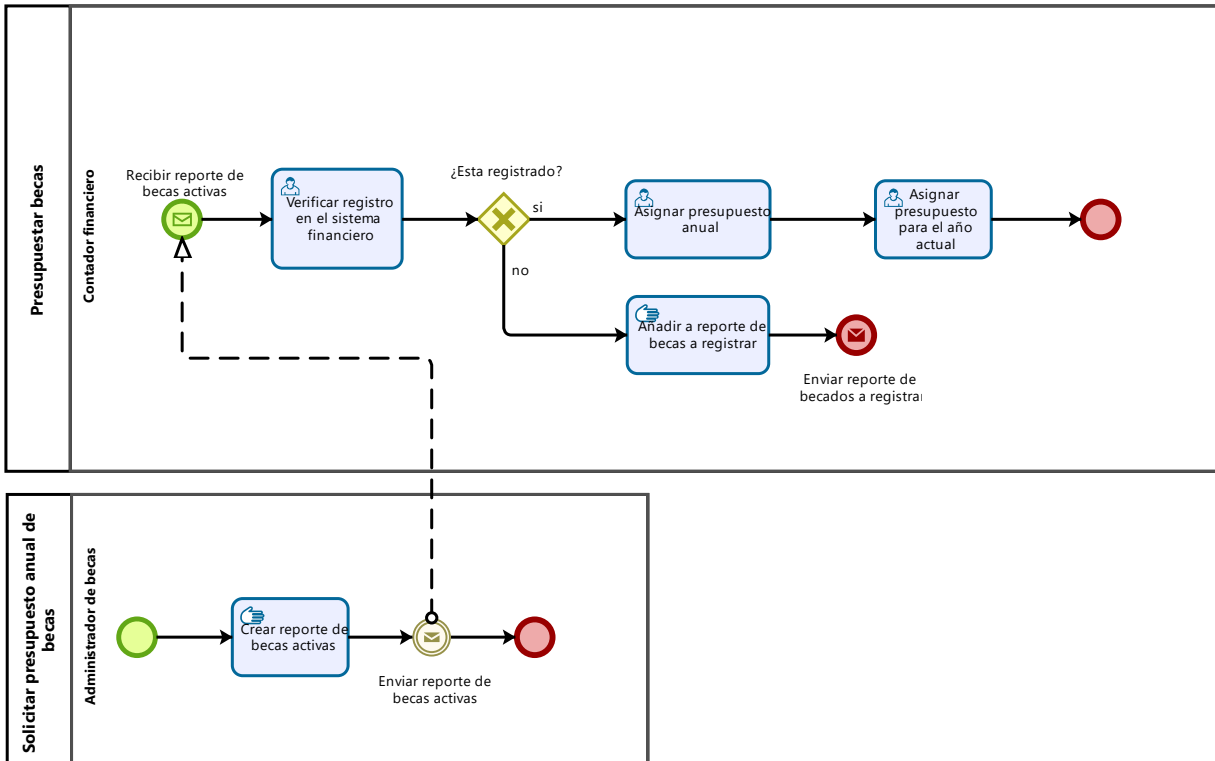


Figura 37. Diagrama BPMN - subprocesos en la Etapa de Notificación de Becas.

### 1.2.6 Modelo BPMN de los subprocesos para la Etapa de Adjudicación

Es la etapa que se encarga del seguimiento y control de becas por cada periodo académico. En esta etapa el encargado de la Sección de Becas puede suspender una beca por que el becado no cumple las obligaciones que se establecen en el Reglamento de Becas [véase Anexo 2 Capítulo X acerca de las obligaciones del estudiante becario], también es capaz de terminar una beca en caso que el becado haya egresado [véase Figura 38 y Figura 39].

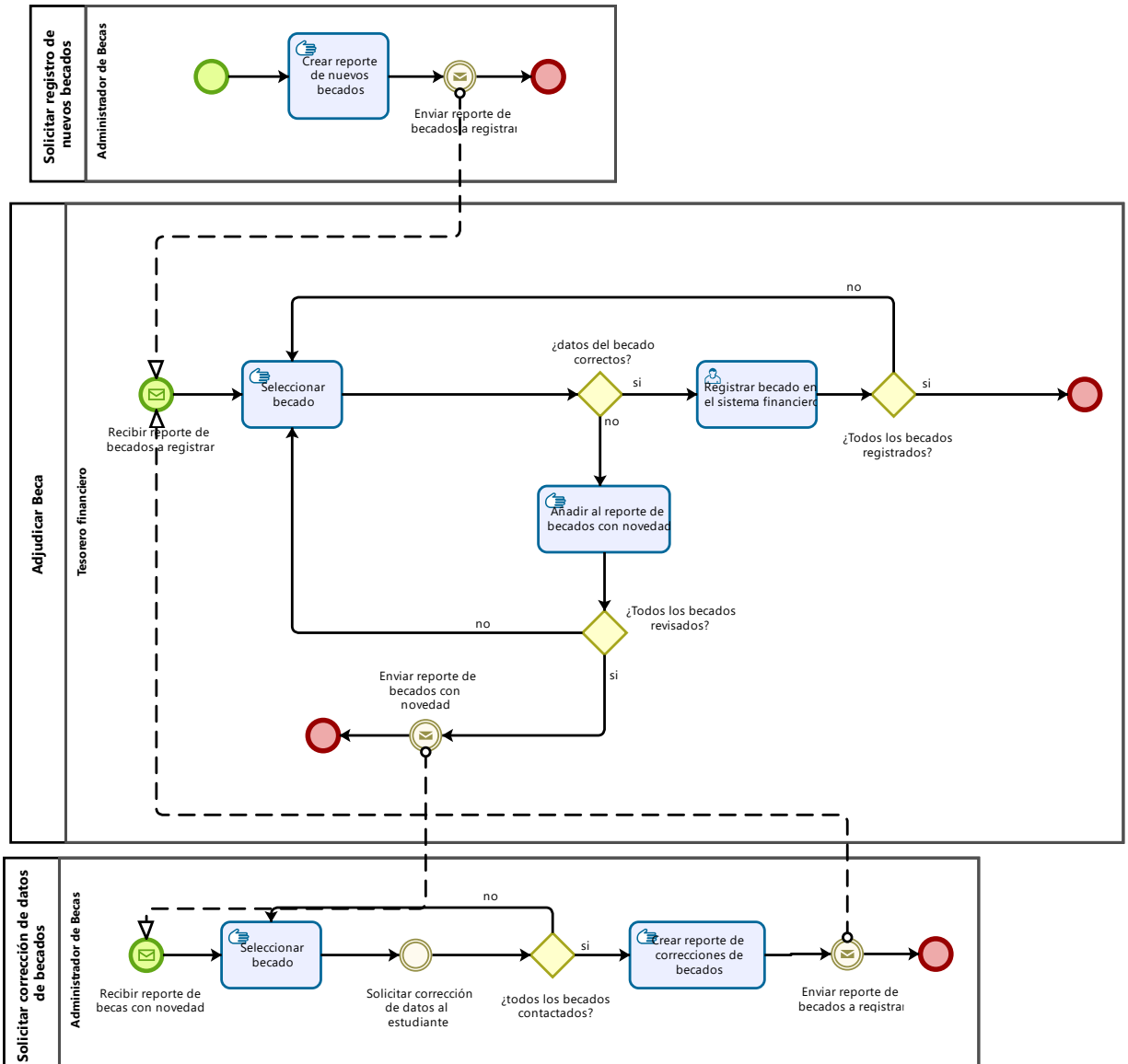


Figura 38. Diagrama BPMN - subprocesos para Adjudicar Becas en la Etapa de Adjudicación.

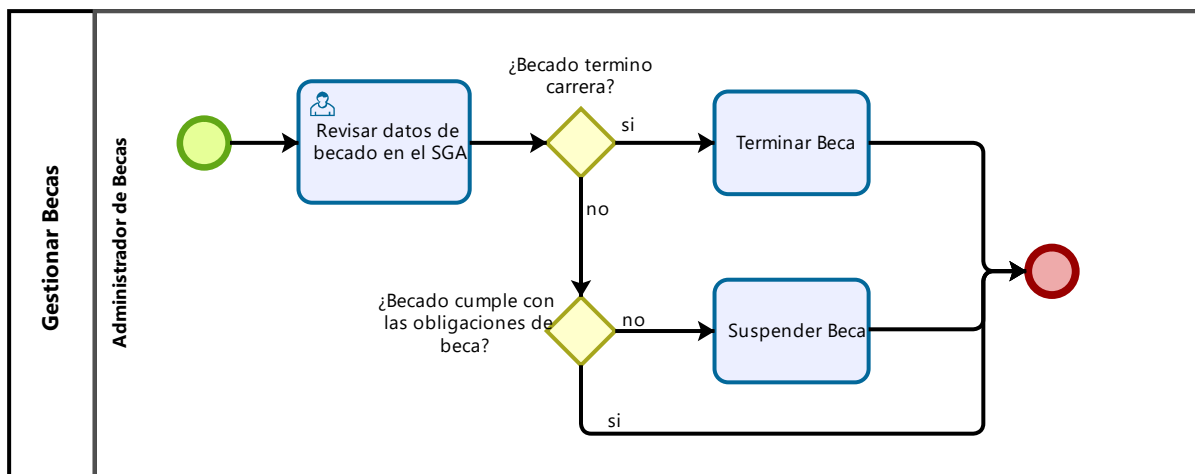


Figura 39. Diagrama BPMN - subprocesos para Gestionar Becas en la Etapa de Adjudicación.

### 1.2.7 Modelo BPMN de los subprocesos para la Etapa de Concesión

En esta etapa se realiza la gestión de pagos para las becas, el encargado de la sección becas provee el listado de becados activos en el tiempo acordado entre la Comisión de Becas y el Departamento Financiero. Por su parte, Contabilidad Financiera registra el devengado de los becados en el Sistema Financiero, obtiene un reporte que es enviado a Tesorería quien se encarga de realizar y registrar los pagos de las becas. [véase Figura 40].

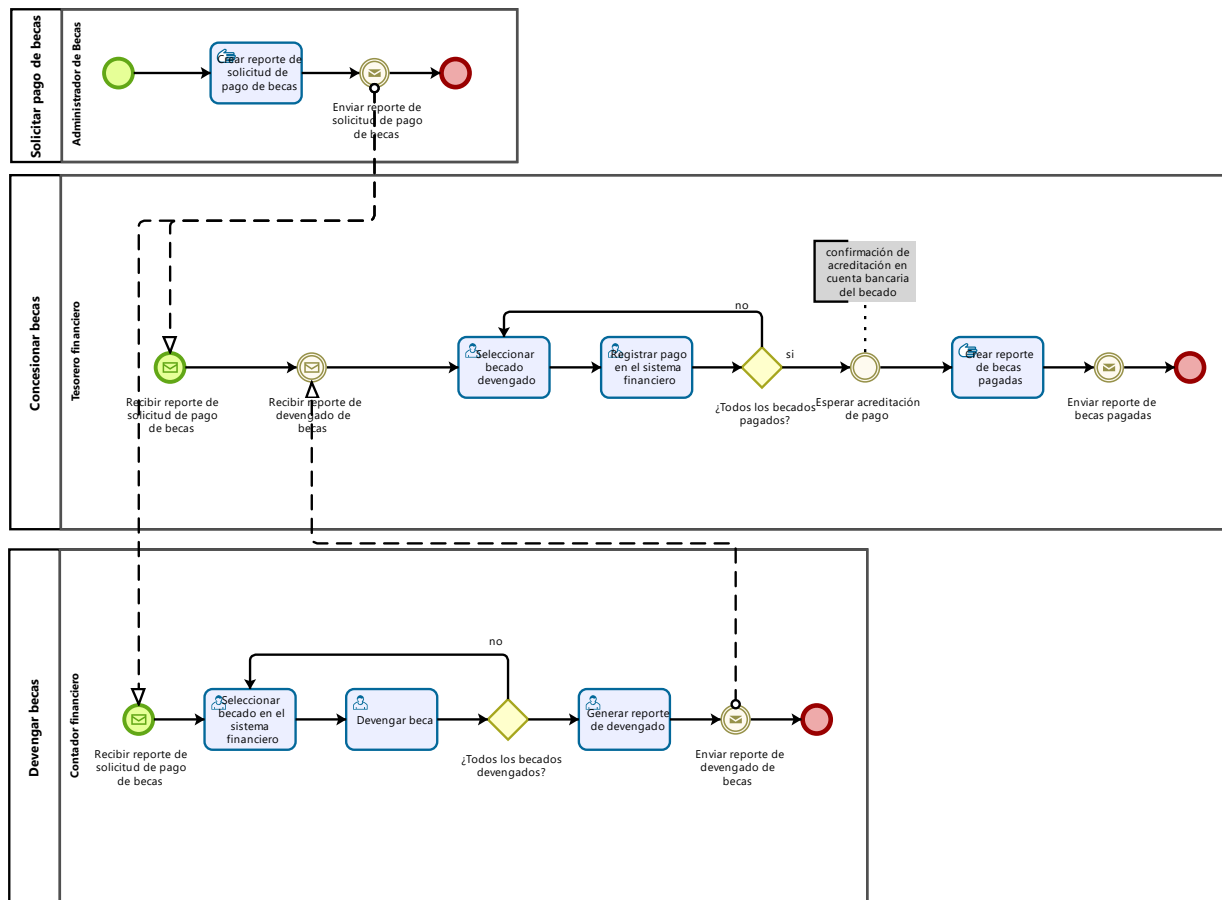


Figura 40. Diagrama BPMN - subprocesos para Concesionar Becas.

### 1.2.8 Especificación de los Requisitos de Software para el proceso de seguimiento y control de becas mediante el uso del estándar IEEE 830

Esta actividad inició luego del análisis de los procesos que intervienen en cada una de las etapas para la gestión de becas y tiene por objetivo definir y establecer los requisitos de software que están cubiertos en el desarrollo del módulo informático. La elaboración de este documento [véase Anexo 3] define el propósito, alcance, personal involucrado, perspectiva, restricciones suposiciones y dependencias, así como cada una de las funcionalidades que se requiere.

Los requisitos funcionales y no funcionales se proponen en base al Reglamento de Becas vigente [1], las consideraciones obtenidas a partir de las entrevistas generadas con los interesados en su desarrollo [véase Anexo 17] y los modelos BPMN diseñados en la actividad anterior. Los requisitos de software fueron validados por DTI y la UBU como clientes [véase Anexo 13 y Anexo 14].

(Véase Especificación de Requisitos de Software en base a la norma del IEEE 830 para el MGB)

## 2. Objetivo 2: Desarrollar el módulo para el proceso de seguimiento y control de becas, utilizando el framework Django y la metodología XP

### 2.1 Especificación de los requisitos de software para el proceso de seguimiento y control de becas mediante el uso del estándar IEEE 830

Durante esta actividad se diseñaron los diagramas de casos de uso para identificar los actores involucrados y las funcionalidades necesarias para lograr los objetivos del software. Para esto, se comenzó con la creación de casos de uso consolidados que agrupan tareas comunes y forman parte de una misma funcionalidad.

El caso de uso de revisión de becas implica varios casos de uso, como la sincronización de postulaciones desde el Módulo de Gestión de Trámites (MGT) para obtener información en el Módulo de Gestión de Becas (MGB). Además, el usuario debe poder ver los trámites asociados desde el MGB, actualizar el estado de postulación, asignar una calificación a la ficha socioeconómica y descargar la documentación individual del postulante. Todos estos casos de uso están subordinados al caso de uso de visualizar postulaciones, ya que el administrador necesita ver la información del postulante [véase Figura 41].

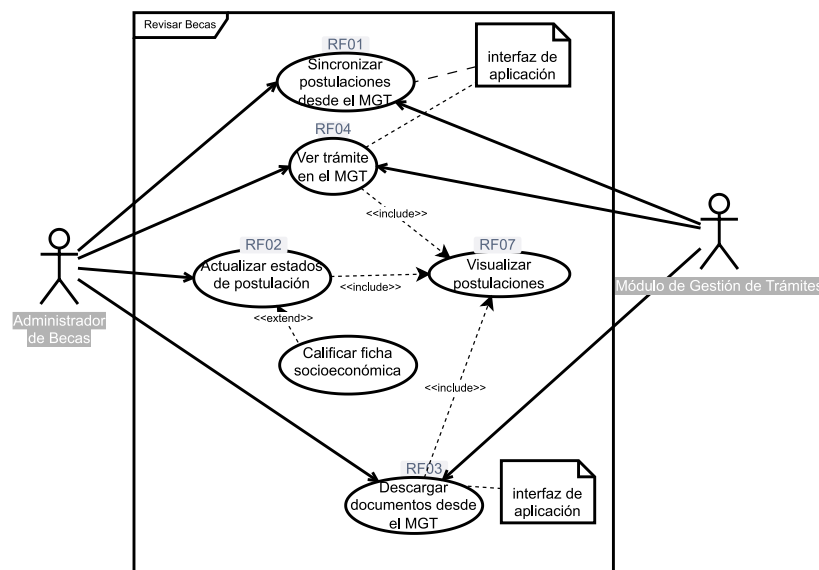


Figura 41. Caso de uso Revisar Becas.

El caso de uso de Seleccionar Becas incluye los mismos casos de uso que el caso de uso de Revisar Becas, ya que se requiere la misma funcionalidad cuando la comisión de becas se reúne para analizar cada caso y decidir si la beca es pertinente [véase Figura 42].

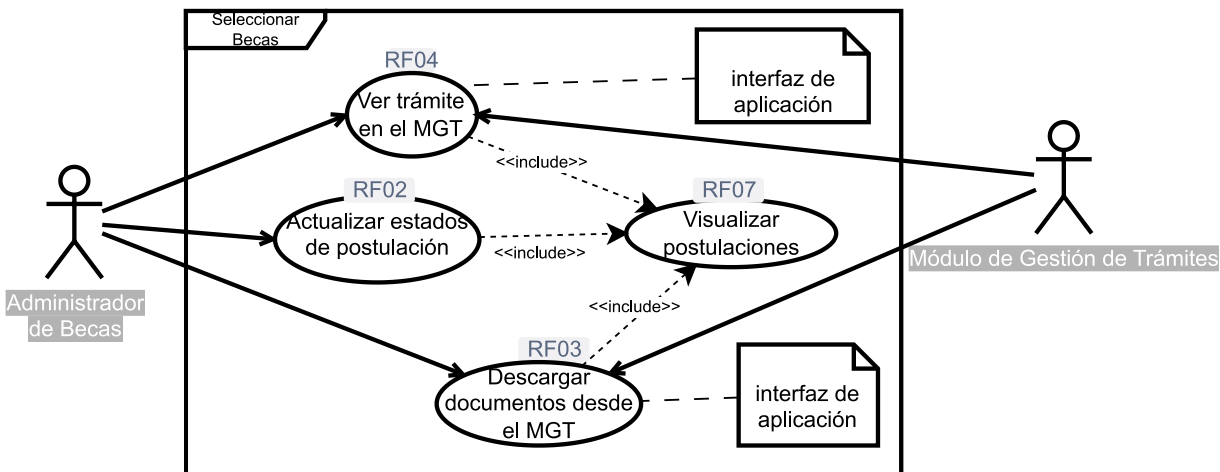


Figura 42. Caso de uso Seleccionar Becas.

En el caso de uso de Adjudicar Becas, se involucran los casos de uso de Confirmar Registro de Becado en el sistema financiero, ya que el actor de tesorería financiera necesita guardar la confirmación de registro para permitir que el becado acceda a los pagos de su beca. Además, el actor administrador de becas se encarga de gestionar las becas activas, actualizando sus estados a "terminado" o "suspendido" según sea necesario. Ambos casos de uso dependen del caso de uso Visualizar Becas para operar [véase Figura 43].

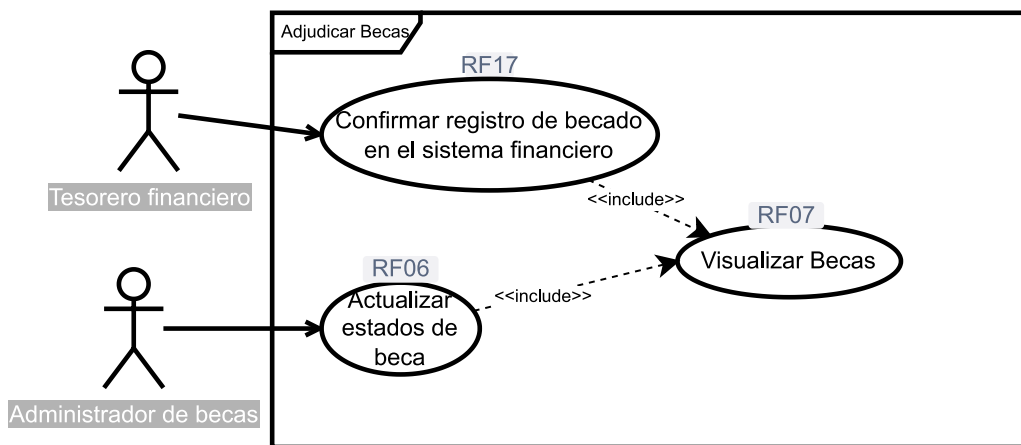


Figura 43. Caso de uso Adjudicar Becas

En el caso de uso de Concesionar Becas, se incluyen los casos de uso de Registrar Pago de Becas, que a su vez depende del caso de uso Visualizar Becas. Es necesario visualizar los registros de becas para gestionar los registros de pago correspondientes [véase Figura 44].

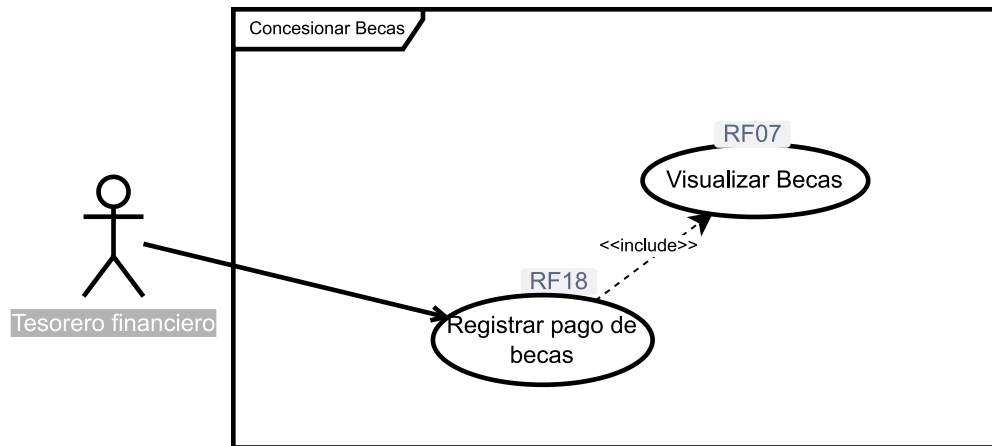


Figura 44. Caso de uso Concesionar Becas.

A partir de los diagramas de casos de uso específicos, se elaboró un diagrama de casos de uso general que agrupa las funcionalidades de cada uno y muestra cómo cada actor interactúa con estos casos de uso. Estos actores pertenecen a un actor usuario, que tiene como caso de uso principal el Iniciar Sesión en el SIAAF, sistema al cual pertenece el MGB.

En el caso de uso Revisar Becas, se incluyen varios casos de uso, como la sincronización de postulaciones y la visualización de la documentación del postulante. En el caso de uso Seleccionar Becas, se involucran los mismos casos de uso que el caso de uso de Revisar Becas. Por otro lado, el caso de uso Adjudicar Becas incluye el caso de uso Confirmar Registro de Becado y el caso de uso Administrar Becas. Finalmente, el caso de uso Concesionar Becas incluye el caso de uso Registrar Pago de Becas, que depende del caso de uso Visualizar Becas.

Cada uno de estos casos de uso involucra distintos actores, como el administrador de becas y la tesorería financiera, quienes requieren interactuar con el sistema para realizar sus tareas específicas. Todo esto se representa en el diagrama de casos de uso general, el cual permite tener una visión global del sistema y su funcionalidad [véase Figura 45].



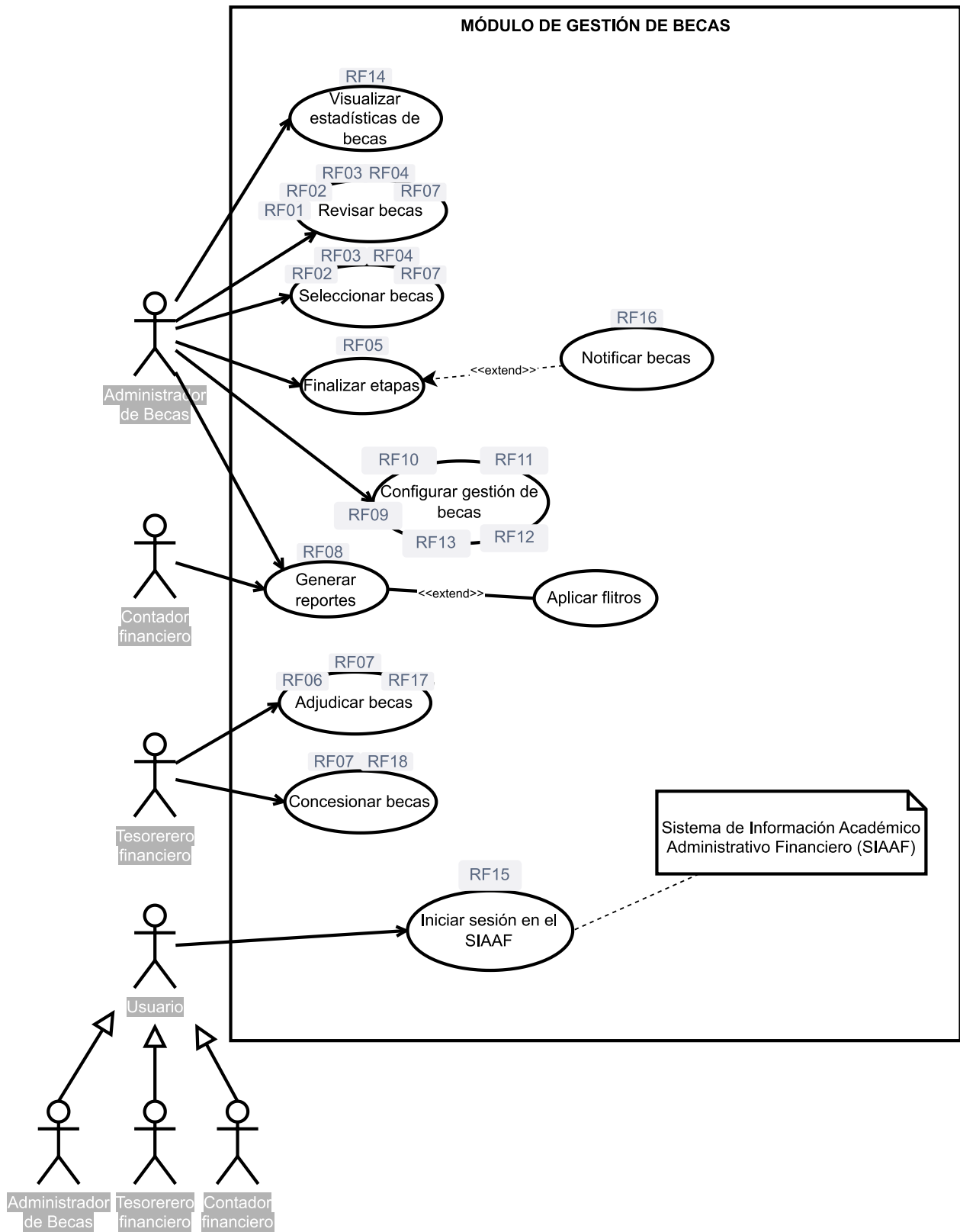


Figura 45. Casos de uso general para el Módulo de Gestión de Becas (MGB).

## **2.2 Diseño del Diagrama de Clases para el proceso de seguimiento y control de becas con el uso de UML**

Esta actividad consistió en el diseño del diagrama de clases para estructurar los datos del módulo informático y las relaciones necesarias entre estos. Además, se prestó especial cuidado para evitar la existencia de relaciones ambiguas, inconsistentes, redundantes o información duplicada en distintas clases. Primeramente, se diseñó una propuesta del diagrama para estructurar los datos, y las relaciones existentes de acuerdo al levantamiento de requisitos descritos en la actividad anterior. Esta propuesta fue presentada al equipo de desarrollo de DTI, quienes revisaron y aportaron con cambios significativos en el correcto uso de relaciones que permita soportar la infraestructura del módulo informático [véase Anexo 15].

Cabe indicar que el módulo informático forma parte del SIAAF, por lo que, algunos de los modelos que se diseñaron como propuesta, fueron sustituidos por otros preexistentes en distintos módulos del sistema; es por ello que el diagrama de clases tuvo que ser modificado, para incluir las relaciones con estos modelos.

## **2.3 Elaboración del Product Backlog mediante la técnica Users Story Mapping y validación con el Product Owner**

En la realización de esta actividad primeramente se elaboró un mapa de impacto que permitió visualizar el alcance del proyecto [véase Figura 46], representando las necesidades de los usuarios en el módulo informático durante la gestión de becas. Se ha ubicado a los tres actores principales (Postulante, Administrador de Becas y Tesorería) con sus correspondientes necesidades; se agrupó cada necesidad en su respectiva etapa y se agregaron notas a considerarse durante el desarrollo.

La actividad de priorizar y versionar fue desarrollada en conjunto con el Product Owner (ing. Máximo Álvarez) [véase reunión de trabajo grabada<sup>3</sup>].

**(véase Anexo 10)**

---

<sup>3</sup> [Priorización y versionado de historias de usuario.mp4 - Google Drive](#)

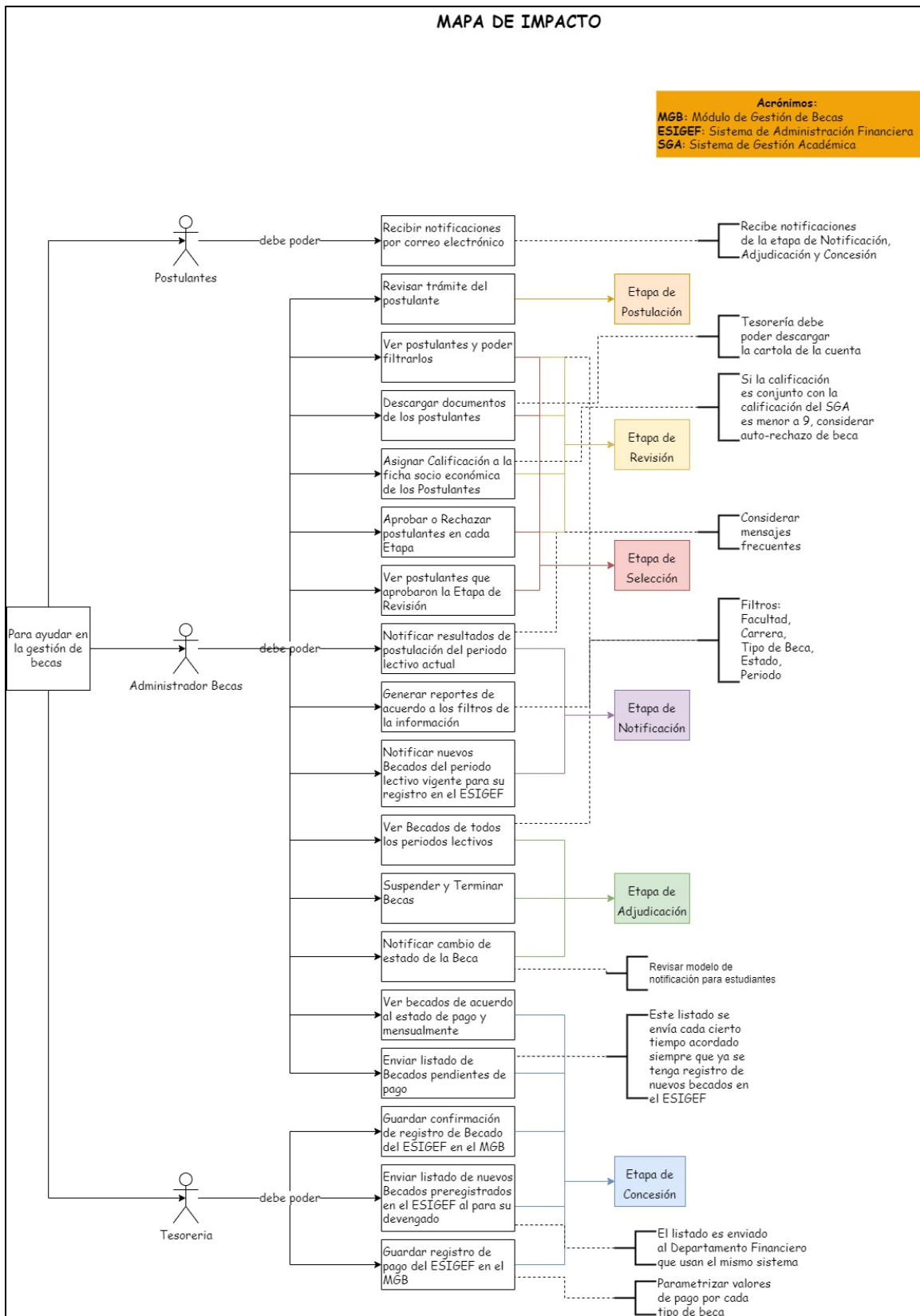


Figura 46. Mapa de Impacto que define el alcance del proyecto.

Con las necesidades claras de los actores definidos en el mapa de impacto, se hizo uso de la técnica del User Story Mapping que consiste en desglosar, priorizar y versionar Historias de Usuario. Empezando con el desglose, las necesidades de los actores se convierten en Historias de Usuario que se organizaron en las Etapas correspondientes para la Gestión de Becas. Luego se priorizaron las historias de acuerdo a la técnica moscow siglas del inglés para describir lo que tiene que estar, lo que debería estar, lo que podría estar y lo que por el momento está descartado. Una vez se priorizaron las Historias de Usuario lo último que se hizo es el versionado del software en base a la prioridad de las actividades tal como se muestra en la *Figura 47* [véase Anexo 10 para un detalle completo en la aplicación de esta técnica].

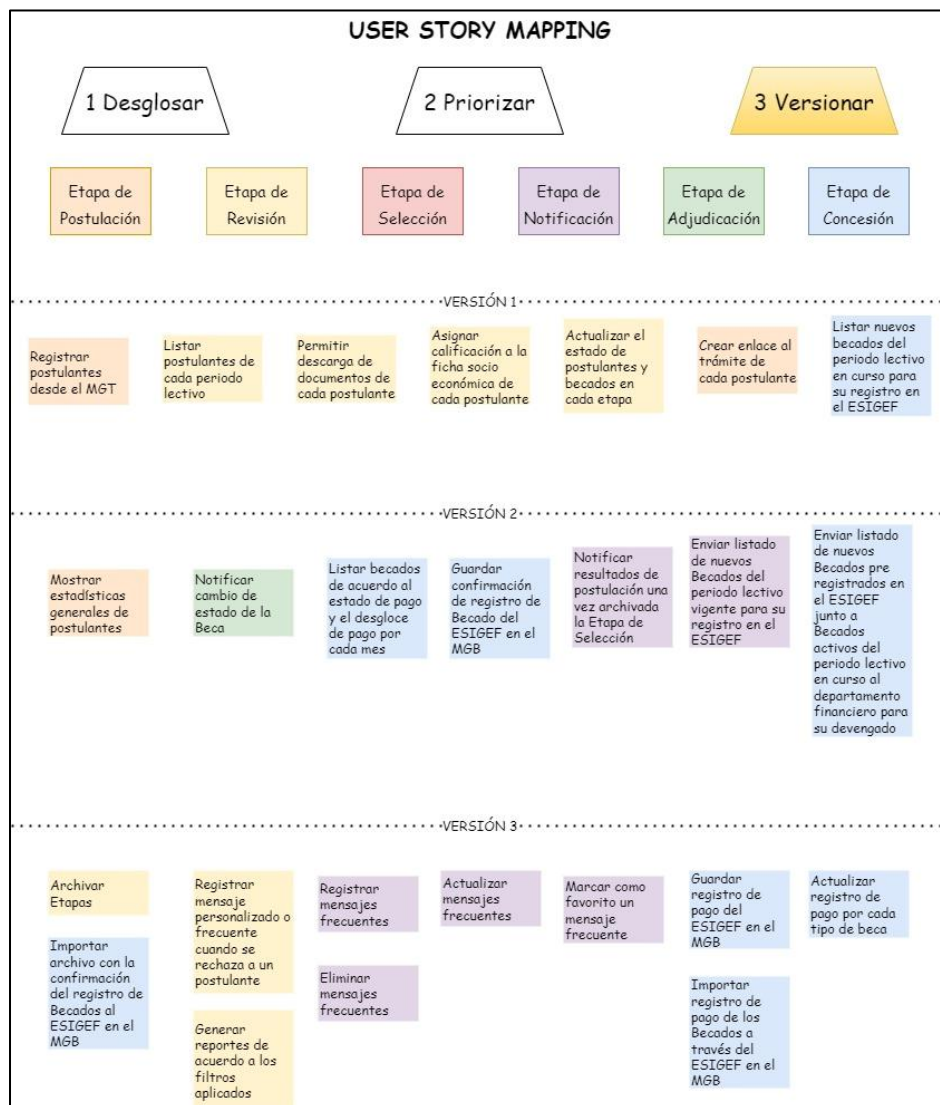


Figura 47. Versionado de las Historias de Usuario para el desarrollo del módulo informático.

Con las Historias de Usuario priorizadas y versionadas se hizo uso de la herramienta Jira para la administración de tareas del proyecto. En esta plataforma lo primero que se hizo fue definir las épicas (representadas por las Etapas de la gestión de Becas) en la hoja de ruta del proyecto. Luego se procedió a organizar cada una de las historias en el Backlog de acuerdo al versionado planificado para poder dividirlo en sprints entregables con fechas establecidas de 15 días cada versión.

## 2.4 Codificación de los Procesos para el Seguimiento y Control de Becas

En este apartado se describe el proceso de codificación iterativa con el fin de cumplir el objetivo 2 en base al ciclo de vida XP para el desarrollo de software.

### 2.4.1 Estructura del Módulo de Gestión de Becas

El MGB sigue el patrón de diseño MVT definido por Django que se compone de Modelos, Vistas y Plantillas [véase Figura 49, Figura 50 y Figura 51]. Además de acuerdo con la arquitectura definida por DTI, las vistas se encargan de gestionar las peticiones del servidor y responder con información que se orquesta a través de su correspondiente capa de aplicación para abstraer información de la base de datos y esta a su vez procesa la información en la capa de dominio. Las vistas gestionan los permisos del usuario a través de la capa de seguridad [véase Figura 48].

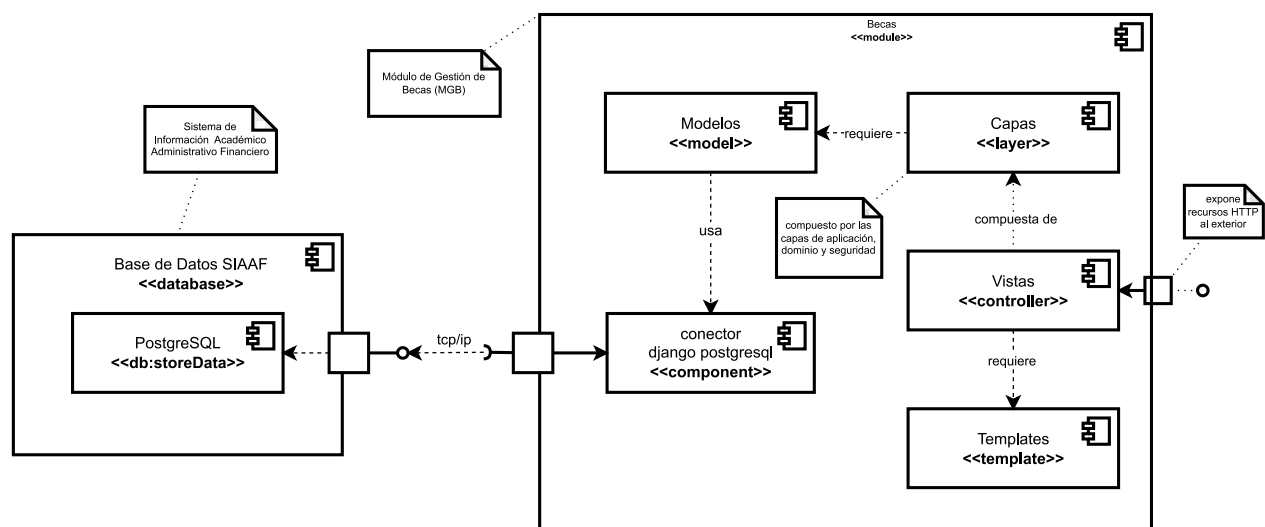


Figura 48. Diagrama de componentes del MGB

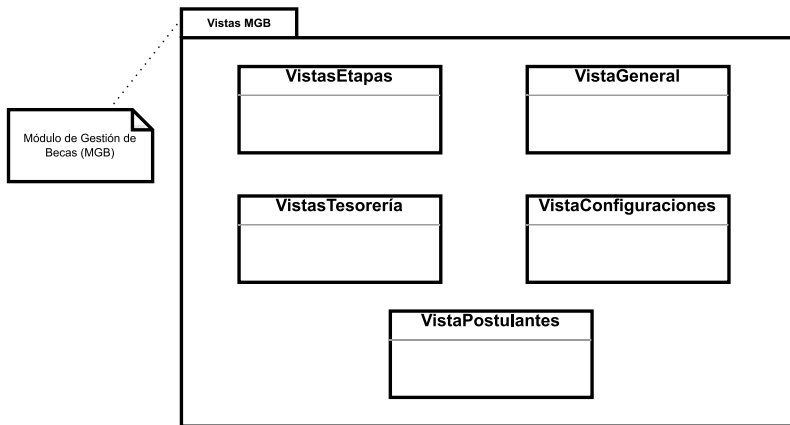


Figura 49. Vistas del MGB

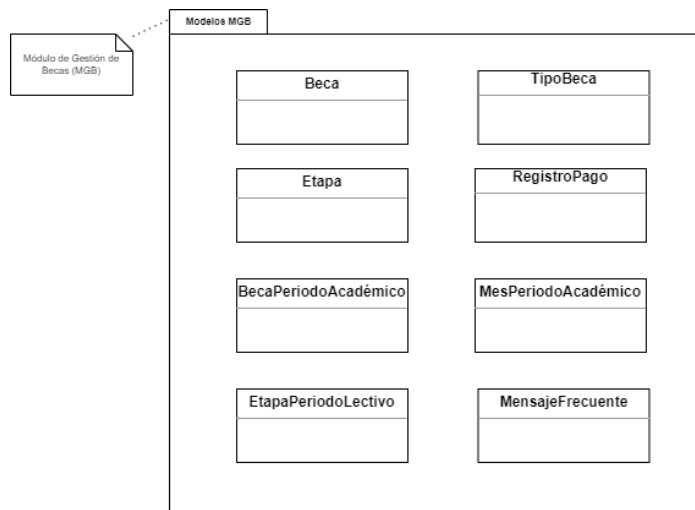


Figura 50. Modelos del MGB

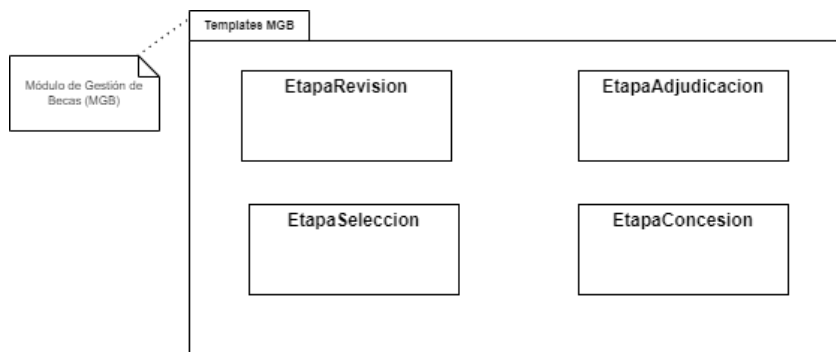


Figura 51. Plantillas del MGB

## 2.4.2 Patrones de codificación establecidos para el SIAAF

El equipo de desarrollo del SIAAF estableció unos patrones para la codificación de los módulos y el sistema en general.

- Las clases se escriben siempre con mayúsculas como lo recomienda Python.
- Las funciones y variables siguen el estilo de escritura "underscore case".
- Todo el código debe estar organizado en orden alfabético y escrito en español.
- Las variables o constantes y archivos JavaScript que se definen por cada archivo HTML llevan versionado en su nombre.
- Se reutilizan los fragmentos de código estructurales del sistema SIAAF para mantener la misma línea gráfica dentro del MGB [véase Figura 52].

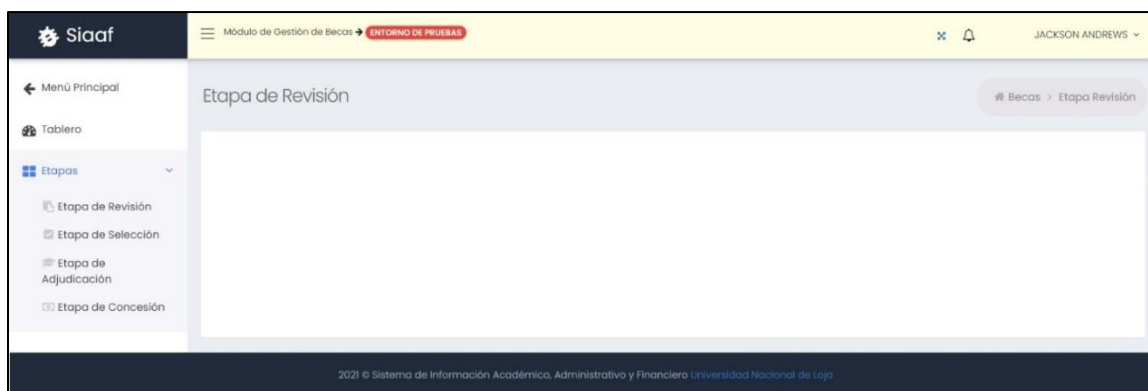


Figura 52. Bloques de código reutilizables del SIAAF.

## 2.4.3 Proceso de desarrollo del MGB

El proceso de desarrollo del módulo informático se llevó a cabo de manera iterativa con entregables periódicos cada 15 días con el fin de ejecutar el proyecto en un ambiente ágil y tomar las rectificaciones necesarias en cada entrega para su mejora.

### 2.4.3.1 Sprint 1

Las incidencias que se trabajaron para la primera versión entregable se detallan a continuación en la Figura 53.

▼ Sprint 1 - PMV 1 22 ene. – 5 feb. (8 incidencias)		
Permitir actualizar estados de las Etapas de Revisión, selección y Adjudicación de los becados		
MGBUNL-13	Actualizar estados del postulante/becado por etapas	ETAPA DE REVISIÓN
MGBUNL-10	Listar postulantes/becados de cada periodo lectivo por eta...	ETAPA DE REVISIÓN
MGBUNL-11	Descargar documentos de cada postulante	ETAPA DE REVISIÓN
MGBUNL-8	Enlace al trámite del postulante	ETAPA DE POSTULACIÓN
MGBUNL-22	Listar nuevos becados del periodo lectivo en curso para su regi...	ETAPA DE SELECCIÓN
MGBUNL-7	Registrar postulantes	ETAPA DE POSTULACIÓN
MGBUNL-12	Asignar calificación a la ficha socio económica del postulante	ETAPA DE REVISIÓN

Figura 53. Tareas planificadas para el Sprint 1 en Jira.

En el primer sprint se logró cumplir con los siguientes requisitos del software (RF01, RF02, RF03, RF04, RF06, RF07, RF15, RF17) que se describen a continuación:

#### 2.4.3.1.1 RF01. Sincronizar datos de postulación desde el MGT hacia el MGB

En esta tarea se creó un endpoint en el MGB que permite sincronizar la información desde el MGT [véase Código 5].

En la creación del endpoint se creó una vista Application Programming Interface (API) con clases en la capa dominio que se encargan de validar los datos recibidos y comprobar si el postulante fue previamente registrado o se trata de uno nuevo.

Código 5. Fragmento de Código del endpoint para registrar postulantes

```

1 def registrar_postulantes(request):
2     (postulantes, direcciones, becas, becas_periodo_academico_Etapa) =
3     RegistrarPostulantesDomService.crear_instancias_postulantes(data)
4     nuevos_postulantes = RegistrarPostulantesDomService.registrar_instancias_postulantes(
5     postulantes, direcciones, becas, becas_periodo_academico_Etapa)
6
7     return Response({'status': status.HTTP_200_OK,
8     'detail': 'Registro exitoso'},
9     status=status.HTTP_200_OK)
10 ...

```



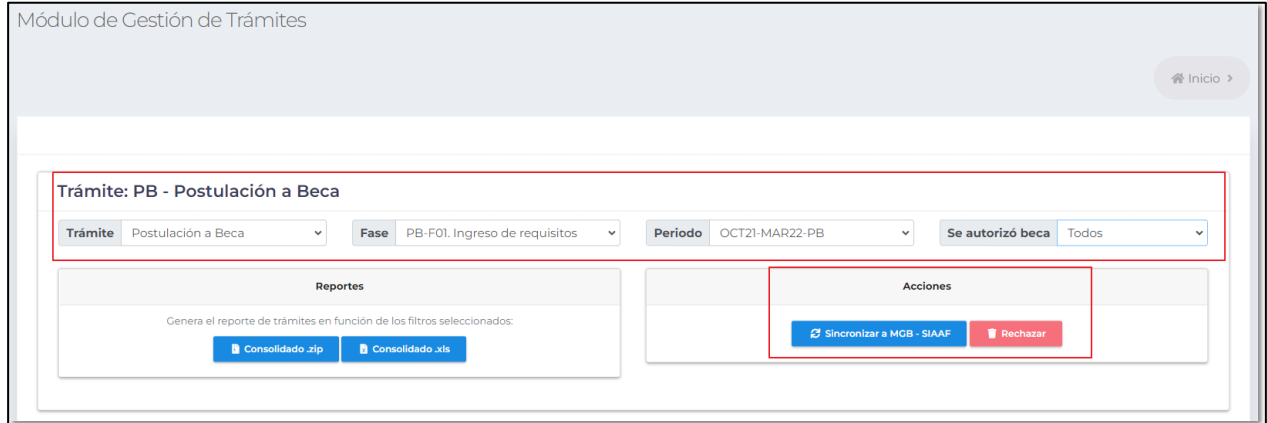


Figura 54. Vista en el MGT que permite sincronizar las postulaciones de cada Periodo Académico hacia el MGB.

#### 2.4.3.1.2 RF02 Actualizar estados de Postulantes

Para el cumplimiento de esta tarea se creó una función API [véase Código 6] que se encarga de actualizar el estado del postulante seleccionado. Desde el lado del cliente se creó un script [véase Código 7] con el uso de Ajax para evitar recargar la página y poder llamar a la función API mencionada. Se estableció una precondición para validar que la ficha socio-económica se encuentre calificada antes de actualizar el estado tanto del lado del cliente como del lado del servidor.

Código 6. Fragmento de Código del endpoint para actualizar estado de postulantes y becados

```

1 def Etapas_actualizar_estado(request):
2 try:
3 data = json.load(request)['json_data']
4 ...
5 for item in data:
6 beca_periodo_academico_Etapa = BecaPeriodoAcademicoEtapa.objects.select_for_update().get(pk= item['id'])
7 # validar que la Etapa no haya sido finalizada
8 ...
9 finalizar_Etapa_app_service.comprobar_estado_Etapa()
10
11 if (item['estado'] == BecaPeriodoAcademicoEtapaEstados.APROBADO or item['estado'] ==
BecaPeriodoAcademicoEtapaEstados.RECHAZADO) and beca_periodo_academico_Etapa.nota_ficha_socioeconomica ==
0.0:
12 return get_response(mensaje='Primero debe asignar una calificación a la ficha socioeconómica', status
=HTTP_400_BAD_REQUEST, data=response_data)
13

```

```

14 if beca_periodo_academico_Etapa.estado == item['estado']:
15 return get_response(mensaje='El estado no puede ser igual al actual, Elija un estado distinto para actualizar',
status=HTTP_400_BAD_REQUEST, data=response_data)
16 beca_periodo_academico_Etapa.razon = item['razon']
17 beca_periodo_academico_Etapa.estado = item['estado']
18
19 beca_periodo_academico_Etapa.save()
20
21 response_data.append({...})
22 return get_response(response_data, mensaje="Actualización exitosa")
23 ...

```

*Código 7. Fragmento de Código de la función Ajax para actualizar estado de postulantes y becados asincrónicamente*

```

1 async function actualizar_estado(indice_estado) {
2 try {
3 let listado = formatear_datos();
4 ...
5 var data = listado.map(item => { return { id: item.id, razon: item.razon, estado: parseInt(indice_estado) } });
6 ...
7 let respuesta = await peticionFetch(url_actualizar_estado, data);
8 respuesta.forEach(item => {
9 ...
10 });
11 $('#form-modal-editar').modal('hide');
12 } catch (error) {
13 ...
14 }
15 };

```

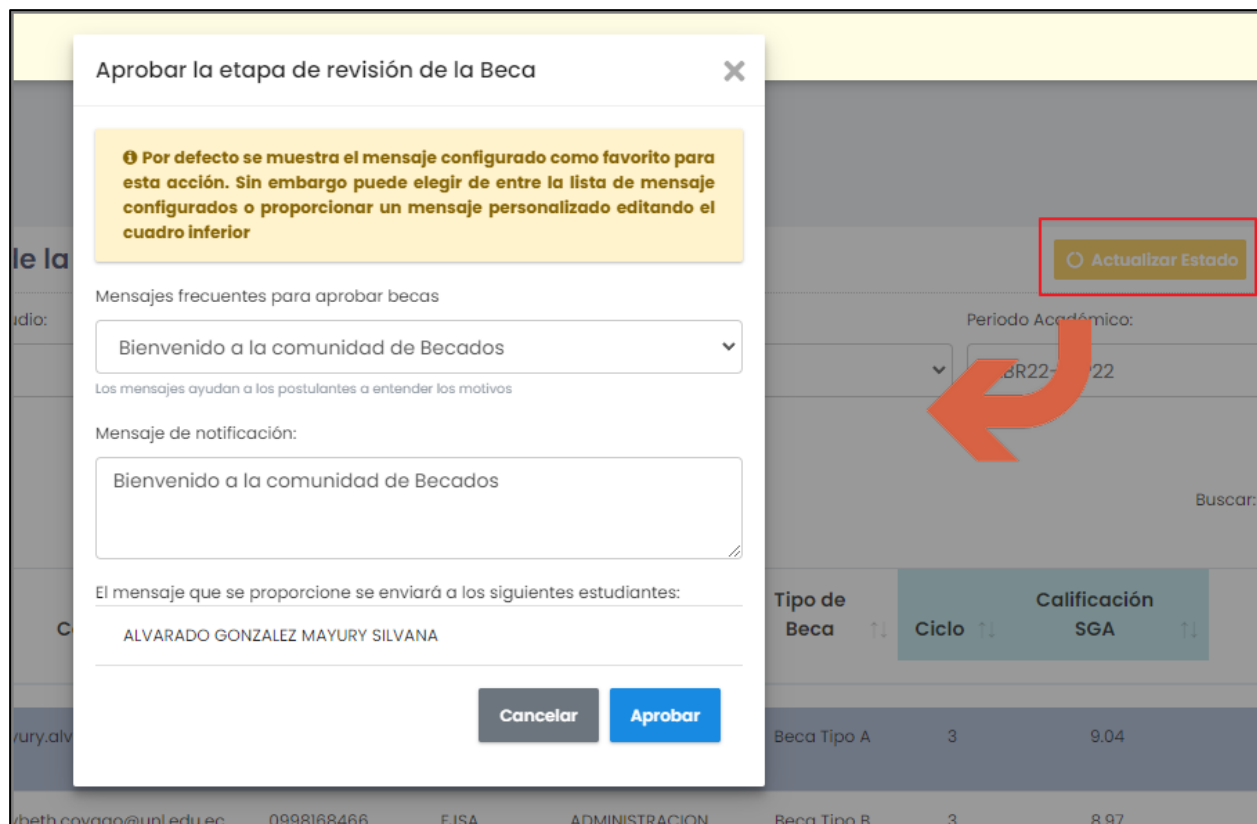


Figura 55. Ventana modal en la Etapa de Revisión que permite actualizar el estado de los Postulantes seleccionados.

#### 2.4.3.1.3 RF02. Asignar calificación a la ficha socio-económica del postulante

Esta tarea forma parte del Requisito Funcional 2, ya que es necesario asignar una calificación a la ficha socio-económica previo a actualizar el estado de un Postulante. Para cumplir con esta tarea se creó un endpoint [véase Código 8] que permita registrar la calificación asignada a los postulantes para la ficha socio-económica en la Etapa de Revisión. A través de un script asíncrono se llama al endpoint [véase Código 9] para no tener que recargar la página.

*Código 8. Fragmento de Código del endpoint que permite actualizar la ficha socio-económica de los Postulantes en la Etapa de Revisión*

```

1 def calificar_ficha_socioeconomica(request):
2     data = request.data
3     beca_periodo_academico_Etapa = BecaPeriodoAcademicoEtapa.objects.get(
4     pk=data['id'])
5     ...
6     finalizar_Etapa_app_service.comprobar_estado_Etapa()
7     beca_periodo_academico_Etapa.nota_ficha_socioeconomica = data['nota_ficha_socioeconomica']

```

```

8 beca_periodo_academico_Etapa.nota_promedio = mean([
9 beca_periodo_academico_Etapa.nota_sga,
10 float(beca_periodo_academico_Etapa.nota_ficha_socioeconomica)
11 ])
12 ...
13 return get_response(response_data, mensaje="Ficha Socioeconómica calificada correctamente")
14 ...

```

*Código 9. Fragmento de Código de la función Ajax que permite actualizar la ficha socio-económica de los postulantes en la Etapa de Revisión*

```

1 async function ficha_actualizar(id) {
2 var nota = $('#ficha-edit-#{id} #ficha-input').val();
3 const data = {...}
4 try {
5 const { id, nota_ficha_socioeconomica } = await ajax_generico(url_calificar_ficha_socioeconomica, data);
6 ficha_cancelar(id, nota_ficha_socioeconomica);
7 } catch (error) {
8 ...
9 }
10 }

```

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
ALVARADO GONZALEZ MAYURY SILVANA	1150047817	mayury.alvarado@unt.edu.ec	0999268475	FJSA	ADMINISTRACION PUBLICA	Beca Tipo A	3	9.04	<input type="text" value="9.04"/>	EN REVISIÓN

*Figura 56. Input que permite asignar una calificación a la ficha socio-económica del Postulante seleccionado.*

#### 2.4.3.1.4 RF03 Descargar documentos de postulación

Para completar esta actividad se habilitó un endpoint en el MGT que permita descargar todos los documentos del postulante. En el MGB se hace un llamado de éste a través de una función que recibe el archivo y lo procesa [véase Código 10] para devolverlo a la vista en donde a través de una función se llama asíncronamente [véase Código 11].

Código 10. Fragmento de Código de la función backend que permite descargar los documentos del postulante seleccionado

```

1 def descargar_documentos_postulantes(request: HttpRequest):
2     codigo_proceso = json.load(request)['json_data']['codigo_proceso']
3     return TramitesAppService.descargar_zip_postulantes(codigo_proceso)
4 ...

```

Código 11. Fragmento de Código de la función que permite descargar los documentos del postulante seleccionado

```

1 descargar_documentos = async function (url, codigo_proceso) {
2     try {
3         const data = {...}
4         const respuesta = await peticionFetchForFiles(url, data);
5         ...
6         link.download = file_name;
7         ...
8     } catch (error) {
9         ...
10    }
11 };

```

Etapa de Revisión

# Becas > Etapa Revisión

Listado de Postulaciones - Revisión de la Documentación Actualizar Estado Descargar documentos Ver Trámite

Facultad: --Todas-- Programa de estudio: --Todas-- Tipo de Beca: --Todas-- Estado: --Todos-- Período Académico: ABR22-SEP22

Mostrar 10 registros Ayuda Deseleccionar todo

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
ALVARADO GONZALEZ MAYURY SILVANA	1150047817	mayury.alvarado@uni.edu.ec	0989268475	FJSA	ADMINISTRACION PUBLICA	Beca Tipo A	3	9.04	0	EN REVISIÓN
COYAGO TUQUERES MAYBETH ARACELY	1751979657	maybeth.coyago@uni.edu.ec	0998168466	FJSA	ADMINISTRACION PUBLICA	Beca Tipo B	3	8.97	0	PENDIENTE
CUEVA YOMAIRA MERCEDES	1900779156	yomaira.cueva@uni.edu.ec	0988260036	FJSA	ADMINISTRACION PUBLICA	Beca Tipo A	3	8.59	0	PENDIENTE
GIRON TORRES GIAMPIER	1105212029	giampier.giron@uni.edu.ec	0961053467	FJSA	ADMINISTRACION PUBLICA	Beca Tipo B	3	8.75	0	PENDIENTE
GUAMAN DIAZ GABRIELA ANABELL	1150332482	gabriela.a.guaman@uni.edu.ec	0989276470	FJSA	ADMINISTRACION PUBLICA	Beca Tipo C	3	9.39	0	¡Archivo generado correctamente!
HERRERA JIMENEZ MARIA VANESSA	1150958724	maria.v.herrera@uni.edu.ec	0988683432	FJSA	ADMINISTRACION PUBLICA	Beca Tipo A	3	9.22	0	¡Archivo generado correctamente!
HURTADO ALVERCA DIANA ALEJANDRA	1150721502	diana.hurtado@uni.edu.ec	0968362387	FJSA	ADMINISTRACION PUBLICA	Beca Tipo C	3	9.26	0	¡Archivo generado correctamente!

Figura 57. Funcionalidad en la Etapa de Revisión que permite descargar la documentación de los Postulantes seleccionados.

### 2.4.3.1.5 RF04. Ver trámite en el MGT

Al igual que en la tarea anterior se requirió de un endpoint de redirección en el MGT que se encuentra registrado en el MGB [véase Código 12].

*Código 12. Fragmento de Código de la función ajax que permite abrir el trámite correspondiente al postulante seleccionado*

```
1 $('tools').on('click', '#btn-descargar-documentos, #btn-ver-tramite', function (currentTarget) {
2 let listado = formatear_datos();
3 let url = (path) => `${url_MGT}${path}/`;
4 ...
5 if (currentTarget.id == 'btn-descargar-documentos') {
6 ...
7 } else {
8 listado.forEach(element => {
9 setTimeout(function () {
10 window.open(url(url_tramite_postulante) + element.beca_codigo_proceso, '_blank');
11 }, tiempo_espera);
12 });
13 }
14 });
```

The screenshot shows the 'Etapa de Revisión' (Review Stage) interface. At the top, there are navigation links for 'Becas' and 'Etapa Revisión'. The main heading is 'Listado de Postulaciones - Revisión de la Documentación'. Below this, there are several filters: 'Facultad' (dropdown), 'Programa de estudio' (dropdown), 'Tipo de Beca' (dropdown), 'Estado' (dropdown), and 'Período Académico' (dropdown). There are also buttons for 'Actualizar Estado', 'Descargar documentos', and 'Ver Trámite' (highlighted with a red box). A search bar is present with the text 'DNI / apellidos / nombres'. Below the filters, there is a table with columns: Postulante, DNI, Correo, Celular, Facultad, Carrera, Tipo de Beca, Ciclo, Calificación SGA, Calificación Ficha Socioeconómica, and Estado. The table contains three rows of applicant data.

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
ALVARADO GONZALEZ MAYURY SILVANA	1150047817	mayury.alvarado@uni.edu.ec	0969268475	FJSA	ADMINISTRACION PUBLICA	Beca Tipo A	3	9.04	0	EN REVISIÓN
COYAGO TUQUERES MAYBETH ARACLIV	1751979657	maybeth.coyago@uni.edu.ec	0998168466	FJSA	ADMINISTRACION PUBLICA	Beca Tipo B	3	8.97	0	PENDIENTE
CUEVA YOMAIRA MERCEDES	1800779156	yomaira.cueva@uni.edu.ec	0988260036	FJSA	ADMINISTRACION PUBLICA	Beca Tipo A	3	8.59	0	PENDIENTE

*Figura 58. Funcionalidad en la Etapa de Revisión que permite ver el trámite de los Postulantes seleccionados en el MGT.*

### 2.4.3.1.6 RF06. Actualizar estados de Becas

Para el cumplimiento de esta tarea se creó una función API [véase Código 13] que se encarga de actualizar el estado del becado seleccionado. Desde el lado del cliente se creó un script con el uso de Ajax para evitar recargar la página y poder llamar a la función API mencionada.

*Código 13. Fragmento de Código del endpoint para actualizar estado de becados*

```
1 def Etapas_actualizar_estado(request):
2 try:
3 data = json.load(request)['json_data']
4 ...
5 for item in data:
6 beca_periodo_academico_Etapa = BecaPeriodoAcademicoEtapa.objects.select_for_update(
7 ).get(pk=item['id'])
8 # validar que la Etapa no haya sido finalizada
9 ...
10 finalizar_Etapa_app_service.comprobar_estado_Etapa()
11 ...
12 # cuando se actualice al estado a SUSPENDIDO se actualiza los registros de pago de pendiente a -
, ! suspendido
13 ...
14 if beca_periodo_academico_Etapa.estado == BecaPeriodoAcademicoEtapaEstados.SUSPENDIDO:
15 RegistroPago.objects.filter(...).update(...)
16 # cuando se actualice al estado a ACTIVO (reactivar) se actualiza los registros de pago de suspendido -
, ! a pendiente
17 if beca_periodo_academico_Etapa.estado == BecaPeriodoAcademicoEtapaEstados.ACTIVO
18 RegistroPago.objects.filter(...).update(...)
19 ...
20 beca_periodo_academico_Etapa.save()
21
22 if beca_periodo_academico_Etapa.estado == BecaPeriodoAcademicoEtapaEstados.SUSPENDIDO or -
, ! beca_periodo_academico_Etapa.estado == BecaPeriodoAcademicoEtapaEstados. -
, ! TERMINADO:
23 tipo_notificacion = BecaPeriodoAcademicoEtapaEstados.TERMINADO
24 ...
25
26 response_data.append({...})
27 if len(notificar_cambio_estado_becas) > 0:
28 # notificar a los postulantes que se suspendieron
29 EtapasAppService.notificar_beca(...)
30 return get_response(response_data, mensaje="Actualización exitosa")
31 ...
```

Etapa de Adjudicación # Becas > Etapa Adjudicación

**Listado de Becas** Confirmar Registro Actualizar Estado

Facultad: --Todas-- Programa de estudio: --Todas-- Tipo de Beca: --Todos-- Estado: --Todos-- Período Académico: --Todos--

Mostrar 10 registros Ayuda Deseleccionar todo

Buscar: DNI / apellidos / nombres

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Calificación Promedio	Estado	Registro en el ESIGEF
<input type="checkbox"/> Carrero Carrero Guillermo Villareal	1157961856	jan.luque@uni.edu.ec	0994277946	FSH	ODONTOLOGIA	Beca Tipo B	6	8.42	9	9.71	SUSPENDIDO	REGISTRADO
<input type="checkbox"/> Cortés Cortés Mirela Casas	1138581571	ualvarez@uni.edu.ec	0924884999	FJSA	ECONOMIA	Beca Tipo B	5	9.01	9	9	SUSPENDIDO	EN REGISTRAR
<input type="checkbox"/> Flores Flores Gabriela Farias	1140648803	joseantonio37@uni.edu.ec	0941431498	FERNNR	Ingeniería en Electrónica y Telecomunicaciones	Beca Tipo A	5	9.72	9	9.36	SUSPENDIDO	EN REGISTRAR

Figura 59. Funcionalidad que permite actualizar el estado de los Becados seleccionados en la Etapa de Adjudicación.

### 2.4.3.1.7 RF07 Visualizar Postulaciones y Becas

Para el listado de los postulantes y becados se hizo uso de datatable [véase Código 14] que permite optimizar el consumo de datos, consultando de manera eficiente bajo demanda y paginando los resultados. Se creó un script para cada Etapa que permiten obtener la información mediante peticiones Ajax al servidor [véase Código 15].

Código 14. Fragmento de Código de la función ajax que permite cargar datos de Postulantes en la Etapa de Revisión

```

1 function initDataTable() {
2   table = $('#tabla').DataTable({
3     ...
4     "ajax": {
5       "url": url_Etapa_revision_paginador,
6       "type": "POST",
7       "data": function (d) {
8         d.facultad = $('#facultad').val();
9         d.carrera = $('#carrera').val();
10      ...
11    },
12    ...
13  },
14  columnDefs: [...],

```



```

15 "columns": [
16 { "data": null, "defaultContent": "" },
17 { "data": "beca__persona" },
18 ...
19 ],
20 ...
21 });
22 }

```

*Código 15. Fragmento de Código de la vista para cargar información de Postulantes en la Etapa de Revisión*

```

1 def Etapa_revisión_paginador(request):
2 try:
3 Etapa_params = DataTableParams(request, **request.POST)
4 Etapa = Etapa.objects.get(
5 código=Constantes.ETAPA_REVISIÓN.value)
6 EtapasAppService.get_datatable(
7 Etapa_params=Etapa_params, Etapa_código=Etapa.código)
8 data = [{
9 'id': item.id,
10 'beca_persona': item.beca.persona.get_nombres_completos_inverso(),
11 ...
12 } for item in Etapa_params.items]
13 result = Etapa_params.result(data)
14 return JsonResponse(result)
15 ...

```

Para las Etapas de Revisión, Selección y Adjudicación era necesario filtrar la información de los postulantes o becados según sea el caso, acuerdo a los siguientes parámetros:

- Facultad
- Carrera
- Tipo de Beca
- Estado
- Periodo Académico

Para ello se creó un fragmento de vista que permita reutilizar la funcionalidad en cada Etapa [véase Código 16].

Código 16. Fragmento de Código HTML con los filtros de las Etapas de Revisión, Selección y Adjudicación

```
1 {% block filtrosEtapas %}
2 <div class="form-row">
3 <div class="form-group col-md-2">
4 <label for="facultad">Facultad: </label>
5 <select id="facultad" class="form-control">
6 ...
7 </select>
8 </div>
9 <div class="form-group col-md-2">
10 <label for="carrera">Programa de estudio: </label>
11 <select id="carrera" class="form-control">
12 ...
13 </select>
14 </div>
15 <div class="form-group col-md-2">
16 <label for="tipo_beca">Tipo de Beca: </label>
17 <select id="tipo_beca" class="form-control">
18 ...
19 </select>
20 </div>
21 <div class="form-group col-md-2">
22 <label for="estado">Estado: </label>
23 <select id="estado" class="form-control">
24 ...
25 </select>
26 </div>
27 <div class="form-group col-md-2">
28 <label class="text-nowrap" for="periodo_academico">Periodo Académico: </label>
29 <select id="periodo_academico" class="form-control">
30 ...
31 </select>
32 </div>
33 </div>
34 {% endblock filtrosEtapas %}
```

Etapa de Revisión

# Beca > Etapa Revisión

Listado de Postulaciones - Revisión de la Documentación

Facultad: --Todas-- Programa de estudio: --Todas-- Tipo de Beca: --Todas-- Estado: Revisar Período Académico: A0122-S1P22

Mostrar: 10 registros

Buscar: DNI / apellidos / nombres

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
ALVARADO NICOLA DZ MAYURY ZIVADA	10004387	mayuryalvaradomayury@uclac	0962268475	ESIA	ADMINISTRACION PUBLICA	Beca Tipo A	3	804	0	Revisar
PIÑA ALFARZO JUAN ANDRES	10050493	juan.alfarzo@uclac	0982259827	ESIA	ADMINISTRACION PUBLICA	Beca Tipo A	3	808	0	Revisar

Mostrando registros del 1 al 2 de un total de 2 registros (Filtrado de un total de 87 registros) Clic en una fila para seleccionar

Estados disponibles: [cancelado](#) [rechazado](#) [revisado](#)

[Generar Reporte](#) [Finalizar Etapa](#)

Figura 60. Lista de Postulantes en la Etapa de Revisión.

Etapa de Selección

# Beca > Etapa Selección

Listado de Postulaciones - Evaluación interna de la Comisión de Becas

Facultad: --Todas-- Programa de estudio: --Todas-- Tipo de Beca: Beca Tipo B Estado: Aprobado Período Académico: A0122-S1P22

Mostrar: 10 registros

Buscar: DNI / apellidos / nombres

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Calificación Promedio	Estado
Correa Correa Guillermo Wilson	E2760335	wilsoncorrea@uclac	066-279145	ESIA	ADMINISTRACION PUBLICA	Beca Tipo B	3	842	9	87	Seleccionado
Correa Correa Wilson Wilson	E3088937	wilsoncorrea@uclac	0124604555	ESIA	ECONOMIA	Beca Tipo B	3	307	9	8	Seleccionado

Mostrando registros del 1 al 2 de un total de 2 registros (Filtrado de un total de 83 registros) Clic en una fila para seleccionar

Estados disponibles: [cancelado](#) [rechazado](#) [revisado](#)

[Generar Reporte](#) [Finalizar Etapa](#)

Figura 61. Lista de Postulantes en la Etapa de Selección.

Etapa de Adjudicación

# Beca > Etapa Adjudicación

Listado de Becas

Facultad: --Todas-- Programa de estudio: --Todas-- Tipo de Beca: Beca Tipo B Estado: Suspendido Período Académico: --Todas--

Mostrar: 10 registros

Buscar: DNI / apellidos / nombres

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Calificación Promedio	Estado	Registro en el ESIGAF
Correa Correa Wilson Wilson	E2760335	wilsoncorrea@uclac	066-279145	ESIA	ADMINISTRACION PUBLICA	Beca Tipo B	3	842	9	87	Suspendido	Seleccionado
Correa Correa Wilson Wilson	E3088937	wilsoncorrea@uclac	066-279145	ESIA	ECONOMIA	Beca Tipo B	3	307	9	8	Suspendido	Seleccionado

Mostrando registros del 1 al 2 de un total de 2 registros (Filtrado de un total de 83 registros) Clic en una fila para seleccionar

Estados disponibles: [cancelado](#) [rechazado](#) [revisado](#)

[Generar Reporte](#)

Figura 62. Lista de Becados en la Etapa de Adjudicación.



```
17 return get_response(response_data, mensaje="Registro exitoso")
```

Código 18. Fragmento de Código de la función ajax que permite guardar la confirmación de registro en el ESIGEF de los Becados

```
1 $(".tools").on("click", "#confirmar_registro_esigef", async function () {
2 try {
3 let listado = formatear_datos();
4 ...
5 var data = listado.map(item => { return { id: item.id, confirmacion_registro_esigef: true } });
6 let respuesta = await peticionFetch(url_confirmar_registro_esigef, data);
7 respuesta.forEach(item => {...});
8 } catch (error) {
9 ...
10 }
11 });
```

Etapa de Adjudicación

Becas > Etapa Adjudicación

### Listado de Becas

Facultad: --Todas-- Programa de estudio: --Todas-- Tipo de Beca: --Todas-- Estado: Reactivar Período Académico: --Todos--

Mostrar 10 registros

Buscar: DNI / apellidos / nombres

¡ Ayuda X Deseleccionar todo

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Calificación Promedio	Estado	Registro en el ESIGEF
<input type="checkbox"/> Meraz Meraz Malak Lara	1153320850	guillermo.barrientos@uni.edu.ec	0995501179	FSH	Laboratorio Clínico	Beca Tipo C	6	9.9	8.7	9.3	ACTIVO	REGISTRADO
<input checked="" type="checkbox"/> Rivera Rivera Rayan Zepeda	1133689093	melendezrodrigo@uni.edu.ec	0986466573	FSH	Medicina Humana	Beca Tipo C	6	7.53	9.8	8.87	ACTIVO	EN REGISTRO

Mostrando registros del 1 al 2 de un total de 2 registros (filtrado de un total de 9 registros) Fila seleccionada

Estados disponibles: resoluto inscripciones terminar

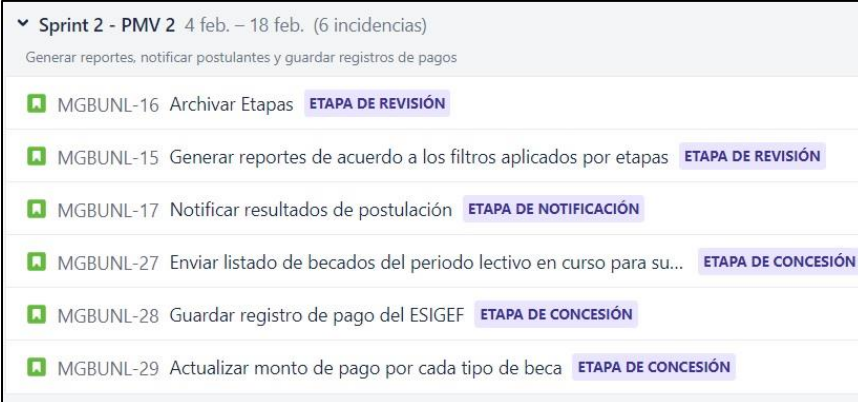
Anterior 1 Siguiente

Generar Reporte

Figura 64. Funcionalidad en la Etapa de Adjudicación que le permite al usuario de tesorería guardar la confirmación de registro en el ESIGEF.

### 2.4.3.2 Sprint 2

Las incidencias que se trabajaron en la segunda versión entregable se detallan a continuación en la Figura 65.



ID	Tarea	Estatus
MGBUNL-16	Archivar Etapas	ETAPA DE REVISIÓN
MGBUNL-15	Generar reportes de acuerdo a los filtros aplicados por etapas	ETAPA DE REVISIÓN
MGBUNL-17	Notificar resultados de postulación	ETAPA DE NOTIFICACIÓN
MGBUNL-27	Enviar listado de becados del periodo lectivo en curso para su...	ETAPA DE CONCESIÓN
MGBUNL-28	Guardar registro de pago del ESIGEF	ETAPA DE CONCESIÓN
MGBUNL-29	Actualizar monto de pago por cada tipo de beca	ETAPA DE CONCESIÓN

Figura 65. Tareas planificadas para el Sprint 2 en Jira.

Para el segundo Sprint se alcanzó la culminación de los siguientes requisitos funcionales (RF05, RF08, RF13, RF16 y RF18) que se detalla a continuación:

RF05 Finalizar Etapas

Para cumplir con esta tarea se creó un nuevo modelo denominado *EtapasFinalizadas* con el fin de guardar los registros de Etapas finalizadas. Se crearon clases en las capas de Aplicación y Dominio para poder hacer uso de sus métodos en la Vista que se encarga de finalizar Etapas [véase Código 19].

Con el cumplimiento de esta tarea el usuario Administrador de Becas es capaz de finalizar las Etapas de Revisión y Selección de acuerdo al periodo académico correspondiente. Esto con el propósito de que no se edite la información por error en un futuro y que exista un respaldo confiable [véase Figura 66].

*Código 19. Fragmento de Código de la Vista que permite finalizar las Etapas de Revisión y Selección*

```
1 def finalizar_Etapa(request: HttpRequest):
2     try:
3         data = json.load(request)['json_data']
4         ...
5         finalizar_Etapa_app_service = FinalizarEtapasAppService(
6             Etapa, periodo_academico_id)
```

```

7 finalizar_Etapa_app_service.set_siguiete_Etapa()
8 finalizar_Etapa_app_service.comprobar_estado_Etapa()
9 finalizar_Etapa_app_service.set_becas_periodo_academico_Etapa()
10 finalizar_Etapa_app_service.crear_becas_siguiete_Etapa()
11 with transaction.atomic():
12 finalizar_Etapa_app_service.guardar_becas_siguiete_Etapa()
13 Etapa_finalizada_ok = finalizar_Etapa_app_service.finalizar_Etapa()
14 if Etapa_finalizada_ok:
15 return get_response(...)
16 return JsonResponse(FinalizarEtapasAppService.mensaje_Etapa_no_finalizada)
17 except Exception as e:
18 ...

```

Etapa de Revisión Becas > Etapa Revisión

Listado de Postulaciones – Revisión de la Documentación Actualizar Estado Descargar documentos Ver Trámite

Facultad: --Todas-- Programa de estudio: --Todas-- Tipo de Beca: Beca Tipo C Estado: --Todos-- Período Académico: ABR22-SEP22

Mostrar 10 registros Ayuda Deseleccionar todo

Buscar: DNI / apellidos / nombres

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
<input type="checkbox"/> GUAMAN DIAZ GABRIELA ANABELL	1150332482	gabriela.guaman@uni.edu.ec	0989276470	FJSA	ADMINISTRACION PUBLICA	Beca Tipo C	3	9.39	0	PENDIENTE
<input type="checkbox"/> HURTADO ALVERCA DIANA ALEJANDRA	1150721502	diana.hurtado@unitedu.ec	0988362387	FJSA	ADMINISTRACION PUBLICA	Beca Tipo C	3	9.26	0	PENDIENTE

Mostrando registros del 1 al 2 de un total de 2 registros (filtrado de un total de 87 registros) Clic en una fila para seleccionarla

Estados disponibles: sin revisar revisar aprobar rechazar Generar Reporte Finalizar Etapa

Anterior 1 Siguiente

Figura 66. Funcionalidad en la Etapa de Revisión que permite finalizar la Etapa para el Período Académico elegido.

#### 2.4.3.2.1 RF08. Generar Reportes

Para cumplir con esta tarea se crearon clases en las capas de Aplicación y Dominio que obtiene la información de acuerdo a los filtros aplicados en cada una de las Etapas y devuelve una hoja de cálculo con los registros correspondientes.

Las clases de las capas se llaman en la Vista que gestiona el tipo de Reporte correspondiente [véase Código 20].

Con esta tarea el usuario Administrador de Becas puede descargar reportes en las Etapas de

Revisión, Selección, Adjudicación y Concesión filtrando los resultados por facultad, carrera, tipo de beca, estado y periodo académico. El usuario de Tesorería es capaz de descargar reportes generales en las Etapas de Adjudicación y Concesión, además del reporte para importar registros de pago. El usuario de Contabilidad Financiera es capaz de descargar reportes en la Etapa de Concesión [véase Figura 67].

*Código 20. Fragmento de Código de la Vista que permite generar Reportes para todas las Etapas*

```

1 def generar_reporte(request, *args, **kwargs):
2     params = json.load(request)['json_data']
3     if EtapasSecService.puede_descargar_reporte_importacion(request.user) and params.get('Etapa_codigo') == -
4     ,! Constantes.ETAPA_CONCESION.value and params.get('concesion'):
5     return ReportesDomService.get_reporte_para_concesionar(params)
6     elif EtapasSecService.puede_descargar_reporte_Etapas(request.user) and params.get('Etapa_codigo') == -
7     ,! Constantes.ETAPA_CONCESION.value:
8     return ReportesDomService.get_reporte_Etapa_concesion(params)
9     elif EtapasSecService.puede_descargar_reporte_Etapas(request.user):
10    return ReportesDomService.get_reporte_Etapas(params)
11 else:
12 return get_response(mensaje='No tiene los permisos suficientes, contactese con el administrador del sitio', -
13 ,! status=400)

```

The screenshot shows the 'Etapa de Concesión' interface. At the top, there are navigation links for 'Becas' and 'Etapa de Concesión'. Below this is the 'Registro de Pago de Becas' section, which includes a filter box with dropdowns for 'Tipo de Beca' (set to '--Todas--'), 'Periodo Académico' (set to 'ABR22-SEP22'), and 'Estado Registro' (set to '--Todas--'). There is also a 'Meses' field with a button 'Elija los meses a mostrar'. To the right of the filter box are buttons for 'Reporte para Concesión' (highlighted with a red box), 'Importar Registros', 'Actualizar Estado', and 'Sincronizar Becas Activas'. Below the filter box is a search bar with the text 'DNI / apellidos / nombres' and a 'Mostrar 10 registros' indicator. The main part of the interface is a table with columns: 'Becado', 'DNI', 'Tipo Beca', 'Mes 1', 'Mes 2', 'Mes 3', 'Mes 4', 'Mes 5', and 'Mes 6'. Two rows of data are visible, each with a checkbox and a 'PENDIENTE \$ 45' status for each month. At the bottom, there is a 'Mostrando registros del 1 al 2 de un total de 2 registros' message, a 'Generar Reporte' button (highlighted with a red box), and a 'Botón presenten en todas las Etapas' callout bubble.

*Figura 67. Funcionalidad que permite generar reportes de acuerdo a los filtros aplicados con la información de cada Etapa.*



### 2.4.3.2.2 RF13. Actualizar montos de pago

Para cumplir con esta tarea se creó una vista que se encargue de recibir y procesar la actualización del monto de pago correspondiente según el tipo de beca [véase Código 21].

*Código 21. Fragmento de Código de la Vista que permite actualizar los montos de pago de cada tipo de Beca*

```
1 def monto_beca(request):
2 try:
3 data = json.load(request)['json_data']
4 tipos_becas = TipoBeca.objects.all()
5 try:
6 monto_a = float(data['monto_a'])
7 ...
8 if monto_a < 0 or monto_b < 0 or monto_c < 0:
9 raise Exception("El monto de pago debe ser un valor positivo")
10 except Exception as e:
11 ...
12 return get_response(mensaje=mensaje, data={}, status=400)
13 for tipo_beca in tipos_becas:
14 if tipo_beca.codigo == TipoBecaCodigos.BECA_TIPO_A:
15 tipo_beca.monto_pago = monto_a
16 ...
17 tipos_becas.bulk_update(tipos_becas, ['monto_pago'])
18 return get_response(...)
19 except Exception as e:
20 ...
```

Con esta tarea el usuario Administrador de Becas es capaz de actualizar los montos de pago para las Becas Tipo A, B y C.

A continuación, se puede visualizar la funcionalidad implementada en la sección Configuraciones de Becas [véase Figura 68].

Figura 68. Sección de Configuraciones que le permite al Administrador de Becas actualizar los valores de pago para cada tipo de Beca.

### A.1.3.2. RF16 Notificar resultados de postulación

Para cumplir con esta tarea se creó una función que se encarga de notificar a los Postulantes y al usuario de Tesorería acerca de los resultados de Postulación [véase Código 22]. Además, se creó un modelo para registrar las notificaciones que se envían a los Postulantes con el mensaje frecuente de aprobación o rechazo según sea el caso.

Con esta tarea el usuario Administrador de Becas es capaz de notificar a los Estudiantes sus resultados de postulación [véase Figura 69], y al usuario de Tesorería para que pueda proceder con el registro de los nuevos Becados en el sistema financiero.

Código 22. Fragmento de Código de la función de la capa de aplicación que se encarga de guardar la notificación de los Estudiantes y Tesorería

```

1 def notificar_nuevos_becados(listado_estudiantes: List[BecaPeriodoAcademicoEtapa] = []):
2 ...
3 BecasAppService.notificar_estudiantes(asunto, listado_estudiantes)
4 asunto = 'Nuevos Becados para el presente periodo lectivo'
5 texto = render_to_string(
6 '././becas/templates/fragmentos/fragmento_notificacion_tesoreria.html',
7 {'numero_nuevos_becados': len(
8 listado_estudiantes), 'comentario': comentario}
9 )
10 BecasAppService.notificar_tesoreria(asunto, texto)

```



Figura 69. Notificación recibida por un estudiante con los resultados de su postulación.

#### 2.4.3.2.3 RF18 Registrar Pagos de Becas

El MGB permite importar los registros de devengado y pago desde una hoja de cálculo al usuario del Tesorería. Además, puede actualizar individualmente los registros de pago.

Esta funcionalidad les permite a los usuarios, depurar el archivo que se desea importar, considerando una serie de validaciones. Si se cumplen, la importación se realiza exitosamente, de lo contrario se crea un archivo con los registros que presentan inconsistencias disponibles para su descarga.

Para completar esta tarea se creó una vista que a través de las capas de Aplicación y Dominio se encarga de leer el archivo que se va a importar, tomando en cuenta ciertas validaciones que debe cumplir el archivo para que se pueda procesar correctamente [véase Código 23], para casos alternos se creó una función en la capa de Aplicación que permite descargar un reporte con los registros que no pudieron procesarse [véase Código 24]. Además, se creó una Vista endpoint que permite actualizar manualmente el estado de Concesión en una Beca [véase Código 25].

*Código 23. Fragmento de Código de la función en la capa de aplicación que se encarga de procesar el archivo a importar*

```
1 def procesar_datos(book, tipo):
2 #lee cada uno de las filas de la hoja de cálculo
3 ...
4 # validar que exista registros para procesar
5 if not registros_importar:
6 ...
7 # validar que no exista registro de devengado/pagos duplicados
8 (registros_importar, lista_duplicados, mensaje) = TesoreriaAppService.validar_duplicados(
9 registros_importar)
10 ...
11 # validar que se encuentre un registro de pago para cada beca
12 (registros_importar_registros_pagos, lista_registros_inconsistentes,
13 mensaje) = TesoreriaAppService.validar_existencia_registro_pago_beca(registros_importar, -
, ! periodo_academico_nombre)
14 ...
15 # validar que exista relación entre monto de pago y cantidad de meses
16 (registros_importar_registros_pagos, lista_registros_inconsistentes,
17 mensaje) = TesoreriaAppService.validar_relacion_monto_pago_mes(registros_importar_registros_pagos)
18 ...
19 # validar que no exista un registro de pago ya existente
20 (registros_importar_registros_pagos, lista_registros_inconsistentes,
21 mensaje) = TesoreriaAppService.validar_no_existencia_registro_pago_previo( -
, ! registros_importar_registros_pagos, tipo)
22 ...
23 # importar registros que cumplan con todas las validaciones
24 (lista_registros_importados, lista_registros_no_importados, mensaje) = TesoreriaAppService. -
, ! importar_registros(
25 registros_importar_registros_pagos, tipo)
26 ...
27 # devolver los registros que se pudieron procesar y los que no
28 ...
29 return (lista_registros_importados, mensaje, registros_no_procesados, errores_importacion, -
, ! total_importados, total_no_procesados, periodo_academico_nombre)
```

*Código 24. Fragmento de Código de la función en la capa de aplicación que permite descargar un reporte con los registros no procesados durante la importación*

```
1 def get_reporte_errores_importacion(request):
2 data = json.load(request)['json_data']
3 ...
4 cabeceras = ["NRO_IDENTIFICACION", "BECADO",
5 "NRO_EXPEDIENTE", "MESES", "CREDITO", "FECHA_TRANSACCION"]
6 ...
```

```

7 filas = [
8 [
9 item['dni'] if 'dni' in item else "",
10 item['becado'] if 'becado' in item else "",
11 ...
12 ]
13 for item in registros]
14 workbook = generar_excel(...)
15 workbook.save(response)
16 return response

```

*Código 25. Fragmento de Código del endpoint que permite actualizar manualmente el estado de Concesión de una Beca*

```

1 def actualizar_estado_concesion(request):
2 data = json.load(request)['json_data']
3 ...
4 with transaction.atomic():
5 for registro_id in data['registros_id']:
6 registro_pago = RegistroPago.objects.select_for_update(
7 ).get(pk=registro_id)
8 if registro_pago.estado == RegistroPagoEstados.SUSPENDIDO:
9 ...
10 if registro_pago.get_estado_display() == data['estado']:
11 ...
12 registroPagoEstados = RegistroPagoEstados()
13 if 'nro_expediente' in data:
14 ...
15 registro_pago.estado = registroPagoEstados.get_estado_by_nombre(
16 data['estado'])
17 if registro_pago.estado == RegistroPagoEstados.PAGADO and registro_pago.fecha_pagado is None:
18 ...
19 if registro_pago.estado == RegistroPagoEstados.DEVENGADO and registro_pago.fecha_devengado is None:
20 ...
21 registro_pago.save()
22
23 response_data.append({...})
24 return get_response(response_data, mensaje=mensaje)

```

A continuación, se puede visualizar las funcionalidades implementadas en la Etapa de Concesión.

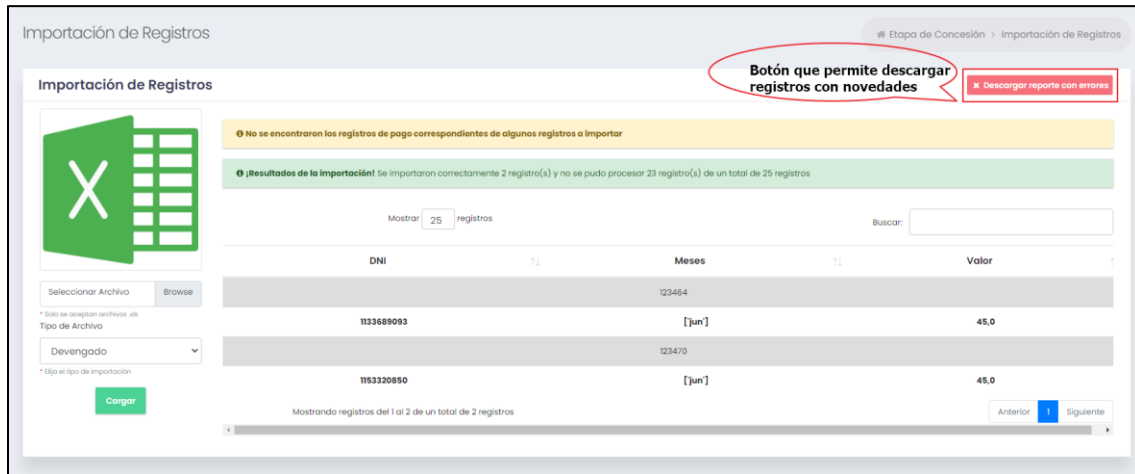


Figura 70. Template que permite importar los registros de Devengado o Pago de las Becas.

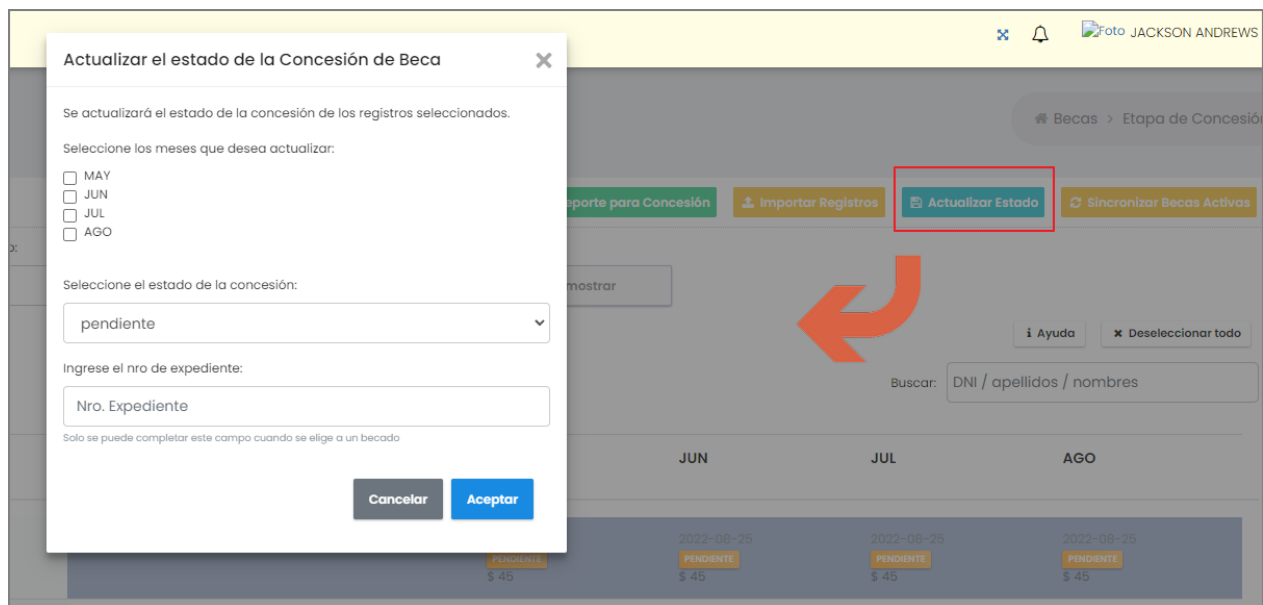


Figura 71. Ventana modal que permite actualizar manualmente el estado de Concesión de las Becas.

### 2.4.3.3 Sprint 3

El tercer Sprint estuvo conformado por el desarrollo de los siguientes requisitos funcionales (RF09, RF10, RF11, RF12, RF14) que se muestran en la Figura 72.

▼ Sprint 3 - PMV 3 18 feb. – 4 mar. (9 incidencias)	
Mostrar estadísticas generales y personalizar mensajes de notificaciones	
MGBUNL-23	Notificar cambio de estado de la Beca ETAPA DE ADJUDICACIÓN
MGBUNL-9	Mostrar estadísticas generales de postulantes ETAPA DE POSTULACIÓN
MGBUNL-25	Mostrar estadísticas generales de becados ETAPA DE ADJUDICACIÓN
MGBUNL-14	Guardar mensaje personalizado o frecuente para los cambios de... ETAPA DE REVISIÓN
MGBUNL-18	Registrar mensajes frecuentes ETAPA DE NOTIFICACIÓN
MGBUNL-19	Actualizar mensajes frecuentes ETAPA DE NOTIFICACIÓN
MGBUNL-20	Eliminar mensajes frecuentes ETAPA DE NOTIFICACIÓN
MGBUNL-21	Marcar como favorito un mensaje frecuente ETAPA DE NOTIFICACIÓN
MGBUNL-24	Enlazar becados activos del periodo lectivo anterior con el ... ETAPA DE ADJUDICACIÓN

Figura 72. Tareas planificadas para el Sprint 3 en Jira.

#### 2.4.3.3.1 RF09. Agregar Mensajes Frecuentes, RF10. Actualizar Mensajes Frecuentes, RF11. Eliminar Mensaje Frecuente, RF12. Marcar como favorito Mensaje Frecuente

Para estos requisitos funcionales se crearon vistas (GET, POST, PUT y DELETE) para poder agregar, actualizar, eliminar y marcar como favorito a los mensajes [véase Código 26 y Código 27].

Código 26. Fragmento de Código de la vista paginadora que lista los Mensajes Frecuentes

```

1 def mensaje_frecuente_paginador(request):
2     try:
3         mensajes_params = DataTableParams(request, **request.POST)
4         ConfiguracionesAppService.get_datatable(
5             mensajes_params=mensajes_params)
6         data = [{
7             'id': item.id,
8             'texto': item.texto,
9             ...
10        } for item in mensajes_params.items]
11        result = mensajes_params.result(data)
12        return JsonResponse(result)
13    except Exception as e:
14        ...

```

Código 27. Fragmento de Código de la Vista que permite crear, actualizar y eliminar Mensajes Frecuentes

```

1 def mensaje_frecuente(request: HttpRequest):
2     try:
3         data = json.load(request)['json_data']

```

```

4 mensajes_frecuentes = MensajeFrecuente.objects.filter(
5 tipo=data['tipo']).order_by('-orden')
6 if 'id' in data:
7 ...
8 ...
9 orden_maximo = mensajes_frecuentes.first(
10 ).orden if mensajes_frecuentes.exists() else 0
11 if orden_deseado > orden_maximo:
12 ...
13 if 'favorito' in data and bool(data['favorito']):
14 ...
15 for mensaje in mensajes_frecuentes:
16 if request.method == 'DELETE' and mensaje.orden > orden_deseado:
17 ...
18 if request.method == 'PUT' and mensaje.orden == orden_deseado:
19 ...
20 if request.method == 'POST' and mensaje.orden == orden_deseado:
21 ...
22 if mensajes_frecuentes.count() > 0:
23 ...
24 if request.method == 'POST':
25 ...
26 nuevo_mensaje_frecuente.save()
27 response_data = {...}
28 return get_response(mensaje="Mensaje agregado con exito", data=response_data)
29 if request.method == 'PUT':
30 ...
31 mensaje_frecuente.save()
32 response_data = {...}
33 return get_response(mensaje="Mensaje actualizado con exito", data=response_data)
34 if request.method == 'DELETE':
35 mensaje_frecuente.delete()
36 return get_response(mensaje="Mensaje eliminado con exito", data={'id': data['id']})
37 except Exception as e:
38 ...

```

En la figura a continuación se muestra la sección de Configuraciones desde donde se puede administrar los Mensajes Frecuentes de las Etapas [véase Figura 73].



Configuraciones para las Etapas

### Mensajes Frecuentes

Mensajes frecuentes para las Etapas

Tipo de Mensaje

Filtre los mensajes frecuentes por su tipo

Mostrar  registros

Buscar:

Orden	Mensaje Frecuente	Tipo
<input type="checkbox"/>	No se evidencia necesidades	RECHAZAR
<input type="checkbox"/>	suspendido ★	SUSPENDER
<input type="checkbox"/>	Éxitos en su ambiente profesional	TERMINAR
<input type="checkbox"/>	Bienvenido a la comunidad de Becados ★	APROBAR
<input type="checkbox"/>	Informamos que su beca a concluido y ya no recibirá más beneficios a partir de ahora ★	TERMINAR
<input type="checkbox"/>	Felicidades su beca ha sido aprobada	APROBAR
<input type="checkbox"/>	No alcanza la nota mínima requerida	RECHAZAR
<input type="checkbox"/>	Se encontró libando en las instalaciones de la UNL	SUSPENDER
<input type="checkbox"/>	Falta grave a las autoridades superiores	SUSPENDER
<input type="checkbox"/>	No cumple con los requisitos requeridos	RECHAZAR

Mostrando registros del 1 al 10 de un total de 11 registros. Clic en una fila para seleccionarla

Anterior **1** 2 Siguiente

Mensaje frecuente

Estos mensajes ayudarán a comprender el motivo de actualización de estado a los postulantes y becados

Tipo de Mensaje

- Aprobar Beca
- Rechazar Beca
- Reactivar Beca
- Suspender Beca
- Terminar Beca

Cada mensaje frecuente agregado debe pertenecer a un tipo, elija uno para el mensaje

Orden del mensaje

Escriba el orden para el tipo de mensaje

Mensaje Favorito

Sí

Marca el mensaje como favorito de entre el mismo tipo al que pertenece

Figura 73. Sección de Configuraciones para administrar los mensajes Frecuentes que se presentan en cada una de las Etapas.

#### 2.4.3.3.2 RF14 Visualizar estadísticas generales de postulantes y becados

En este requisito funcional se creó una Vista principal [véase Código 28] que a través de la capa de Aplicación se encarga de consultar cifras estadísticas de Becados y Postulantes de acuerdo a las siguientes especificaciones:

**Estadísticas de Becados:** se consulta el número total, del periodo actual, de acuerdo al tipo de beca, por cada periodo académico y por facultad.

**Estadísticas de Postulantes:** se consulta el número total, aprobados, rechazados y del periodo

académico actual.

**Estadísticas de Pagos:** se consulta los registros de pago, para obtener el balance general entre el monto pagado y por pagar. En el template [véase Figura 74] se construyeron tarjetas para representar las cifras y se usó la librería Chart.js para representar de manera gráfica las Becas activas por Periodo Académico y por Facultad.

*Código 28. Fragmento de Código de la Vista que permite obtener estadísticas generales de Postulantes y Becados*

```
1 def index_becas(request):
2 periodo_academico_actual = EtapasAppService.get_periodos_academicos_orden_desc().first()
3 becas = EstadisticasBecasAppService.get_becas()
4 # becas activas
5 becas_activas = EstadisticasBecasAppService.get_becas_activas(becas)
6 ...
7 # becas suspendidas
8 becas_suspendidas = EstadisticasBecasAppService.get_becas_suspendidas(
9 becas)
10 ...
11 # becas terminadas
12 becas_terminadas = EstadisticasBecasAppService.get_becas_terminadas(becas)
13 ...
14 # becas activas tipo A
15 becas_activas_tipo_a = becas_activas.filter(
16 beca_tipo_beca_codigo=TipoBecaCodigos.BECA_TIPO_A)
17 ...
18 # becas activas tibo B
19 becas_activas_tipo_b = becas_activas.filter(
20 beca_tipo_beca_codigo=TipoBecaCodigos.BECA_TIPO_B)
21 ...
22 # becas activas tibo C
23 becas_activas_tipo_c = becas_activas.filter(
24 beca_tipo_beca_codigo=TipoBecaCodigos.BECA_TIPO_C)
25 ...
26 # postulantes
27 postulantes = EstadisticasBecasAppService.get_postulantes()
28 ...
29 # postulantes aprobados
30 postulantes_aprobados = EstadisticasBecasAppService.get_postulantes_aprobados(
31 postulantes)
32 ...
33 # postulantes rechazados
34 postulantes_rechazados = EstadisticasBecasAppService.get_postulantes_rechazados(
```

```

35 postulantes)
36 ...
37 # balance general de becas
38 registros_pago = EstadisticasBecasAppService.get_registros_pago()
39 ...
40 return render(request, "becas/index_becas.html", locals())

```

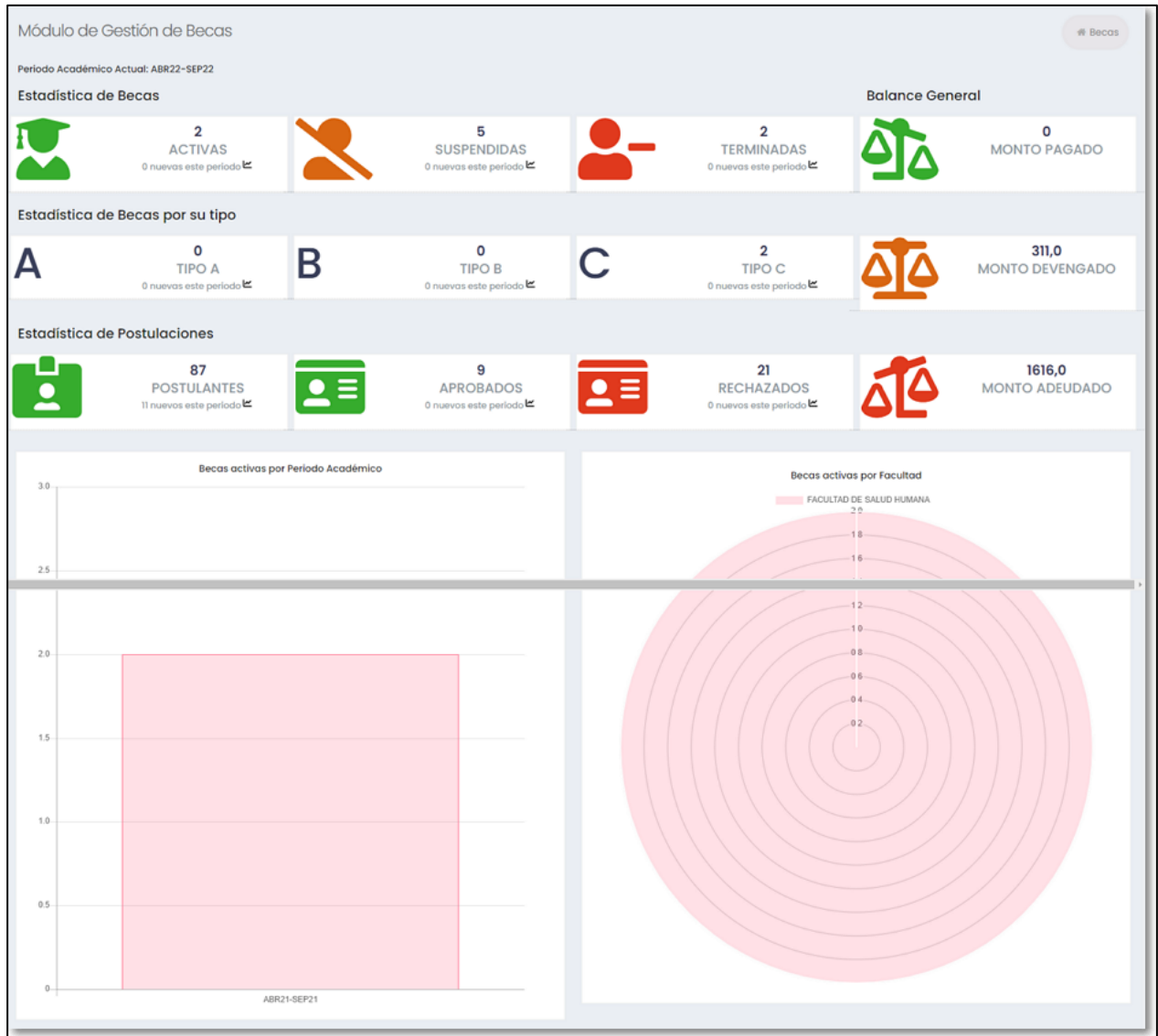


Figura 74. Template que muestra estadísticas generales de Postulantes y Becados.

### **3. Objetivo 3: Validar la implementación del módulo para el proceso de seguimiento y control de becas en un entorno de pruebas**

#### **3.1 Planificación de las Pruebas en un entorno simulado para la validación de funcionamiento del módulo desarrollado**

En la Planificación de las Pruebas para el MGB se elaboró un documento siguiendo el estándar ISO 29119<sup>4</sup> y se estableció el alcance de 6 tipos de pruebas: Pruebas de Caso de Uso, Pruebas de Carga y Estabilidad, Pruebas de Seguridad, Pruebas Unitarias, Pruebas de Código y Pruebas de Aceptación. En este documento quedaron establecidas las suposiciones y restricciones, las partes interesadas, el ambiente, las herramientas y el cronograma previsto para el Plan de Pruebas.

**(véase Plan de Pruebas para el MGB)**

#### **3.2 Ejecución del Plan de Pruebas en un entorno simulado**

Para la ejecución del Plan de Pruebas se elaboró un documento de acuerdo al estándar ISO 29119 para cada una de las pruebas consideradas y aprobadas en el Plan de Pruebas.

En este documento se describe el desarrollo de las pruebas ejecutadas al módulo presentando la evidencia que sustenta cada una de las pruebas. En el apartado anexos de este documento se adjuntó evidencia detallada de cada tipo de prueba, así como referencias a otros documentos como la Especificación de Requisitos de Software, Las Pruebas de Casos de Uso (a detalle), actas de reunión, encuestas de satisfacción previo al uso del software y posterior al uso del mismo, enlaces a sesiones grabadas de asistencia virtual a los usuarios y encuestas de aceptación aplicadas a los usuarios clientes.

**(véase Ejecución del Plan de Pruebas para el MGB)**

#### **3.3 Elaboración de la documentación necesaria para la implantación y uso del módulo desarrollado.**

Se consideró la elaboración de 2 manuales, el manual de usuario que especifica el uso de las funcionalidades desarrolladas de cara a los usuarios cliente [véase **Anexo 4 Manual de Usuario**

---

<sup>4</sup> [ISO/IEC/IEEE 29119 Software Testing Standard - Grupo de Trabajo AEN/CTN71/SC7/GT26 Pruebas de Software \(uniovi.es\)](http://www.iso.org/iso/standards/catalogue_tc/list_standards.html?csnumber=70687)

**para el uso del MGB]** y el manual técnico de implantación para su puesta en producción [véase **Anexo 5 Manual Técnico de Requisitos Básicos para la configuración del MGB]**. Estos manuales se elaboraron siguiendo el estándar especificado por el equipo de DTI para este tipo de documentos.

## **Anexo 2. Análisis del Reglamento de Becas**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Análisis del Reglamento de Becas



**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

[ANÁLISIS DE REGLAMENTO]

<<DIRECCIÓN DE BIENESTAR UNIVERSITARIO>>

<<SUBDIRECCIÓN DE CONTABILIDAD FINANCIERA>>

<<SUBDIRECCIÓN DE TESORERÍA FINANCIERA>>

# MÓDULO DE GESTIÓN DE BECAS - SIAAF

**Elaborado por:**

*Jackson Andrews Guzmán Tituana (tesista)*

Diciembre 2021

Página 1 de 5

Educamos para **Transformar**



# “REGLAMENTO DE LA SECCIÓN DE BECAS E INCENTIVOS DE BIENESTAR UNIVERSITARIO DE LA UNIVERSIDAD NACIONAL DE LOJA”

## CAPÍTULO III

### COMISIÓN DE BECAS E INCENTIVOS

Art. 5.- De la comisión de becas e incentivos. - La Comisión de Becas e Incentivos de la Institución estará integrada por:

- a. El Rector de la Institución o su delegado.
- b. El Director/a
- c. El Trabajador/a Social
- d. secretario/a
- e. Un representante del estamento estudiantil al Órgano Colegiado Superior

Art. 6.- Funciones de la Comisión de Becas e Incentivos. - La Comisión de Becas e Incentivos de la Institución tendrá las siguientes funciones:

- a. Establecer en cada periodo académico el cupo de estudiantes beneficiarios de becas e incentivos.
- b. Negar o rechazar las becas o incentivos de los estudiantes que no cumplan con el proceso establecido.
- c. Solicitar al Departamento Financiero de la Institución, una certificación presupuestaria.<sup>1</sup>
- d. Elaborar los formularios y formatos.<sup>2</sup>
- e. Resolver la suspensión y/o terminación del contrato de becas y ayudas económicas, estableciendo los valores a ser devueltos por el estudiante.<sup>3</sup>

Art. 7.- Funciones del responsable de la sección de becas e incentivos. - El responsable de la Sección de becas e incentivos tendrá las siguientes funciones:

- a. Informar, apoyar, asesorar.
- b. Realizar el proceso de selección de estudiantes beneficiarios, en forma conjunta con la Trabajadora Social.<sup>4</sup>
- c. Conocer los reclamos de los estudiantes beneficiarios y procurar su solución.<sup>5</sup>

Art. 8.- De los tipos de becas. - La Universidad Nacional de Loja, establece los siguientes tipos de becas:

---

<sup>1</sup> Notas de análisis: sería útil registrar el presupuesto con el que cuenta la UNL para cada periodo académico dentro del sistema.

<sup>2</sup> La plataforma debería aceptar estos formatos para que se complete con la información del estudiante.

<sup>3</sup> Suspender la beca dando a conocer sus motivos e indicando los reglamentos en los que se basa, así como informar la forma de devolver el dinero.

<sup>4</sup> La plataforma debe ser adecuada para poder seleccionar a los estudiantes.

<sup>5</sup> Módulo de gestión de quejas por parte de becados y creación de ticket con seguimiento de causa para su solución.





**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

- 1859 a. Beca Tipo “A” (Ayuda Socioeconómica).<sup>6</sup>  
b. Beca Tipo “B” (Estudiantes con Discapacidad).  
c. Beca Tipo “C” (Reconocimiento al Esfuerzo).

## CAPITULO VII

### DE LAS ETAPAS PARA EL PROCESO DE POSTULACIÓN

Esta postulación se realizará luego de iniciado el segundo periodo académico y por una sola vez en el año.

## CAPITULO VIII

### DEL CUPO PARA LAS BECAS

- La asignación de becas serán establecidos por la Comisión de Becas previo informe del responsable de becas.
- La distribución de estudiantes becarios será proporcional por Facultades, Carreras y al número.
- Los estudiantes que postulen por primera vez, se considerará una nota mínima de 8.5.<sup>7</sup>
- Fórmula para el cálculo de los estudiantes becarios.<sup>8</sup>
- La beca tipo A se otorgará por el lapso que dure la Carrera, a excepción de los estudiantes de la carrera de Medicina Humana, ciclos once y doce; de la carrera de Enfermería, ciclos siete y ocho; por percibir una remuneración otorgada por el Ministerio de Salud Pública.<sup>9</sup>

## CAPITULO IX

### REGISTRO, MANTENIMIENTO Y SEGUIMIENTO

El/la responsable de la Sección de Becas e Incentivos, implementará un archivo personal, se registrarán y actualizarán en el módulo creado por la Unidad de Telecomunicaciones e Informática para el efecto.

---

<sup>6</sup> Crear un módulo blog para registrar la información necesaria sobre cada uno de los tipos de becas.

<sup>7</sup> Ver viabilidad de consultar automáticamente la nota de los 2 últimos periodos.

<sup>8</sup> Considerar este cálculo en la selección de becados, revisar viabilidad de obtener número de matriculados en presencial.

<sup>9</sup> Considerar estas excepciones.



**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

## CAPITULO X

### OBLIGACIONES DEL ESTUDIANTE BECARIO

- a. Asistir normalmente a clases, cumpliendo el 80% de acuerdo a lo que especifica la normativa institucional.<sup>10</sup>
- b. Participar en actividades organizadas por la Unidad de Bienestar Universitario.
- c. Mantener un promedio de notas de acuerdo a lo que determina el presente Reglamento.
- d. Una vez culminado el periodo académico, deberá presentar el certificado de aprobación del ciclo inmediato anterior, legalizado por el Secretario Abogado de la Facultad.
- e. Presentar el certificado de encontrarse legalmente matriculado al ciclo inmediato superior, legalizado por el Secretario Abogado de la Facultad.
- f. Actualizar cada periodo académico los datos de la Ficha Socioeconómica a través del Sistema de Gestión Académica (SGA).

## CAPITULO XI

### INCENTIVOS<sup>11</sup>

La Unidad de Bienestar Universitario enviará a la Dirección Financiera de la Universidad Nacional de Loja, la nómina de beneficiarios, solicitando el pago o la suspensión del incentivo según corresponda.<sup>12</sup>

## CAPITULO XII

### DE LA SUSPENSIÓN Y PÉRDIDA DE LAS BECAS E INCENTIVOS

- a. Disminución del rendimiento académico y no alcanzar el promedio básico establecido en el presente reglamento.<sup>13</sup>
- b. Pérdida o retiro del ciclo académico.
- c. Renuncia voluntaria del beneficiario.

---

<sup>10</sup> Considerar recordar a los estudiantes estos puntos una vez terminado el periodo académico.

<sup>11</sup> Reorganizar la información en un blog.

<sup>12</sup> Consultar la viabilidad utilizando las TIC 's.

<sup>13</sup> Considerar esta información en el blog.



UNL

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

1859 d. En el caso de comprobarse falsedad en alguno de los datos o documentos entregados al momento de solicitar la beca.

e. Por faltas disciplinarias.

f. Al egresar de la carrera que estuvo cursando al obtener la beca o incentivo.

Art 39.- De la terminación del contrato de Becas e Incentivos. - La Universidad Nacional de Loja, podrá dar por terminada la Beca o Incentivo en los siguientes casos:

a. Por mutuo acuerdo de las partes.

b. Por muerte o ausencia definitiva del beneficiario.

c. Por indisciplina del beneficiario; por faltas cometidas en contra de cualquier miembro de la Comunidad Universitaria.

d. Por encontrarse en estado etílico dentro del campus universitario.

e. Por renuncia voluntaria.

DISPOSICIONES GENERALES: La Sección de Becas e Incentivos de la Universidad Nacional de Loja, tendrá acceso al Sistema Informático de Gestión Académica Estudiantes, para verificar notas y asistencias de los estudiantes becarios.<sup>14</sup>

---

<sup>14</sup> Revisar posibilidad de conexión entre módulos

**Anexo 3. Especificación de Requisitos de Software en base a la norma del IEEE 830 para el MGB**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Especificación de Requisitos de Software<sup>5</sup>

---

<sup>5</sup> Versión original en: <https://drive.google.com/ERS-MGB.pdf>



Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

## Especificación de requisitos de software

# Módulo de Seguimiento y Control de Becas para la Unidad de Bienestar Universitario

Revisión 2022.02

**Elaborado por:**

*Jackson Andrews Guzmán Tituana*

**Revisado por:**

*Ing. Máximo Álvarez*

**Aprobado por:**

*Ing. Danny Muñoz*

*Ing. Pablo F. Ordoñez, Mg.Sc.*

*Ing. Máximo Álvarez*

*Licda. Marlene Valdez*

Febrero.2022



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### TABLA DE CONTENIDO

<b>Historial de versiones</b>	<b>2</b>
<b>Información del Proyecto</b>	<b>3</b>
<b>Introducción</b>	<b>3</b>
1.1 Propósito	3
1.2 Alcance del Producto	4
1.3 Referencias	4
1.4 Personal Involucrado	5
1.5 Glosario de términos	8
Definiciones, Acrónimos y abreviaturas	8
Del negocio	8
Del sistema	9
1.6 Resumen	9
<b>Descripción General</b>	<b>9</b>
2.1 Perspectiva	9
2.2 Objetivos del producto	10
Objetivo General	10
Objetivos Específicos	10
2.3 Roles de usuario	11
2.4 Restricciones	12
2.5 Suposiciones y dependencias	12
<b>Requerimientos de alto nivel</b>	<b>13</b>
3.1 Interfaces de Usuario	13
3.2 Requisitos Funcionales	14
Requisito funcional sincronizar datos de postulación desde el MGT	15
Requisito funcional actualizar estados de postulaciones	16
Requisito funcional descargar documentos de postulación	19
Requisito funcional ver trámite en el MGT	20
Requisito funcional finalizar Etapas	21
Requisito funcional actualizar estados de becas	23
Requisito funcional visualizar postulaciones y becas	25
Requisito funcional generar reportes	28
Requisito funcional agregar mensaje frecuente	30
Requisito funcional actualizar mensaje frecuente	32
Requisito funcional eliminar mensaje frecuente	35



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

Requisito funcional marcar como favorito mensaje frecuente	37
Requisito funcional actualizar montos de pago para los tipos de beca	39
Requisito funcional visualizar estadísticas generales de becados	41
Requisito funcional iniciar sesión	43
Requisito funcional notificar resultados de postulación	44
Requisito funcional guardar estado de registro del nuevo becado en el sistema financiero	45
Requisito funcional registrar pago de becas	47
<b>3.3 Requerimientos no funcionales</b>	<b>48</b>
3.1 Requisitos de rendimiento	48
3.2 Seguridad	48
3.3 Fiabilidad	49
3.4 Disponibilidad	49
3.5 Mantenibilidad	49
<b>4. Entorno Operativo</b>	<b>49</b>
4.1 Funcionamiento de auditoría	49
4.2 De control de acceso	50
<b>Historias de usuario</b>	<b>50</b>
<b>Aprobaciones</b>	<b>65</b>



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### Historial de versiones

Versión	Fecha	Responsable	Descripción de la modificación
1.0	15 de Diciembre 2021	Tesista Jackson Guzmán	Incluir caso de uso y prototipado por cada requisito funcional para que ayude a comprender mejor cada requisito por parte de los interesados. Sugerido por el Director de Tesis Ing. Pablo ordoñez
1.5	05 de Enero de 2022	Tesista Jackson Guzmán	Simplificación de casos de uso y unión de casos de uso similares. Análisis de flujos alternos. Algunos casos de uso pueden unirse ya que poseen un flujo similar. Considerar más flujos alternos y precondiciones. Revisor externo: Sugerido por: Ing. María Ruilova
2.0	11 de enero del 2022	Tesista Jackson Guzmán	Sugerencias de sintaxis y puntualización de requerimientos. Ciertas expresiones pueden confundir, se dieron sugerencias y recomendaciones para su mejor entendimiento. Sugerencias de la Dirección de Telecomunicaciones e Información: Ing. Danny Muñoz e Ing. Máximo Álvarez

### Información del Proyecto

<b>Unidad requirente</b>	Unidad de Bienestar Universitario
<b>Nombre del proyecto</b>	Módulo de Seguimiento y Control de Becas para la Unidad de Bienestar Universitario
<b>Fecha de elaboración</b>	21 de Febrero de 2022
<b>Solicitante / Patrocinador principal</b>	Unidad de Bienestar Universitario





## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

---

### 1. Introducción

El proyecto "**Módulo para el Seguimiento y Control de Becas de la Unidad de Bienestar Universitario**" tiene como objetivo principal mejorar la gestión de becas en la universidad a través de la implementación de un módulo informático en el marco del Sistema de Información Académico Administrativo Financiero (SIAAF). Este módulo permitirá gestionar las becas desde la etapa de revisión hasta la concesión, lo que se traducirá en una mayor eficiencia y transparencia en el proceso de selección y asignación de becas a los estudiantes universitarios. En este documento se definirán los requerimientos necesarios para el desarrollo del módulo y se establecerán las bases para su implementación exitosa.

#### 1.1 Propósito

El desarrollo de módulo informático para el Seguimiento y Control de Becas se realiza con el fin de cumplir el objetivo general del trabajo de titulación denominado "**Módulo para el Seguimiento y Control de Becas para la Unidad de Bienestar Universitario**" desarrollado por el tesista Jackson Guzmán previo a la obtención del título de Ingeniero en Sistemas. Tesis la cuál pretende presentar una solución de software en la Gestión de Becas de la Unidad de Bienestar Universitario automatizando los procesos existentes que ayude al Responsable de la Sección de Becas e Incentivos en el manejo de información tratando de evitar la asimetría de los datos.

#### 1.2 Alcance del Producto

El Módulo informático denominado a partir de ahora como **MGB** siglas para **Módulo para la Gestión de Becas** perteneciente al **SIAAF**, permitirá la gestión de postulantes a partir de la información obtenida desde el **Módulo de Gestión de Trámites (MGT)** durante la Etapa de Postulación.

A partir de la Etapa de Revisión con base a la información proporcionada por los postulantes, el módulo informático posibilitará el registro de la aprobación o rechazo de postulaciones y la asignación de calificación para la ficha socioeconómica. Permitirá además, llevar el registro de los postulantes que aprueban o no la Etapa de Selección según el análisis realizado por la comisión de becas. Una vez que los postulantes hayan pasado la Etapa de Selección forman parte de los estudiantes becados y deben ser gestionados con ayuda del MGB; gestión para los beneficiarios de las **becas de tipo A, B y C**, ayudando al control de la información en: la prohibición de poseer más de una beca, la suspensión y terminación de la beca con sus respectivas notificaciones, obtener información del ciclo actual del estudiante y la nota del ciclo inmediato anterior; así como del registro y controles de pagos. Además hará posible la generación de reportes que



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

se consideren necesarios de acuerdo a los actuales procesos manuales que se llevan a cabo.

Por otra parte cabe aclarar lo que el **MGB** no hará. Las actividades que no se contemplan en el módulo informático son: la Etapa de Postulación ya que esta se lleva a cabo mediante el **MGT** en donde se recepta y filtra la información proporcionada por los postulantes. El módulo informático tampoco se encargará del análisis y criterios de selección que son provistos por la Comisión de Becas durante la Etapa de Selección ya que la misma es un proceso de carácter manual y revisiones de la documentación proporcionada por los postulantes a través del MGT. Así también, tampoco se encarga de la realización de pagos, ya que este proceso se lleva a cabo por el departamento financiero a través del Sistemas Financiero.

### 1.3 Referencias

- REGLAMENTO DE LA SECCIÓN DE BECAS E INCENTIVOS DE BIENESTAR UNIVERSITARIO DE LA UNIVERSIDAD NACIONAL DE LOJA, aprobado en primera sesión extraordinaria de 18 de diciembre de 2020; y, en segunda y definitiva, en sesión extraordinaria de 8 de febrero de 2021.

### 1.4 Personal Involucrado

**Tabla 1.4.1: Director de Tesis**

<b>Nombre</b>	Pablo F. Ordoñez Ordoñez
<b>Rol</b>	Director de Tesis
<b>Categoría profesional</b>	Docente Director
<b>Responsabilidades</b>	Dirigir el trabajo de titulación de acuerdo al cronograma de trabajo.
<b>Información de contacto</b>	pfordonez@unl.edu.ec



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

**Tabla 1.4.2: Especialista de Sistemas de Información DTI**

<b>Nombre</b>	Danny Emanuel Muñoz Flores
<b>Rol</b>	Scrum Master
<b>Categoría profesional</b>	Ingeniero en Sistemas
<b>Responsabilidades</b>	Dirigir el desarrollo del proyecto coordinando sus actividades.
<b>Información de contacto</b>	dmunoz@unl.edu.ec

**Tabla 1.4.1: Analista de Sistemas de Información DTI**

<b>Nombre</b>	Máximo Álvarez
<b>Rol</b>	Product Owner
<b>Categoría profesional</b>	Ingeniero en Sistemas
<b>Responsabilidades</b>	Priorizar y guiar las actividades de desarrollo del proyecto de acuerdo al cronograma de trabajo establecido.
<b>Información de contacto</b>	maalvarezp@unl.edu.ec

**Tabla 1.4.1: Tesista**

<b>Nombre</b>	Jackson Andrews Guzmán Tituana
<b>Rol</b>	Developer
<b>Categoría profesional</b>	Estudiante Tesista



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Responsabilidades</b>	Analizar, Diseñar y Desarrollar el Módulo de Seguimiento y Control de Becas para la UBU
<b>Información de contacto</b>	jackson.guzman@unl.edu.ec

**Tabla 1.4.1: Directora Unidad Bienestar Universitario**

<b>Nombre</b>	Cecilia Rocío Ruíz Toledo
<b>Rol</b>	Stakeholder
<b>Categoría profesional</b>	Trabajadora Social
<b>Responsabilidades</b>	Revisar y aprobar el módulo informático
<b>Información de contacto</b>	cecilia.ruiz@unl.edu.ec

**Tabla 1.4.1: Trabajadora social - Responsable de la Sección de Becas**

<b>Nombre</b>	Marlene Valdez Pardo
<b>Rol</b>	Stakeholder
<b>Categoría profesional</b>	Trabajadora Social
<b>Responsabilidades</b>	Proporcionar información y evaluar la calidad del producto.
<b>Información de contacto</b>	marlene.valdez@unl.edu.ec



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

**Tabla 1.4.1: Subdirección de Tesorería - Responsable de Pagos**

<b>Nombre</b>	Mónica Raquel Mora Naula
<b>Rol</b>	Stakeholder
<b>Categoría profesional</b>	Contabilidad y Auditoría
<b>Responsabilidades</b>	Proporcionar información y evaluar la calidad del producto.
<b>Información de contacto</b>	monica.r.mora@unl.edu.ec

**Tabla 1.4.1: Departamento Financiero - Contador**

<b>Nombre</b>	Juan Sebastián Varela Mena
<b>Rol</b>	Stakeholder
<b>Categoría profesional</b>	Contabilidad y Auditoría
<b>Responsabilidades</b>	Proporcionar información y evaluar la calidad del producto.
<b>Información de contacto</b>	jvarela@unl.edu.ec

## 1.5 Glosario de términos

### Definiciones, Acrónimos y abreviaturas

**DTI:** Dirección de Tecnologías de Información

**ERS:** Especificación de Requisitos de Software

**MGT:** Módulo de Gestión de Trámites

**MGB:** Módulo de Gestión de Becas

**RF:** Requisito Funcional

**RFN:** Requisito no Funcional



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

---

**SIAAF:** Sistema de Información Académico Administrativo Financiero

**UNL:** Universidad Nacional de Loja

**UBU:** Unidad de Bienestar Universitario

### Del negocio

**Postulante.-** Estudiante que aplica a una determinada beca.

**Becado.-** Estudiante al cuál se le aprobó la postulación de beca.

**Etapas de Postulación.-** Período en el que se reciben aplicaciones a becas.

**Etapas de Revisión.-** Período en el que se revisa la información adjuntada por los postulantes.

**Etapas de Selección.-** Período en el que la Comisión de Becas se encarga de evaluar y analizar la situación de los postulantes.

**Etapas de Adjudicación.-** Período en el que se da seguimiento a las becas.

**Etapas de Concesión.-** Período en el que se registran los pagos realizados a los becados.

**Beca Tipo A.-** Ayuda Socioeconómica.

**Beca Tipo B.-** Estudiante con discapacidad.

**Beca Tipo C.-** Reconocimiento al esfuerzo.

### Del sistema

**Framework.-** Conjunto de herramientas y metodologías para desarrollo de software, que permitieron que el desarrollo sea más rápido y con funcionalidades que reducen el tiempo de programación.

**Base de datos.-** Conjunto de datos almacenados, ordenados y listos a ser utilizados por el software, con llamadas desde el software, interactuando de manera dinámica.

**Templates.-** Plantilla, para la interfaz gráfica de la aplicación que serán llamadas y ejecutarán código html dentro del software.

**Api.-** Es la interfaz de programación de aplicaciones, es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de las aplicaciones.

**Views.-** Métodos que permitirán realizar toda la lógica del sistema como guardar, modificar o eliminar registros, además de validar permisos y demás funcionalidades en torno a la seguridad y utilidades del sistema.

**Modelos.-** Manipulación de la información, por medio del orm object-relational mapping de Django, se transformará un esquema orientado a objetos a un modelo relacional en la base de datos.



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

---

### 1.6 Resumen

Este documento fue elaborado con el fin de detallar todas las consideraciones técnicas en el desarrollo del módulo informático, para el Seguimiento y Control de Becas de la Unidad de Bienestar Universitario. Está organizado en 3 secciones principales, en la primera sección se describe de manera general el propósito, alcance y personal involucrado en el desarrollo de este proyecto. En la segunda sección se describen los objetivos que se espera alcanzar, las características de los actores que usarán el módulo informático, las restricciones, suposiciones y dependencias que posee. Finalmente en la tercera sección se presenta cada uno de los requisitos funcionales (RF) y requisitos no funcionales (RNF) que se van a implementar en el desarrollo del Módulo de Gestión de Becas.

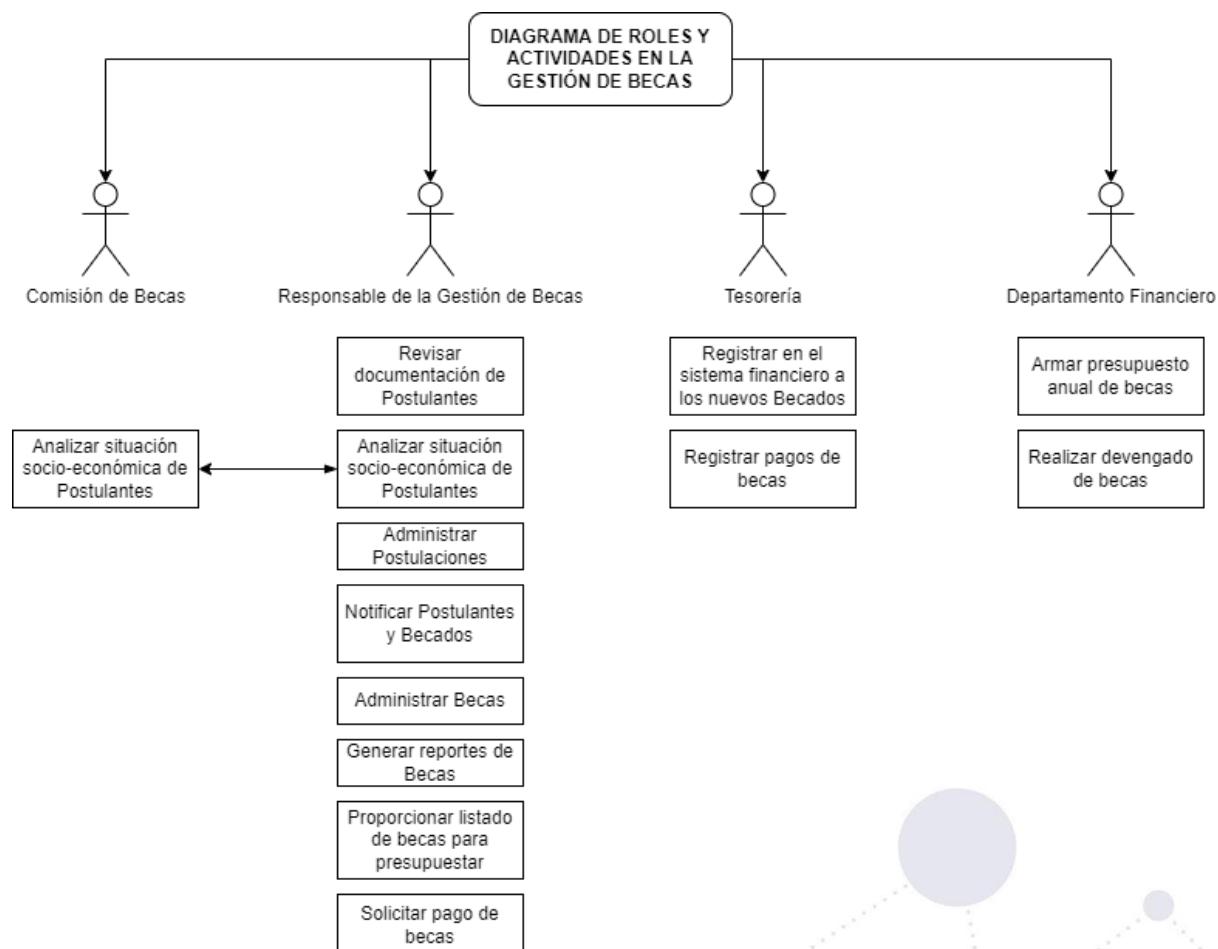
## 2. Descripción General

### 2.1 Perspectiva

El módulo informático MGB tiene el propósito de ayudar en la Gestión de Becas de la UBU-UNL, si bien trabaja de manera independiente requiere de la información obtenida a través del MGT en el que se recepta la información de los postulantes. En la figura 1 se detallan los roles y actividades del personal que interviene en la Gestión de Becas.



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE



**Figura 1.** Diagrama de roles y actividades del personal que interviene en la gestión de becas (Fuente propia)

## 2.2 Objetivos del producto

### Objetivo General

- Coadyuvar en el proceso de Seguimiento y Control de Becas estudiantiles para la Unidad de Bienestar Universitario de la Universidad Nacional de Loja.

### Objetivos Específicos

- Administrar la información de los becarios.
- Notificar acerca de las actualizaciones de estado a los becados.
- Registrar y controlar los pagos de los becarios.
- Obtener reportes de postulantes y becarios.





## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### 2.3 Roles de usuario

**Tabla 2.3.1: Usuario Administrador de Becas**

<b>Tipo de Usuario</b>	Responsable de la Sección de Becas de la UBU - UNL
<b>Formación</b>	Trabajadora Social
<b>Actividades</b>	Revisión de documentación de postulantes. Aprobación de las Etapas de Revisión y Selección. Asignación de calificación de ficha socioeconómica. Suspensión de Becas. Terminación de Becas. Generación de reportes.

**Tabla 2.3.2: Usuario Tesorería**

<b>Tipo de Usuario</b>	Responsable del registro y realización de pagos Tesorería - UNL
<b>Formación</b>	Contabilidad
<b>Actividades</b>	Confirmación de Registro en el Sistema Financiero. Registro de Pagos de los becados en el MGB. Generación de reportes. Importar registros de pagos.

**Tabla 2.3.3: Usuario Departamento Financiero**

<b>Tipo de Usuario</b>	Responsable del registro de devengados
<b>Formación</b>	Contabilidad
<b>Actividades</b>	Importar registros de devengado



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

**Tabla 2.3.3: Usuario Administrador SIAAF**

<b>Tipo de Usuario</b>	Responsable de la Seguridad de Información
<b>Formación</b>	Ingeniero en Sistemas
<b>Actividades</b>	Vigilancia de los datos. Administración de los datos de la gestión de becas Dar de alta/baja nuevos usuarios

### 2.4 Restricciones

- El desarrollo del módulo informático fue basado en la presente ERS recopilada a través de las entrevistas realizadas a los interesados y en base al REGLAMENTO DE LA SECCIÓN DE BECAS E INCENTIVOS DE BIENESTAR UNIVERSITARIO DE LA UNIVERSIDAD NACIONAL DE LOJA.
- El módulo informático forma parte del SIAAF.
- Si el módulo informático por parte del tesista no está completo en un 100% de acuerdo al tiempo planificado, la documentación quedará completa para la continuación del desarrollo por parte del Scrum Team de DTI.
- Se requiere el entorno de desarrollo del SIAAF para ejecutar el módulo informático.

### 2.5 Suposiciones y dependencias

El módulo informático depende de la información de postulantes que le proporciona el Módulo de Gestión de Trámites (MGT), misma que es esencial para su manejo en el MGB, así como poder consultar los documentos enviados por los postulantes a través del MGT.

El módulo informático requiere de navegadores basados en Chromium y navegadores basados en el motor Gecko de la fundación Mozilla, lo que lo convierte en un módulo informático multiplataforma, que puede ser accedido desde cualquier sistema operativo. Depende de la conexión a internet para manejar la información del módulo.



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

---

### 3. Requerimientos de alto nivel

#### 3.1 Interfaces de Usuario

La interfaz de usuario del módulo informático contará con elementos comunes que permitan una comunicación consistente para el usuario considerando que sea fácil de usar. Estos elementos comunes están compuestos por barra lateral o sidebar, botones, tablas, cuadros de entrada de texto, ventanas emergentes o modals, mensajes de alerta y tarjetas de información, tratando en la medida de lo posible mantener la consistencia para su fácil entendimiento y adaptabilidad al uso del módulo informático. Cabe destacar que la línea gráfica está sujeta a las directrices provistas por los templates del SIAAF por lo que se considera únicamente la estructura del prototipo propuesto, más no su línea gráfica.

ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

3.2 Requisitos Funcionales

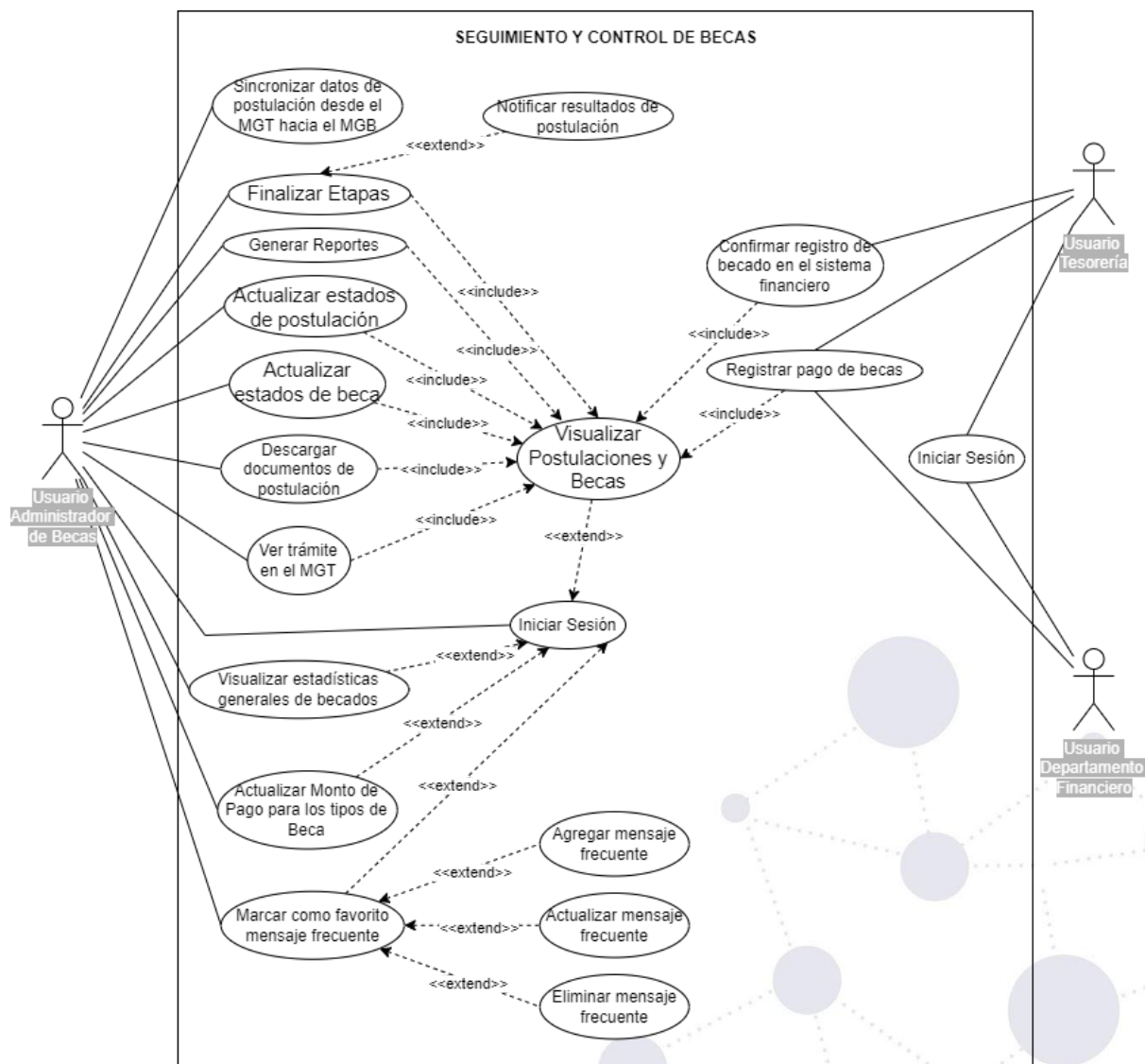


Figura 2. Casos de Uso General para el Seguimiento y Control de Becas (Fuente Propia).

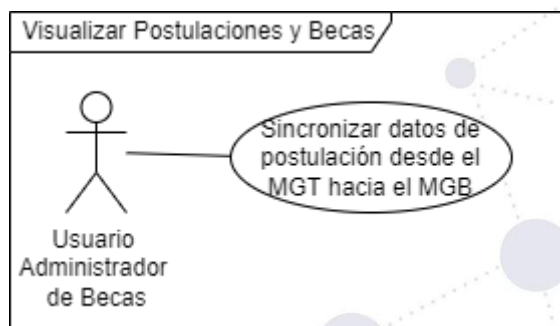
Tabla 3.2.1: Resumen de Requisitos Funcionales

Identificador	Nombre
RF01	Sincronizar datos de postulación desde el MGT hacia el MGB
RF02	Actualizar estados de postulaciones
RF03	Descargar documentos de postulación

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

RF04	Ver trámite en el MGT
RF05	Finalizar Etapas
RF06	Actualizar estados de Becas
RF07	Visualizar Postulaciones y Becas
RF08	Generar Reportes
RF09	Agregar mensaje frecuente
RF10	Actualizar mensaje frecuente
RF11	Eliminar mensaje frecuente
RF12	Marcar como favorito mensaje frecuente
RF13	Actualizar montos de pago para los tipos de beca
RF14	Visualizar estadísticas generales de becados
RF15	Iniciar Sesión
RF16	Notificar resultados de postulación
RF17	Guardar registro de becado en el sistema financiero
RF18	Registrar Pago de Becas

### Requisito funcional sincronizar datos de postulación desde el MGT

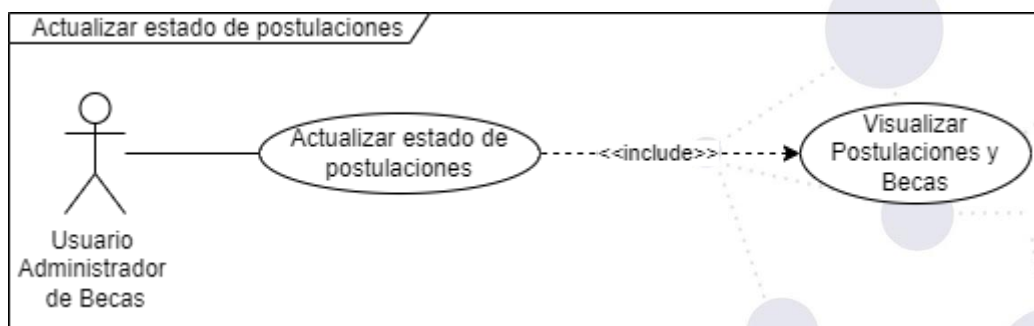


<b>Identificador</b>	CU01
<b>Nombre</b>	Sincronizar datos de postulación desde el MGT hacia el MGB
<b>Descripción</b>	El módulo informático permitirá cargar la información que se procesa en el MGT para realizar una copia de la información indispensable y poderla usar dentro del MGB.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Debe existir un periodo académico activo.</li> <li>• El Usuario Administrador de Becas deberá iniciar sesión en el MGT.</li> </ul>

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	<ul style="list-style-type: none"> <li>El Usuario Administrador de Becas debe estar en el apartado de Configuración, componente de postulación de becas.</li> </ul>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>El Usuario Administrador de Becas, debe seleccionar la Fase 4: Revisión y validación de requisitos.</li> <li>El Usuario Administrador de Becas, debe seleccionar el periodo de postulación a sincronizar.</li> <li>El Usuario Administrador de Becas presiona el botón sincronizar.</li> <li>El MGT llama al servicio API del MGB para registrar los postulantes.</li> </ol>
<b>Flujo Alternativo</b>	<p>2.1 No existen postulantes para el periodo académico seleccionado.</p> <p>2.1 El MGB devuelve un mensaje indicando que no existen datos para el periodo seleccionado.</p> <p>4.1 El MGT no puede establecer conexión con el MGB.</p>
<b>Requisito</b>	RF01

### Requisito funcional actualizar estados de postulaciones



<b>Identificador</b>	CU02
<b>Nombre</b>	Actualizar estados de postulaciones
<b>Descripción</b>	El módulo informático permitirá la actualización del <u>estado de los postulantes</u> en las Etapas de Revisión y Selección.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>Debe existir un periodo académico activo.</li> <li>El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>Debe existir datos de postulaciones.</li> </ul>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>El Usuario Administrador de Becas selecciona el registro del postulante que va a actualizar el estado.</li> </ol>





## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	<p>2. El Usuario Administrador de Becas actualiza el estado del postulante.</p> <p>3. El sistema actualiza el estado de la postulación.</p>
<b>Flujo Alternativo</b>	<p>2.1. Si el Usuario Administrador de Becas selecciona el estado Aprobado o Rechazado en la Etapa de Revisión, el sistema solicita agregar una calificación a la ficha socioeconómica.</p> <p>2.2. El Usuario Administrador de Becas proporciona una calificación a la ficha socioeconómica.</p> <p>2.2.1. Si se trata del estado Rechazado ya sea en la Etapa de Revisión o Selección o el estado de Aprobado en la Etapa de Selección, el sistema solicitará describir los motivos.</p> <p>2.2.2. Si no se describen los motivos, el sistema tomará el mensaje favorito para dicha acción.</p> <p>2.2.2.1. Si el sistema no tiene configurado un mensaje favorito, mostrará una advertencia solicitando configurar un mensaje favorito en el apartado de configuraciones.</p> <p>2.3. El Usuario Administrador de Becas presiona en el botón guardar notificación.</p> <p>2.4. El sistema guarda la notificación.</p> <p>3.1. El sistema no actualiza el estado debido a un problema de comunicación.</p> <p>3.2. El sistema muestra un mensaje de que no se ha podido guardar la información, por lo que el Usuario Administrador de Becas debe volver a intentarlo.</p>
<b>Requisito</b>	RF02

### **Páginas de Revisión y Selección de Postulantes - actualizar estado de postulante**

<b>Nro de página</b>	03 y 04		
<b>Estado de Pruebas</b>	<b>Aceptada cambios</b>	<b>sin</b>	si
	<b>Aceptada cambios</b>	<b>con</b>	
	<b>Rediseñar</b>		
<b>Requisito Funcional</b>	RF2		
<b>Observaciones</b>			
Aprobar y Rechazar Postulación en la Etapa de Revisión			

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE



**REVISIÓN RECHAZADA**

CALIFICACIÓN FICHA SOCIO-ECONÓMICA: 0

Por favor detalle a continuación el motivo por la que el postulante es RECHAZADA, es mensaje se utilizará para dar a conocer al postulante el motivo del RECHAZO de su postulación.  
El mensaje que se proporcione a continuación se enviará al postulante con el siguiente correo: **correo@unl.edu.ec**

Detalle claramente las razones por las que el postulante es RECHAZADO

**GUARDAR NOTIFICACIÓN**

Aprobar y Rechazar Postulación en la Etapa de Selección



**SELECCIÓN RECHAZADA**

CALIFICACIÓN FICHA SOCIO-ECONÓMICA: 8.6

Por favor detalle a continuación el motivo por la que el postulante es RECHAZADO, este mensaje se utilizará para dar a conocer al postulante el motivo del RECHAZO de su postulación.  
El mensaje que se proporcione a continuación se enviará al postulante con el siguiente correo: **correo@unl.edu.ec**

Detalle claramente las razones por las que el postulante es RECHAZADO

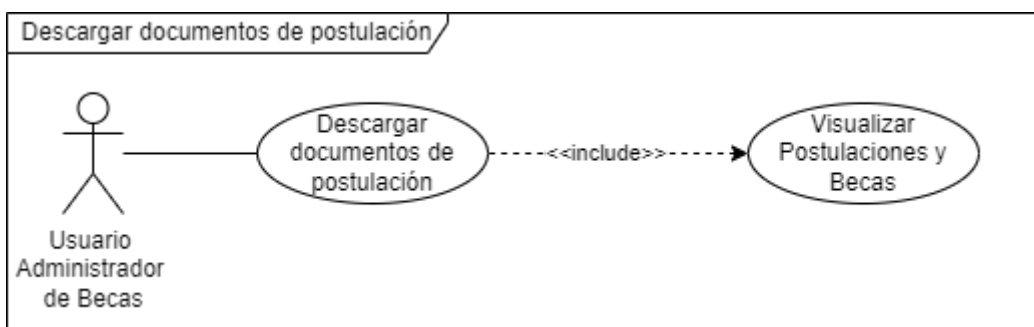
**GUARDAR NOTIFICACIÓN**

Nota: se utilizan la misma ventana emergente, lo que cambia es la acción.



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### Requisito funcional descargar documentos de postulación




<b>Identificador</b>	CU03
<b>Nombre</b>	Descargar documentos de postulación
<b>Descripción</b>	El módulo informático permitirá al Usuario Administrador de Becas descargar la documentación de los postulantes durante la Etapa de Revisión y Selección.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>• El Usuario Administrador de Becas debe estar en la Etapa de Revisión o en la Etapa de Selección.</li> <li>• Debe existir datos de postulaciones.</li> </ul>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El Usuario Administrador de Becas selecciona uno o varios postulantes.</li> <li>2. El Usuario Administrador de Becas presiona el botón descargar.</li> <li>3. El sistema descarga los documentos solicitados.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>1.1 El Usuario Administrador de Becas no selecciona ningún postulante.</li> <li>1.2 El Usuario Administrador de Becas presiona el botón descargar.</li> <li>1.3 El sistema no encuentra los archivos solicitados.</li> <li>2.1 El sistema no encuentra los archivos solicitados.</li> <li>2.2 El sistema no descarga los archivos y pide volver a intentar más tarde.</li> </ol>
<b>Requisito</b>	RF03

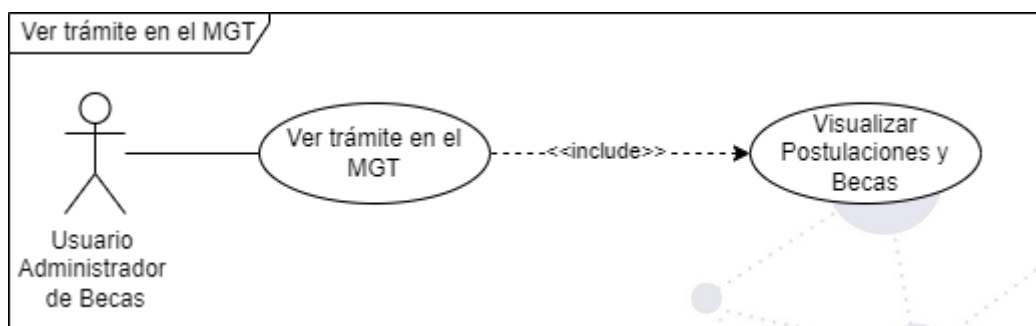
### Página Revisión de Postulantes - descargar documentos de postulación

<b>Nro de página</b>	03
----------------------	----

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Estado de Pruebas</b>	<b>Aceptada cambios</b>	<b>sin</b>	si
	<b>Aceptada cambios</b>	<b>con</b>	
	<b>Rediseñar</b>		
<b>Requisito Funcional</b>	RF03		
<b>Observaciones</b>	La opción de descarga de documentos estará disponible en la Etapa de Revisión y Selección.		
			


### Requisito funcional ver trámite en el MGT



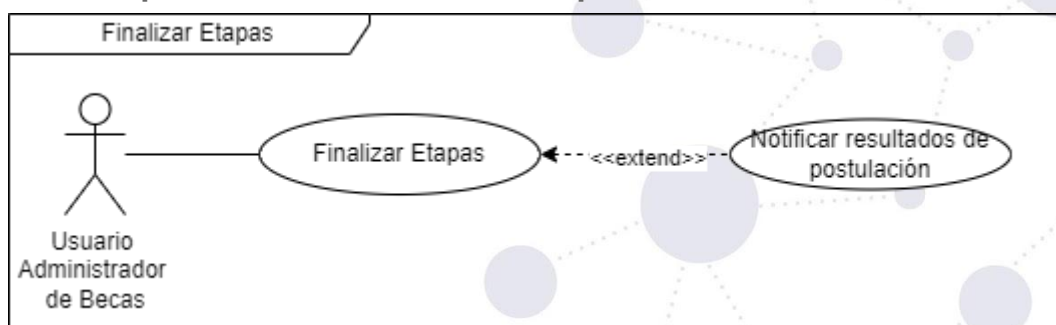
<b>Identificador</b>	CU04
<b>Nombre</b>	Ver trámite en el MGT
<b>Descripción</b>	El módulo informático permitirá al Usuario Administrador de Becas acceder al trámite desde el MGB, para lo cual tendrá un enlace directo al trámite que le corresponde a la postulación seleccionada.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Debe existir un periodo académico activo.</li> <li>• El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>• El Usuario Administrador de Becas debe estar en la Etapa de Revisión o en la Etapa de Selección.</li> <li>• El Usuario Administrador de Becas selecciona un postulante.</li> <li>• Debe existir datos de postulaciones.</li> </ul>

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El Usuario Administrador de Becas presiona el botón ver trámite.</li> <li>2. El sistema abrirá una nueva pestaña del navegador que dirige al trámite del postulante seleccionado.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>1. El Usuario Administrador de Becas no selecciona ningún postulante.</li> <li>2. El sistema no activará el botón de ver trámite.</li> </ol>
<b>Requisito</b>	RF04

Página Revisión de Postulantes - ver trámite en el MGT			
<b>Nro de página</b>	03		
<b>Estado de Pruebas</b>	<b>Aceptada cambios</b>	<b>sin</b>	si
	<b>Aceptada cambios</b>	<b>con</b>	
	<b>Rediseñar</b>		
<b>Requisito Funcional</b>	RF04		
<b>Observaciones</b>	La opción de ver trámite estará disponible en las Etapas de Revisión y Selección.		
			

### Requisito funcional finalizar Etapas



<b>Identificador</b>	CU05
<b>Nombre</b>	Finalizar Etapas




## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

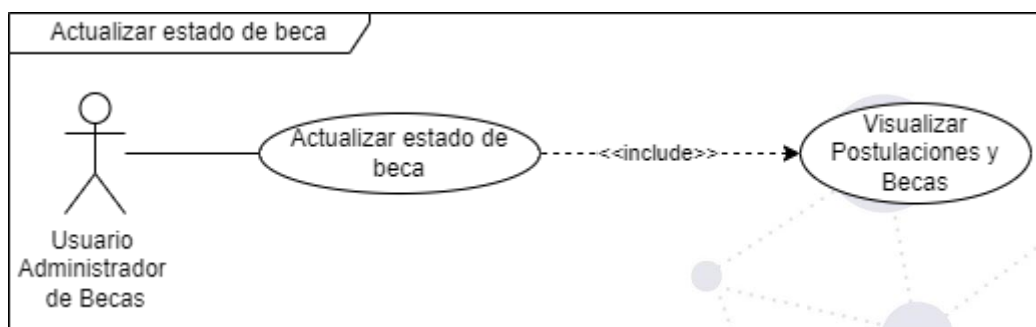
<b>Descripción</b>	El módulo informático permitirá al Usuario Administrador de Becas finalizar las Etapas de Revisión y Selección para archivarlas y evitar cambios accidentales a futuro.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Debe existir un periodo académico activo.</li> <li>• El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>• El Usuario Administrador de Becas debe estar en la que desea finalizar (Etapas de Revisión o Selección).</li> <li>• Debe existir datos de postulaciones.</li> </ul>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El Usuario Administrador de Becas presiona el botón finalizar Etapa.</li> <li>2. El sistema solicita la confirmación Usuario Administrador de Becas indicando que la información no podrá ser cambiada.</li> <li>3. El Usuario Administrador de Becas da la confirmación finalizar Etapa.</li> <li>4. El sistema redirige al Usuario Administrador de Becas a la siguiente Etapa.</li> </ol>
<b>Flujo Alternativo</b>	<p>2.1. El Usuario Administrador de Becas no da la confirmación de finalizar la Etapa.</p> <p>2.2. El sistema cancela la operación.</p> <p>3.1. Si se trata de la Etapa de Selección el sistema enviará una notificación vía correo electrónico a los postulantes, acerca de los resultados.</p>
<b>Requisito</b>	RF05

Páginas de Revisión y Selección de Postulantes - Finalizar Etapa			
<b>Nro de página</b>	03 y 04		
<b>Estado de Pruebas</b>	<b>Aceptada cambios</b>	<b>sin</b>	si
	<b>Aceptada cambios</b>	<b>con</b>	
	<b>Rediseñar</b>		
<b>Requisito Funcional</b>	RF05		

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<p><b>Observaciones</b></p>	<p>Finalizar la Etapa de Revisión hará que la información registrada en esta Etapa no pueda ser modificada a futuro.</p> <p>Finalizar la Etapa de Selección enviará la notificación a los postulantes de los resultados de postulación (si fueron o no seleccionados y los motivos) esto se corresponde con la Etapa de Notificación.</p>
	

### Requisito funcional actualizar estados de becas



<b>Identificador</b>	CU06
<b>Nombre</b>	Actualizar estados de Becas
<b>Descripción</b>	El módulo informático permitirá al Usuario Administrador de Becas la suspensión, reactivación o terminación de los becados.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>● Debe existir un periodo académico activo.</li> <li>● El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>● El Usuario Administrador de Becas debe estar en la Etapa de Adjudicación.</li> <li>● Debe existir mensajes favoritos configurados.</li> <li>● Debe existir datos de becas.</li> </ul>
<b>Flujo Principal</b>	1. El Usuario Administrador de Becas debe seleccionar una beca.



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	<ol style="list-style-type: none"> <li>2. El Usuario Administrador de Becas presiona el botón actualizar estado.</li> <li>3. El sistema solicita la confirmación del Usuario Administrador de Becas solicitando un mensaje que indique el motivo de actualización de estado del becado.</li> <li>4. El Usuario Administrador de Becas proporciona el mensaje y da la confirmación de actualizar el estado del becado.</li> <li>5. El sistema guarda la acción y notifica al estudiante.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>4.1. El Usuario Administrador de Becas no proporciona un motivo para actualizar el estado del becado.</li> <li>4.2. Si no se describen los motivos, el sistema tomará el mensaje favorito para dicha acción.             <ol style="list-style-type: none"> <li>4.2.1. Si el sistema no tiene configurado un mensaje favorito, mostrará una advertencia solicitando configurar un mensaje favorito en el apartado de configuraciones.</li> </ol> </li> <li>4.3. El sistema envía el mensaje genérico al becado seleccionado.</li> <li>5.1. El sistema no puede notificar al estudiante.</li> <li>5.2. El sistema advierte que no se envió la notificación y que no se actualizó el estado.</li> </ol>
<b>Requisito</b>	RF06


<b>Página Adjudicación de Becas - suspender, reactivar o terminar becas</b>	
<b>Nro de página</b>	05
<b>Estado de Pruebas</b>	<b>Aceptada cambios</b> <b>sin</b> si
	<b>Aceptada cambios</b> <b>con</b>
	<b>Rediseñar</b>
<b>Requisito Funcional</b>	RF06
<b>Observaciones</b>	
Suspender una beca	

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### SUSPENSIÓN DE BECA

Por favor especifique las razones por las cuales se suspende la beca al beneficiario <nombre de becario> con CI: <número de cédula>. Este mensaje se enviará al correo del becario <correo@unl.edu.ec>

Detalle claramente las razones por las que el becario será suspendido


 **SUSPENDER**

Terminar una Beca

### TERMINACIÓN DE BECA

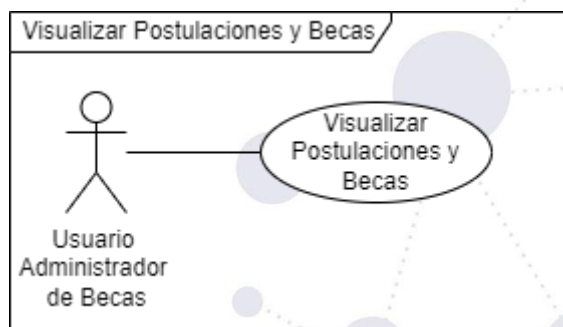
Esta acción dará por terminado los beneficios de la beca para el beneficiario <nombre beneficiario> con CI: <número de cédula>  
Se enviará un correo indicando la terminación de la beca al siguiente correo: <correo@unl.edu.ec>

(Opcional) Agregue un mensaje adicional para el beneficiario

 **TERMINAR BECA**

Nota: se utiliza la misma ventana emergente, lo que cambia es la acción.

### Requisito funcional visualizar postulaciones y becas







## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Identificador</b>	CU07
<b>Nombre</b>	Visualizar Postulaciones y Becas
<b>Descripción</b>	El módulo informático permitirá al Usuario Administrador de Becas visualizar la lista general de las postulaciones y las becas por periodo académico. Así como aplicar filtros de acuerdo a los tipos de becas, facultad, carrera, estados para cada una de las etapas.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>El Usuario Administrador de Becas debe elegir una Etapa.</li> <li>Debe existir datos de postulaciones o becas.</li> </ul>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>El Usuario Administrador de Becas selecciona los filtros necesarios para la lista de becados.</li> <li>El Usuario Administrador de Becas presiona el botón filtrar.</li> <li>El sistema devuelve los resultados de acuerdo al filtro aplicado.</li> <li>El Usuario Administrador de Becas puede visualizar la lista de becas.</li> </ol>
<b>Flujo Alternativo</b>	3.1. El filtro aplicado no devuelve datos, el sistema muestra que no existen datos.
<b>Requisito</b>	RF07

Página Adjudicación de Becas - visualización de becados		
<b>Nro de página</b>	05	
<b>Estado de Pruebas</b>	<b>Aceptada cambios</b>	sin si
	<b>Aceptada cambios</b>	con
	<b>Rediseñar</b>	
<b>Requisito Funcional</b>	RF07	
<b>Observaciones</b>		





### ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

#### Vista de Postulantes en la Etapa de Revisión

Módulo de Gestión de Becas Marlene Valdez

**ETAPA DE REVISIÓN** ACTUALIZAR ESTADO DESCARGAR DOCUMENTOS VER TRÁMITE

Facultad	Selección	Carrera	Selección	Tipo de Beca	Selección	Estado	Selección	Año	Selección	Bucar	Apellidos, Nombres y Cédula de Identidad	
Postul.	FISA	dula	DERECHO	Cori	Tipo A	Nota EVA	Sin Revisar	Oct21-Mar22	Ciclo	Tipo Beca	Estado	Nota UBU
JHULEIDY ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo B	8.7	Revisando	Oct22-Mar23	8	Tipo A	Sin Revisar	-
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo C	8.7	Aprobado	Oct23-Mar24	10	Tipo A	Revisando	-
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	8.7	Revisando	Oct24-Mar25	7	Tipo A	Aprobado	9.1
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	8.7	Revisando		6	Tipo A	Rechazado	-
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	8.7	Revisando		5	Tipo A	Sin Revisar	-
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	8.7	Revisando		3	Tipo A	Sin Revisar	-
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	8.7	Revisando		2	Tipo A	Sin Revisar	-
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	8.7	Revisando		6	Tipo A	Sin Revisar	-
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	8.7	Revisando		7	Tipo A	Sin Revisar	-

Mostrando 10 resultados de 250 << 1 de 25 >>

Estados: Sin Revisar, Revisando, Aprobado, Rechazado DESCARGAR REPORTE FINALIZAR ETAPA

#### Vista de Postulantes en la Etapa de Selección

Módulo de Gestión de Becas Marlene Valdez

**ETAPA DE SELECCIÓN** ACTUALIZAR ESTADO DESCARGAR DOCUMENTOS VER TRÁMITE

Facultad	Selección	Carrera	Selección	Tipo de Beca	Selección	Estado	Selección	Año	Selección	Bucar	Apellidos, Nombres y Cédula de Identidad
P	FISA	Cédula	DERECHO	ular	Tipo A	Facul	Sin Revisar	Oct21-Mar22	Tipo Beca	Estado	Nota EVA/UBU
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo B	FISA	Revisando	Oct22-Mar23	Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo C	FISA	Aprobado	Oct23-Mar24	Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	FISA	Rechazado	Oct24-Mar25	Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	FISA	Revisando		Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	FISA	Revisando		Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	FISA	Revisando		Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	FISA	Revisando		Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	FISA	Revisando		Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	FISA	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	Tipo A	FISA	Revisando		Tipo A	Sin Revisar	8.9

Mostrando 10 resultados de 250 << 1 de 25 >>

Estados: Sin Revisar, Revisando, Aprobado, Rechazado DESCARGAR REPORTE FINALIZAR ETAPA

#### Vista de Becados en la Etapa de Adjudicación

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

Módulo de Gestión de Becas Marlene Valdez

### ETAPA DE ADJUDICACIÓN ACTUALIZAR ESTADO

Facultad	Selección	Carrera	Selección	Tipo de Beca	Selección	Estado	Selección	Periodo	Selección	Bucar: Apellidos, Nombres y Cédula de Identidad	
<input type="checkbox"/>	Becario	Cédula	Celular	Correo	Facultad	Carrera	Ciclo	Tipo Beca	Pagos	Nota EVA/UBU	Estado
<input type="checkbox"/>	JHULEIDY DELIA ALLULIMA GUAMAN	1950025617	0969689022	jhuleidy.allulima@unl.edu.ec	FJSA	DERECHO	5	Tipo A	OCT/NOV/DIC/ENE/FEB/ABR/MAY/JUN/JUL/AGO	8.9	Activo
<input type="checkbox"/>	JHULEIDY DELIA ALLULIMA GUAMAN	1950025617	0969689022	jhuleidy.allulima@unl.edu.ec	FJSA	DERECHO	2	Tipo A	OCT/NOV/DIC/ENE/FEB	8.9	Activo
<input type="checkbox"/>	JHULEIDY DELIA ALLULIMA GUAMAN	1950025617	0969689022	jhuleidy.allulima@unl.edu.ec	FJSA	DERECHO	3	Tipo A	OCT/NOV	8.9	Activo
<input type="checkbox"/>	JHULEIDY DELIA ALLULIMA GUAMAN	1950025617	0969689022	jhuleidy.allulima@unl.edu.ec	FJSA	DERECHO	7	Tipo A	Sin Pagar	8.9	Suspendido
<input type="checkbox"/>	JHULEIDY DELIA ALLULIMA GUAMAN	1950025617	0969689022	jhuleidy.allulima@unl.edu.ec	FJSA	DERECHO	8	Tipo A	OCT/NOV/DIC/ENE/FEB/ABR/MAY/JUN/JUL/AGO	8.9	Terminado
<input type="checkbox"/>	JHULEIDY DELIA ALLULIMA GUAMAN	1950025617	0969689022	jhuleidy.allulima@unl.edu.ec	FJSA	DERECHO	5	Tipo A	OCT/NOV/DIC	8.9	Activo
<input type="checkbox"/>	JHULEIDY DELIA ALLULIMA GUAMAN	1950025617	0969689022	jhuleidy.allulima@unl.edu.ec	FJSA	DERECHO	8	Tipo A	Sin Pagar	8.9	Activo
<input type="checkbox"/>	JHULEIDY DELIA ALLULIMA GUAMAN	1950025617	0969689022	jhuleidy.allulima@unl.edu.ec	FJSA	DERECHO	4	Tipo A	Sin Pagar	8.9	Activo
<input type="checkbox"/>	JHULEIDY DELIA ALLULIMA GUAMAN	1950025617	0969689022	jhuleidy.allulima@unl.edu.ec	FJSA	DERECHO	6	Tipo A	Sin Pagar	8.9	Activo

Mostrando 10 resultados de 250 << 1 de 25 >>

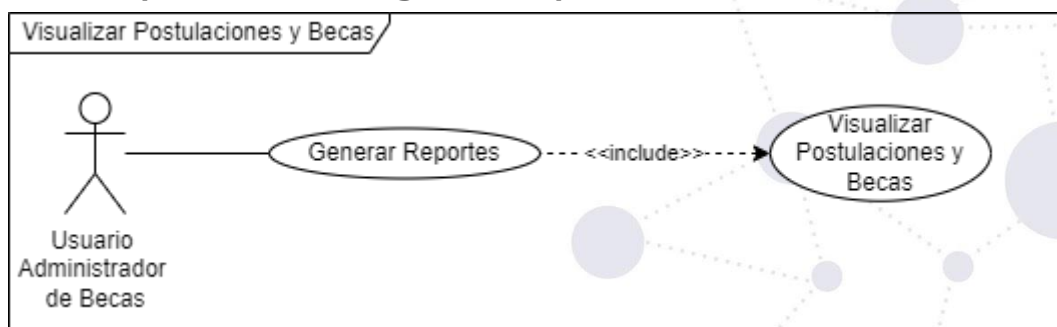
Estados: Sin Pagar, Por Pagar, Pagados, Terminados, Suspendidos, Activos

DESCARGAR REPORTE FINALIZAR ETAPA

### Filtros disponibles

Facultad:	Selección	Carrera:	Selección	Tipo de Beca:	Selección	Estado:	Selección	Año:	Selección
Postulante	FJSA	Cédula	DERECHO	Corre	Tipo A	Nota EVA	Sin Revisar	Facultad	Oct21-Mar22
JHULEIDY ALULIMA GUAMAN	FSH	0969689022	CONTABILIDAD...	jhuleidy.allulima@unl.edu.ec	Tipo B	8.7	Revisando	Estado	Oct22-Mar23
	FEAC		ODONTOLOGIA...		Tipo C		Aprobado		Oct23-Mar24
	FARNR		COMUNICACION...				Rechazado		Oct24-Mar25
	FEIRNNR								

### Requisito funcional generar reportes



<b>Identificador</b>	CU08
<b>Nombre</b>	Generar Reportes
<b>Descripción</b>	<p>El módulo informático permitirá la generación de reportes pdf y hojas de cálculo. El Administrador y el Responsable de Pagos serán capaces de generar reportes de los becados de acuerdo a las siguientes especificaciones:</p> <ul style="list-style-type: none"> <li>Reporte de Postulantes Aprobados en las Etapas de Revisión y Selección.</li> </ul>



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	<ul style="list-style-type: none"> <li>● Reporte de Postulantes Rechazados en las Etapas de Revisión y Selección.</li> <li>● Reporte de Postulantes o Becarios por periodo académico.</li> <li>● Reporte de Becarios Activos.</li> <li>● Reporte de Becados Suspendidos.</li> <li>● Reporte de Becarios con Beneficio de Beca Terminado.</li> <li>● Reporte de Becarios Pendientes de Pago.</li> <li>● Reporte de Postulantes o Becarios por Tipo de Beca.</li> <li>● Reporte de Postulantes o Becarios por Carrera.</li> <li>● Reporte de acuerdo a la combinación de filtros aplicados.</li> </ul>
<b>Precondición</b>	<ul style="list-style-type: none"> <li>● El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>● El Usuario Administrador de Becas debe estar en la Etapa correspondiente al reporte a generar.</li> <li>● Debe existir datos de postulaciones o becas, según sea el caso.</li> </ul>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El Usuario Administrador de Becas selecciona los filtros necesarios para la generación del reporte.</li> <li>2. El Usuario Administrador de Becas presiona el botón descargar reporte.</li> <li>3. El sistema descarga una hoja de cálculo de acuerdo a los filtros aplicados.</li> </ol>
<b>Flujo Alternativo</b>	<p>1.1. El filtro aplicado no devuelve datos, el sistema muestra que no existen datos.</p> <p>3.1 El sistema no descarga la información y pide volver a intentarlo más tarde.</p>
<b>Requisito</b>	RF08

### Página de Adjudicación - generar reportes

<b>Nro de página</b>	03, 04 y 05		
<b>Estado de Pruebas</b>	<b>Aceptada cambios</b>	<b>sin</b>	si
	<b>Aceptada cambios</b>	<b>con</b>	
	<b>Rediseñar</b>		

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

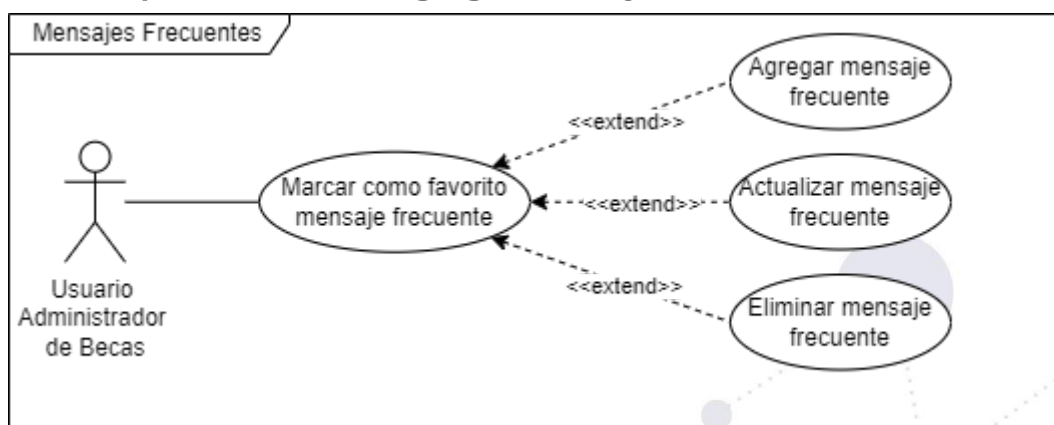
<b>Requisito Funcional</b>	RF08
<b>Observaciones</b>	Los reportes se descargan de acuerdo a los filtros aplicados

Mostrando 10 resultados de 250 << 1 de 25 >>

DESCARGAR REPORTE

FINALIZAR ETAPA

### Requisito funcional agregar mensaje frecuente



<b>Identificador</b>	CU09
<b>Nombre</b>	Agregar mensaje frecuente
<b>Descripción</b>	El módulo informático permitirá agregar mensajes frecuentes para la aprobación, rechazo, suspensión, reactivación y terminación de una beca. El Administrador será capaz de agregar tantos motivos como considere necesario, que se utilizaran para notificar el estado de una beca.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>El Usuario Administrador de Becas debe estar en la página de Configuraciones.</li> </ul>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>El Usuario Administrador de Becas se dirige al apartado del mensaje frecuente que desea agregar de acuerdo a la etapa correspondiente.</li> <li>El Usuario Administrador de Becas escribe un mensaje frecuente y presiona agregar.</li> </ol>



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	6. El sistema guarda el mensaje y refresca la página.
<b>Flujo Alternativo</b>	6.1. Si el mensaje no pudo ser guardado, el sistema muestra una advertencia de que el mensaje no fue guardado.
<b>Requisito</b>	RF09

Página de Configuración - Mensajes Frecuentes			
<b>Nro de página</b>	06		
<b>Estado de Pruebas</b>	<b>Aceptada cambios</b>	<b>sin</b>	si
	<b>Aceptada cambios</b>	<b>con</b>	
	<b>Rediseñar</b>		
<b>Requisito Funcional</b>	RF09		
<b>Observaciones</b>	Agregar mensajes frecuentes para rechazar o aceptar becas y mensajes frecuentes para suspender o terminar becas		

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### Configuraciones

#### Configuraciones para las Etapas de Revisión y Selección

#### Rechazar Postulación

Mensajes Personalizados para Rechazar Postulaciones

Escriba un nuevo mensaje personalizado

lamentamos informarle que la solicitud de su beca fue rechazada

no cumple con los requisitos establecidos en el reglamento

falta la firma del secretario abogado

★ Favorito
➕ Agregar
🔄 Actualizar
🗑 Eliminar

#### Aceptar Postulación

Mensajes Personalizados para Aceptar Postulaciones

Escriba un nuevo mensaje personalizado

nos complace informarle que la solicitud de su beca fue aceptada

queremos darle la bienvenida al programa de becas de la Universidad Nacional de L...

ud ha sido aceptado para recibir el programa de ayuda socio-economica

★ Favorito
➕ Agregar
🔄 Actualizar
🗑 Eliminar

#### Configuraciones para la Etapa de Adjudicación

#### Suspensión de Beca

Mensajes Personalizados para Suspender Becas

Escriba un nuevo mensaje personalizado

estimado becario nos hemos visto en la obligación de suspender su beca debido a...

su beca ha sido suspendida debido a que no present los documentos requeridos en ...

lamentamos informarle que su beca tuvo que ser suspendida por desacato a los ...

★ Favorito
➕ Agregar
🔄 Actualizar
🗑 Eliminar

#### Terminación de Beca

Mensajes Personalizados para Terminar Becas

Escriba un nuevo mensaje personalizado

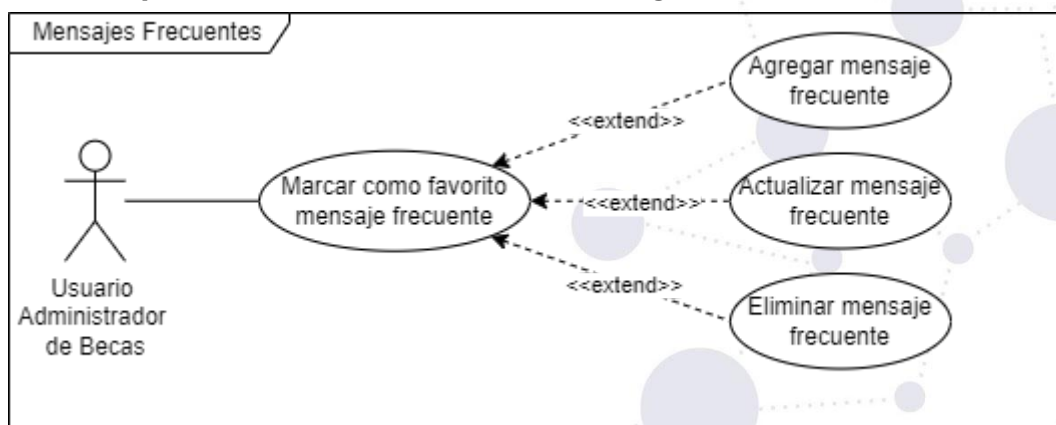
Estimado becario su beca se da por terminado debido a la culminación de sus estud...

La Unidad de Bienestar Universitario le desea éxitos en su etapa profesional y le info...

Le informamos que a partir de ahora ya no recibirá los beneficios de su beca

★ Favorito
➕ Agregar
🔄 Actualizar
🗑 Eliminar

### Requisito funcional actualizar mensaje frecuente



<b>Identificador</b>	CU10
<b>Nombre</b>	Actualizar mensaje frecuente





## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Descripción</b>	El módulo informático permitirá actualizar mensajes frecuentes para la aprobación, rechazo, suspensión, reactivación y terminación de una beca. El Administrador será capaz de actualizar tantos motivos como considere necesario que se utilizaran para notificar el estado de una beca.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>El Usuario Administrador de Becas debe estar en la página de Configuraciones.</li> </ul>
<b>Flujo Principal</b>	<p>7. El Usuario Administrador de Becas se dirige al apartado del mensaje frecuente que desea gestionar de acuerdo a la etapa correspondiente.</p> <p>8. El Usuario Administrador de Becas selecciona el mensaje que va a modificar y presiona el botón actualizar.</p> <p>9. El Usuario Administrador de Becas edita el mensaje y presiona nuevamente en el botón actualizar</p> <p>10. El sistema actualiza el mensaje y refresca la página.</p>
<b>Flujo Alternativo</b>	<p>6.1. El Usuario Administrador de Becas puede eliminar un mensaje agregado, seleccionando el mensaje y presionando en el botón eliminar.</p> <p>6.2. El sistema pide confirmación para eliminar el mensaje.</p> <p>6.3. El sistema elimina el mensaje.</p> <p>6.1. El Usuario Administrador de Becas selecciona un mensaje y presiona el botón favorito.</p> <p>6.3. El sistema actualiza el estado del mensaje.</p> <p>6.1. El mensaje no pudo ser guardado, actualizado, eliminado o marcado como favorito.</p> <p>6.2. El sistema muestra una advertencia de que el mensaje no fue guardado.</p>
<b>Requisito</b>	RF10

Página de Configuración - Mensajes Frecuentes			
<b>Nro de página</b>	06		
<b>Estado de Pruebas</b>	<b>Aceptada cambios</b>	<b>sin</b>	si



### ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	<b>Aceptada cambios</b>	<b>con</b>	
	<b>Rediseñar</b>		
<b>Requisito Funcional</b>	RF10		
<b>Observaciones</b>			

Actualizar mensajes frecuentes para rechazar o aceptar becas y mensajes frecuentes para suspender o terminar becas

#### Configuraciones

##### Configuraciones para las Etapas de Revisión y Selección

#### Rechazar Postulación

Mensajes Personalizados para Rechazar Postulaciones

Escriba un nuevo mensaje personalizado

lamentamos informarle que la solicitud de su beca fue rechazada

no cumple con los requisitos establecidos en el reglamento

falta la firma del secretario abogado

☆ Favorito
➕ Agregar
🔄 Actualizar
🗑 Eliminar

#### Aceptar Postulación

Mensajes Personalizados para Aceptar Postulaciones

Escriba un nuevo mensaje personalizado

nos complace informarle que la solicitud de su beca fue aceptada

querémos darle la bienvenida al programa de becas de la Universidad Nacional de L...

ud ha sido aceptado para recibir el programa de ayuda socio-economica

☆ Favorito
➕ Agregar
🔄 Actualizar
🗑 Eliminar

##### Configuraciones para la Etapa de Adjudicación

#### Suspensión de Beca

Mensajes Personalizados para Suspender Becas

Escriba un nuevo mensaje personalizado

estimado becario nos hemos visto en la obligación de suspender su beca debido a...

su beca ha sido suspendida debido a que no present los documentos requeridos en ...

lamentamos informarle que su beca tuvo que ser suspendida por desacato a los ...

☆ Favorito
➕ Agregar
🔄 Actualizar
🗑 Eliminar

#### Terminación de Beca

Mensajes Personalizados para Terminar Becas

Escriba un nuevo mensaje personalizado

Estimado becario su beca se da por terminado debido a la culminación de sus estud...

La Unidad de Bienestar Universitario le desea éxitos en su etapa profesional y le info...

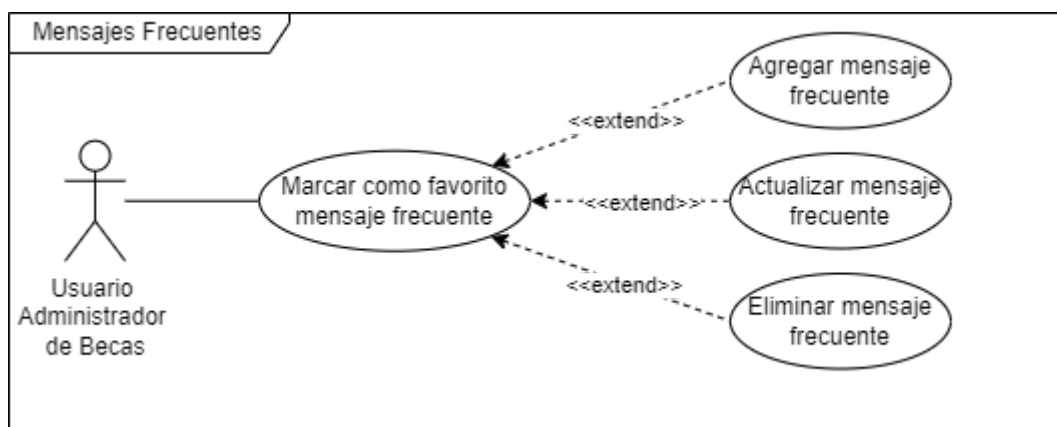
Le informamos que a partir de ahora ya no recibirá los beneficios de su beca

☆ Favorito
➕ Agregar
🔄 Actualizar
🗑 Eliminar



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### Requisito funcional eliminar mensaje frecuente



<b>Identificador</b>	CU11
<b>Nombre</b>	Eliminar mensaje frecuente
<b>Descripción</b>	El módulo informático permitirá eliminar mensajes frecuentes para la aprobación, rechazo, suspensión, reactivación y terminación de una beca. El Administrador será capaz de eliminar tantos motivos como considere necesario que se utilizaran para notificar el estado de una beca.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>El Usuario Administrador de Becas debe estar en la página de Configuraciones.</li> </ul>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>El Usuario Administrador de Becas se dirige al apartado del mensaje frecuente que desea gestionar de acuerdo a la etapa correspondiente.</li> <li>El Usuario Administrador de Becas selecciona el mensaje que desea eliminar.</li> <li>El Usuario Administrador de Becas presiona el botón eliminar.</li> <li>El sistema pide confirmación para eliminar el mensaje.</li> <li>El Usuario Administrador de Becas presiona el botón confirmar.</li> <li>El sistema elimina el mensaje y refresca la página.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>Si existe un error o se desea cambiar algún mensaje agregado, el Usuario Administrador de Becas selecciona el mensaje y presiona el botón actualizar.</li> <li>El Usuario Administrador de Becas edita el mensaje y presiona nuevamente en el botón actualizar.</li> <li>El sistema actualiza el mensaje y la página.</li> </ol>



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	<p>6.1. El Usuario Administrador de Becas puede eliminar un mensaje agregado, seleccionando el mensaje y presionando en el botón eliminar.</p> <p>6.2. El sistema pide confirmación para eliminar el mensaje.</p> <p>6.3. El sistema elimina el mensaje.</p> <p>6.1. El Usuario Administrador de Becas selecciona un mensaje y presiona el botón favorito.</p> <p>6.3. El sistema actualiza el estado del mensaje.</p> <p>6.1. El mensaje no pudo ser guardado, actualizado, eliminado o marcado como favorito.</p> <p>6.2. El sistema muestra una advertencia de que el mensaje no fue eliminado.</p>
<b>Requisito</b>	RF11

Página de Configuración - Mensajes Frecuentes	
<b>Nro de página</b>	06
<b>Estado de Pruebas</b>	<b>Aceptada cambios</b> <b>sin</b> si
	<b>Aceptada cambios</b> <b>con</b>
	<b>Rediseñar</b>
<b>Requisito Funcional</b>	RF11
<b>Observaciones</b>	Eliminar mensajes frecuentes para rechazar o aceptar becas y mensajes frecuentes para suspender o terminar becas

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### Configuraciones

#### Configuraciones para las Etapas de Revisión y Selección

##### Rechazar Postulación

Mensajes Personalizados para Rechazar Postulaciones

Escriba un nuevo mensaje personalizado

lamentamos informarle que la solicitud de su beca fue rechazada

no cumple con los requisitos establecidos en el reglamento

falta la firma del secretario abogado

☆ Favorito
➕ Agregar
🔄 Actualizar
🗑️ Eliminar

##### Aceptar Postulación

Mensajes Personalizados para Aceptar Postulaciones

Escriba un nuevo mensaje personalizado

nos complace informarle que la solicitud de su beca fue aceptada

queremos darle la bienvenida al programa de becas de la Universidad Nacional de L...

ud ha sido aceptado para recibir el programa de ayuda socio-economica

☆ Favorito
➕ Agregar
🔄 Actualizar
🗑️ Eliminar

#### Configuraciones para la Etapa de Adjudicación

##### Suspensión de Beca

Mensajes Personalizados para Suspender Becas

Escriba un nuevo mensaje personalizado

estimado becario nos hemos visto en la obligación de suspender su beca debido a...

su beca ha sido suspendida debido a que no present los documentos requeridos en ...

lamentamos informarle que su beca tuvo que ser suspendida por desacato a los ...

☆ Favorito
➕ Agregar
🔄 Actualizar
🗑️ Eliminar

##### Terminación de Beca

Mensajes Personalizados para Terminar Becas

Escriba un nuevo mensaje personalizado

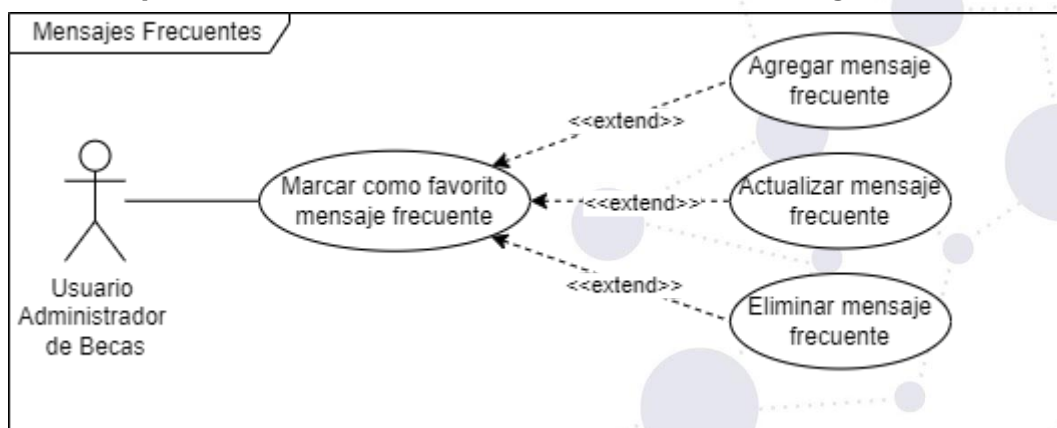
Estimado becario su beca se da por terminado debido a la culminación de sus estud...

La Unidad de Bienestar Universitario le desea éxitos en su etapa profesional y le info...

Le informamos que a partir de ahora ya no recibirá los beneficios de su beca

☆ Favorito
➕ Agregar
🔄 Actualizar
🗑️ Eliminar

### Requisito funcional marcar como favorito mensaje frecuente



Identificador

CU12

Nombre

Marcar como favorito mensaje frecuente



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Descripción</b>	El módulo informático permitirá marcar como favoritos mensajes frecuentes para la aprobación, rechazo, suspensión, reactivación y terminación de una beca. El Administrador será capaz de marcar como favorito un mensaje para utilizarlo al notificar el estado de una beca.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>El Usuario Administrador de Becas debe estar en la página de Configuraciones.</li> </ul>
<b>Flujo Principal</b>	<p>17. El Usuario Administrador de Becas se dirige al apartado del mensaje frecuente que desea gestionar de acuerdo a la etapa correspondiente.</p> <p>18. El Usuario Administrador de Becas selecciona el mensaje que desea marcar como favorito.</p> <p>19. El Usuario Administrador de Becas presiona el botón favorito.</p> <p>20. El sistema guarda el mensaje seleccionado como favorito y refresca la página.</p>
<b>Flujo Alternativo</b>	6.1. Si el mensaje no pudo ser marcado como favorito, el sistema muestra una advertencia de que el mensaje no fue marcado como favorito.
<b>Requisito</b>	RF12

Página de Configuración - Mensajes Frecuentes			
<b>Nro de página</b>	06		
<b>Estado de Pruebas</b>	<b>Aceptada cambios</b>	<b>sin</b>	si
	<b>Aceptada cambios</b>	<b>con</b>	
	<b>Rediseñar</b>		
<b>Requisito Funcional</b>	RF12		
<b>Observaciones</b>	Marcar mensajes frecuentes como favorito para rechazar o aceptar becas y mensajes frecuentes para suspender o terminar becas		

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### Configuraciones

#### Configuraciones para las Etapas de Revisión y Selección

#### Rechazar Postulación

Mensajes Personalizados para Rechazar Postulaciones

Escriba un nuevo mensaje personalizado

lamentamos informarle que la solicitud de su beca fue rechazada

no cumple con los requisitos establecidos en el reglamento

falta la firma del secretario abogado

★ Favorito
➕ Agregar
🔄 Actualizar
🗑️ Eliminar

#### Aceptar Postulación

Mensajes Personalizados para Aceptar Postulaciones

Escriba un nuevo mensaje personalizado

nos complace informarle que la solicitud de su beca fue aceptada

queremos darle la bienvenida al programa de becas de la Universidad Nacional de L...

ud ha sido aceptado para recibir el programa de ayuda socio-economica

★ Favorito
➕ Agregar
🔄 Actualizar
🗑️ Eliminar

#### Configuraciones para la Etapa de Adjudicación

#### Suspensión de Beca

Mensajes Personalizados para Suspender Becas

Escriba un nuevo mensaje personalizado

estimado becario nos hemos visto en la obligación de suspender su beca debido a...

su beca ha sido suspendida debido a que no present los documentos requeridos en ...

lamentamos informarle que su beca tuvo que ser suspendida por desacato a los ...

★ Favorito
➕ Agregar
🔄 Actualizar
🗑️ Eliminar

#### Terminación de Beca

Mensajes Personalizados para Terminar Becas

Escriba un nuevo mensaje personalizado

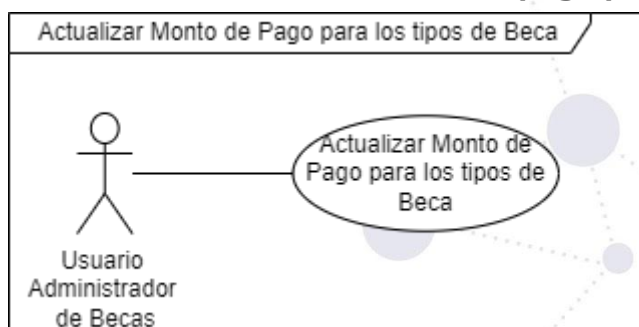
Estimado becario su beca se da por terminado debido a la culminación de sus estud...

La Unidad de Bienestar Universitario le desea éxitos en su etapa profesional y le info...

Le informamos que a partir de ahora ya no recibirá los beneficios de su beca

★ Favorito
➕ Agregar
🔄 Actualizar
🗑️ Eliminar

### Requisito funcional actualizar montos de pago para los tipos de beca



<b>Identificador</b>	CUI3
<b>Nombre</b>	Actualizar montos de pago para los tipos de beca
<b>Descripción</b>	El módulo informático permitirá al Usuario Administrador de Becas modificar el monto de pago de acuerdo al tipo de beca de manera parametrizable, para facilitar el registro de pago de becas.



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Precondición</b>	<ul style="list-style-type: none"> <li>El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>El Usuario Administrador de Becas debe estar en la página de Configuraciones.</li> </ul>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>El Usuario Administrador de Becas actualiza el monto de pago para un cierto tipo de beca.</li> <li>El Usuario Administrador de Becas presiona en el botón actualizar.</li> <li>El sistema solicita confirmación.</li> <li>El Usuario Administrador de Becas da la confirmación de registro.</li> <li>El sistema establece el nuevo registro.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>El monto de pago no pudo ser guardado.</li> <li>El Usuario Administrador de Becas debe volver a intentarlo.</li> </ol>
<b>Requisito</b>	RF13

Página de Configuración - montos de pago para los tipos de beca			
<b>Nro de página</b>	06		
<b>Estado de Pruebas</b>	<b>Aceptada cambios</b>	<b>sin</b>	si
	<b>Aceptada cambios</b>	<b>con</b>	
	<b>Rediseñar</b>		
<b>Requisito Funcional</b>	RF13		
<b>Observaciones</b>	Se podrá cambiar los valores a registrar de pago en caso de que en un futuro estos valores cambien, caso contrario esto se requiere una única vez. El cambio de monto de pago no afectará a registros anteriores, afectará a los nuevos registros de pago.		



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

**Configuraciones de Becas**

**Registro de Pagos**  
 Valores para el pago de becas de acuerdo al tipo de beca

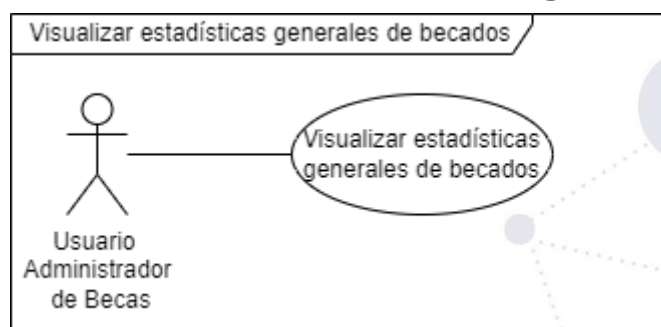
Valor de Pago para el Tipo de Beca A

Valor de Pago para el Tipo de Beca B

Valor de Pago para el Tipo de Beca C

Actualizar

### Requisito funcional visualizar estadísticas generales de becados

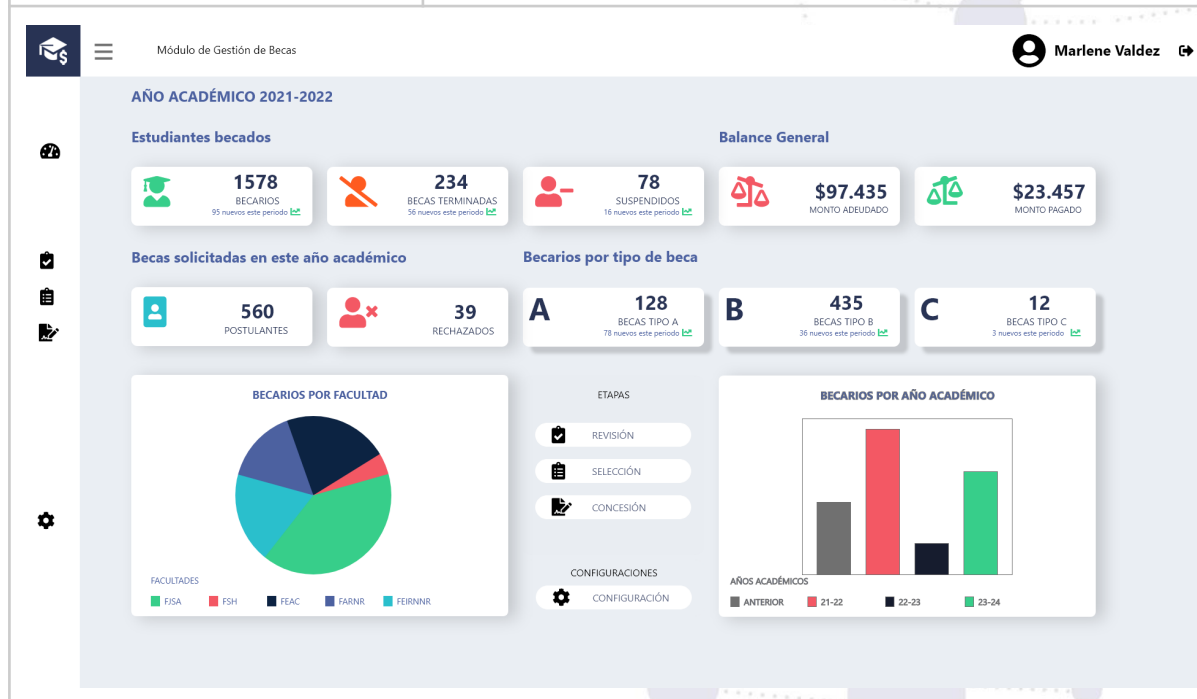


<b>Identificador</b>	CU14
<b>Nombre</b>	Mostrar estadísticas generales de becados
<b>Descripción</b>	El módulo informático mostrará estadísticas generales sobre las becas otorgadas y las postulaciones de cada periodo académico. El Administrador será capaz de visualizar estadísticas generales sobre las becas y postulaciones.
<b>Precondición</b>	<ul style="list-style-type: none"> <li>● Debe existir un periodo académico activo.</li> <li>● El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>● El Usuario Administrador de Becas debe estar en la página dashboard.</li> <li>● Debe existir datos de postulaciones y becas.</li> </ul>

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Flujo Principal</b>	1. El usuario visualiza los datos y gráficas de los postulantes y las becas. por facultad y carrera. por periodo académico.
<b>Flujo Alternativo</b>	1. El sistema no muestra estadísticas debido a que no se encontraron datos para el año activo.
<b>Requisito</b>	RF14

Página Dashboard		
<b>Nro de página</b>	02	
<b>Estado de Pruebas</b>	<b>Aceptada sin cambios</b>	
	<b>Aceptada con cambios</b>	Mostrar becas por periodo académico en vez de becas por régimen
	<b>Rediseñar</b>	
<b>Requisito Funcional</b>	RF14	
<b>Observaciones</b>	Utilizar la gráfica de barras propuesta para mostrar estadísticas de becas por periodo académico en vez de becarios por régimen	



**Módulo de Gestión de Becas** | Marlene Valdez

**AÑO ACADÉMICO 2021-2022**

**Estudiantes becados**

- 1578 BECARIOS (95 nuevos este periodo)
- 234 BECAS TERMINADAS (56 nuevos este periodo)
- 78 SUSPENDIDOS (16 nuevos este periodo)

**Balance General**

- \$97.435 MONTO ADEUDADO
- \$23.457 MONTO PAGADO

**Becas solicitadas en este año académico**

- 560 POSTULANTES
- 39 RECHAZADOS

**Becarios por tipo de beca**

- A: 128 BECAS TIPO A (78 nuevos este periodo)
- B: 435 BECAS TIPO B (36 nuevos este periodo)
- C: 12 BECAS TIPO C (3 nuevos este periodo)

**BECARIOS POR FACULTAD**

FACULTADES: FISA, FSH, FEAC, FARNR, FERNNR

**ETAPAS**

- REVISIÓN
- SELECCIÓN
- CONCESIÓN

**CONFIGURACIONES**

- CONFIGURACIÓN

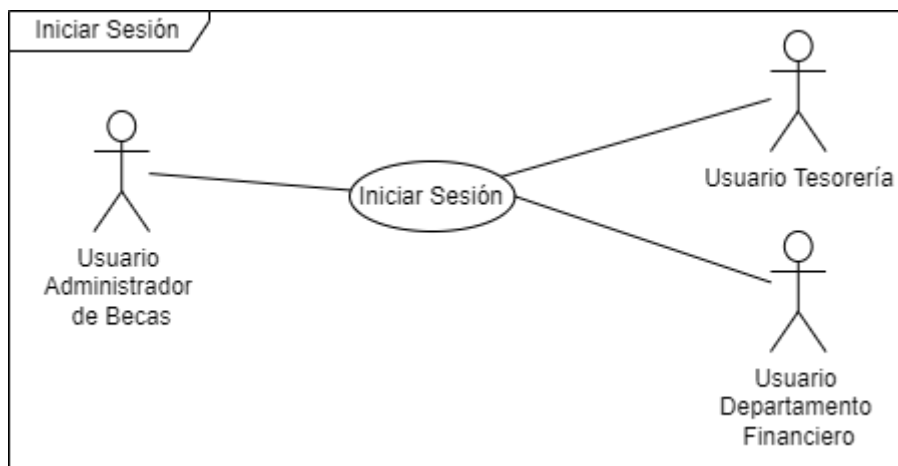
**BECARIOS POR AÑO ACADÉMICO**

AÑOS ACADÉMICOS: ANTERIOR, 21-22, 22-23, 23-24



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### Requisito funcional iniciar sesión




<b>Identificador</b>	CU15
<b>Nombre</b>	Iniciar Sesión
<b>Descripción</b>	El módulo informático permitirá acceder a sus funcionalidades mediante el inicio de sesión del SIAAF. El Administrador será capaz de iniciar sesión en el MGB mediante el inicio de sesión provisto por el SIAAF.
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. Acceder al SIAAF.</li> <li>2. Encontrarse en la página de inicio de sesión del SIAAF.</li> <li>3. Tener los privilegios de acceso asignados.</li> </ol>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>2. El Usuario Administrador de Becas proporciona sus credenciales de acceso.</li> <li>3. El Usuario Administrador de Becas presiona en acceder.</li> <li>4. El sistema redirige a la vista principal.</li> </ol>
<b>Flujo Alternativo</b>	El sistema no encontró la cuenta para los datos ingresados. El usuario no ingresa las credenciales... El sistema muestra que hubo un error en los datos ingresados.
<b>Requisito</b>	RF15

### Página Inicio de Sesión


<b>Nro de página</b>	01
----------------------	----


## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

Estado de Pruebas	Aceptada cambios	sin	si
	Aceptada cambios	con	
	Rediseñar		
Requisito Funcional	RF15		
Observaciones			



### MODULO DE GESTIÓN DE BECAS



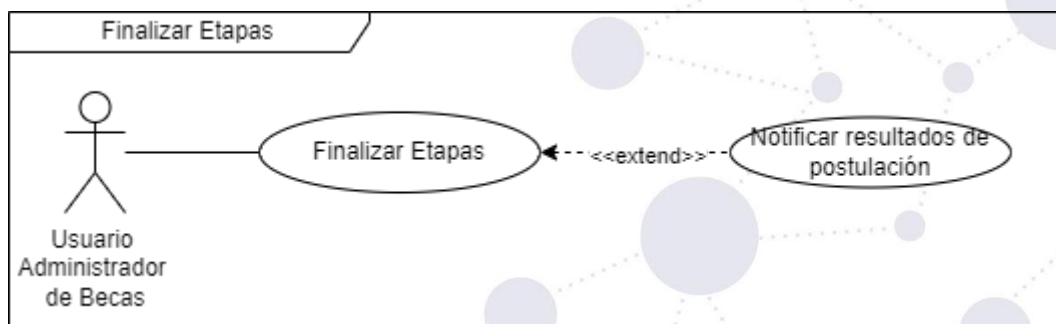


Universidad  
Nacional  
de Loja

INICIAR SESIÓN

Ingresar

### Requisito funcional notificar resultados de postulación

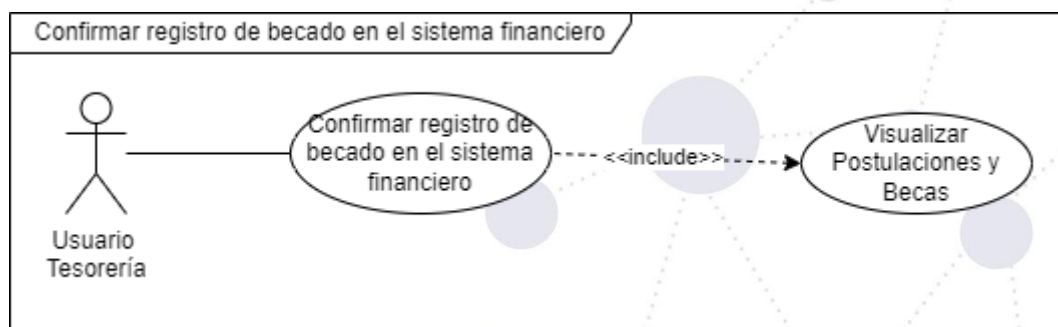


<b>Identificador</b>	CU16
<b>Nombre</b>	Notificar resultados de Postulación

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Descripción</b>	El módulo informático permitirá notificar a los postulantes los resultados de su Postulación y a tesorería los nuevos becados para su registro en el sistema financiero.
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. Debe existir un periodo académico activo.</li> <li>2. El Usuario Administrador de Becas debe iniciar sesión en el SIAAF.</li> <li>3. El Usuario Administrador de Becas debe encontrarse en la etapa de Selección.</li> <li>4. Debe existir datos de postulantes y becados.</li> </ol>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El Usuario Administrador de Becas presiona en el botón Finalizar Etapa.</li> <li>2. El sistema guarda las notificaciones para los postulantes y para la tesorería.</li> <li>3. El sistema envía la notificación de los postulantes con el resultado de su Postulación mediante correo electrónico.</li> <li>4. El sistema envía una notificación a la cuenta de tesorería en el SIAAF y mediante correo electrónico, para informar del registro de nuevos becados.</li> <li>5. El sistema confirma el envío de notificaciones.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2.1. El sistema no puede guardar las notificaciones.</li> <li>2.2. El sistema informa que no se pudo guardar las notificaciones.</li> <li>3.1. El sistema no puede enviar la notificación mediante correo electrónico.</li> <li>3.2. El sistema vuelve a intentarlo más tarde.</li> </ol>
<b>Requisito</b>	RF16

### Requisito funcional guardar estado de registro del nuevo becado en el sistema financiero



<b>Identificador</b>	CU17
----------------------	------

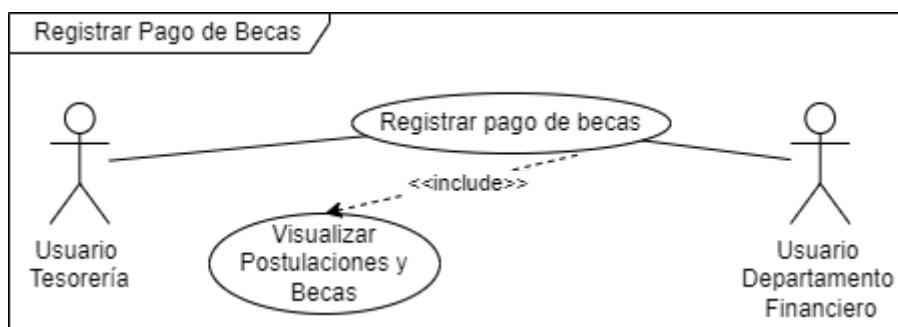


## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Nombre</b>	Guardar estado de registro del nuevo becado en el sistema financiero
<b>Descripción</b>	El módulo informático permitirá guardar un estado de registro para los nuevos becados de cada periodo lectivo. El usuario de tesorería será capaz de actualizar el estado de los nuevos becados para indicar su correcto registro en el departamento financiero.
<b>Precondición</b>	<ol style="list-style-type: none"><li>1. Debe existir un periodo lectivo activo.</li><li>2. El usuario de tesorería debe iniciar sesión en el SIAAF.</li><li>3. El usuario de tesorería debe estar en la página de listado de nuevos becados del periodo lectivo en curso.</li><li>4. Debe existir datos de nuevas becas.</li></ol>
<b>Flujo Principal</b>	<ol style="list-style-type: none"><li>1. El usuario de tesorería selecciona uno o varios becados.</li><li>2. El usuario de tesorería presiona el botón marcar registro.</li><li>3. El sistema muestra una ventana emergente indicando el número de becados a los cuales se va a cambiar el estado de registro en el sistema financiero a el estado de registrado.</li><li>4. El usuario de tesorería confirma la actualización del estado de registro presionando el botón confirmar.</li><li>5. El sistema actualiza el estado de registro en el departamento financiero.</li><li>6. El sistema confirma el registro exitoso y recarga la página para actualizar el registro de pago.</li></ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"><li>3.1. El sistema no muestra el número de becados seleccionados.</li><li>3.2. El sistema inhabilita el botón confirmar por ende no permite actualizar el estado de registro.</li><li>5.1. El sistema no pudo cambiar el estado de registro para los becados seleccionados.</li><li>5.2. El sistema muestra una advertencia y pide volver a intentarlo más tarde.</li></ol>
<b>Requisito</b>	RF17

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### Requisito funcional registrar pago de becas



<b>Identificador</b>	CU18
<b>Nombre</b>	Registrar Pago de Becas
<b>Descripción</b>	El módulo informático permitirá registrar los pagos realizados a los becados. El usuario del Departamento Financiero y el de Tesorería serán capaces de importar los registros de devengado y pago respectivamente con un formato preestablecido.
<b>Precondición</b>	<ol style="list-style-type: none"> <li>5. Debe existir un periodo académico activo.</li> <li>6. El Usuario del Departamento Financiero y el de Tesorería debe iniciar sesión en el SIAAF.</li> <li>7. Deben estar en la sección de registros de pagos.</li> <li>8. Debe existir datos de becas.</li> <li>9. Se requiere archivo de importación del departamento financiero.</li> </ol>
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El Usuario del Departamento Financiero o el de Tesorería presiona el botón importar registros de devengado o pago según sea el caso.</li> <li>2. El sistema redirige a una vista de importación de archivos.</li> <li>3. El Usuario del Departamento Financiero o el de Tesorería presionan el botón Seleccionar archivo.</li> <li>4. El Usuario del Departamento Financiero o el de Tesorería seleccionan el archivo.</li> <li>5. El Usuario del Departamento Financiero o el de Tesorería presionan el botón Cargar archivo.</li> <li>6. El sistema realiza una serie de comprobaciones que son aprobadas.</li> <li>7. El sistema registra el devengado o pago según sea el caso, de cada uno de los becados de acuerdo al archivo cargado.</li> <li>8. El sistema confirma el registro exitoso.</li> </ol>



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Flujo Alterno</b>	<p>1.1 El Usuario de Tesorería presiona el botón actualizar registro de pago. 1.2 El sistema solicita confirmación. 1.3 El sistema actualiza el registro de pago.</p> <p>6.1. El sistema muestra una advertencia indicando que uno o varios de los registros de pago presentan inconsistencias. 6.2. El sistema cancela la operación. 6.3. El sistema devuelve la lista de inconsistencias encontradas.</p> <p>7.1. El sistema no puede guardar el registro de pago. 7.2. El sistema muestra una advertencia y pide volver a intentarlo más tarde.</p>
<b>Requisito</b>	RF18

### 3.3 Requerimientos no funcionales

#### 3.1 Requisitos de rendimiento

- Las transacciones de lectura de información deben realizarse en menos de 2s.
- Las transacciones de lectura deben cargarse bajo demanda organizados en lotes de 10, 25 y 50 registros al mismo tiempo.
- En el 90% de los casos las transacciones de guardado y modificación deben tardar 2s o menos.
- El sistema será capaz de permitir la conexión simultánea de 2 o más usuarios al mismo tiempo.

#### 3.2 Seguridad

- El módulo informático permitirá únicamente el acceso a los administradores del SIAAF, al Responsable de la Gestión de Becas y al Responsable del Registro de Pagos.
- El módulo informático hará la distinción entre los roles y privilegios que tiene cada usuario.
- La sesión del usuario utilizará el servicio de autenticación implementado para el SIAAF.
- La seguridad de la aplicación está cubierta por la arquitectura provista por el SIAAF.





## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

---

### 3.3 Fiabilidad

- La base de datos perteneciente al MGB será respaldada automáticamente y subida a el servidor de respaldos, de acuerdo a lo establecido por el SIAAF.

### 3.4 Disponibilidad

- El módulo informático estará disponible en horarios laborables, es decir, los 5 días de la semana, 8 horas, exceptuando los casos de algún mantenimiento programado mayor el cual será notificado con anticipación por parte de DTI.

### 3.5 Mantenibilidad

- El módulo informático seguirá los patrones de diseño y sintaxis que están establecidos por el SIAAF hasta la fecha para facilitar su continuo desarrollo.
- Una vez realizada las pruebas de calidad y capacitado al personal involucrado y el Módulo esté trabajando bajo los requerimientos solicitados, el MGB queda a cargo de la DTI para la implementación de nuevas funcionalidades y el mantenimiento que se requiera.

### 3.6 Portabilidad

- El módulo informático es parte del SIAAF por lo que puede comunicarse con módulos pertenecientes a este sistema.
- El módulo informático está pensado para interactuar activamente con el MGT, por lo que se hará uso de apis para permitir la comunicación entre estos.
- El módulo informático será construido con el framework Django para mantener la misma tecnología que se usó en la construcción de otros módulos informáticos desarrollados por DTI.
- El Módulo puede accederse desde cualquier dispositivo con conexión a internet para su uso.

## 4. Entorno Operativo

### 4.1 Funcionamiento de auditoría

Se almacenarán los registros de actividad o historial de logs de las distintas acciones que ejecuten los usuarios, respecto a la ejecución de una actualización de estado en la beca.



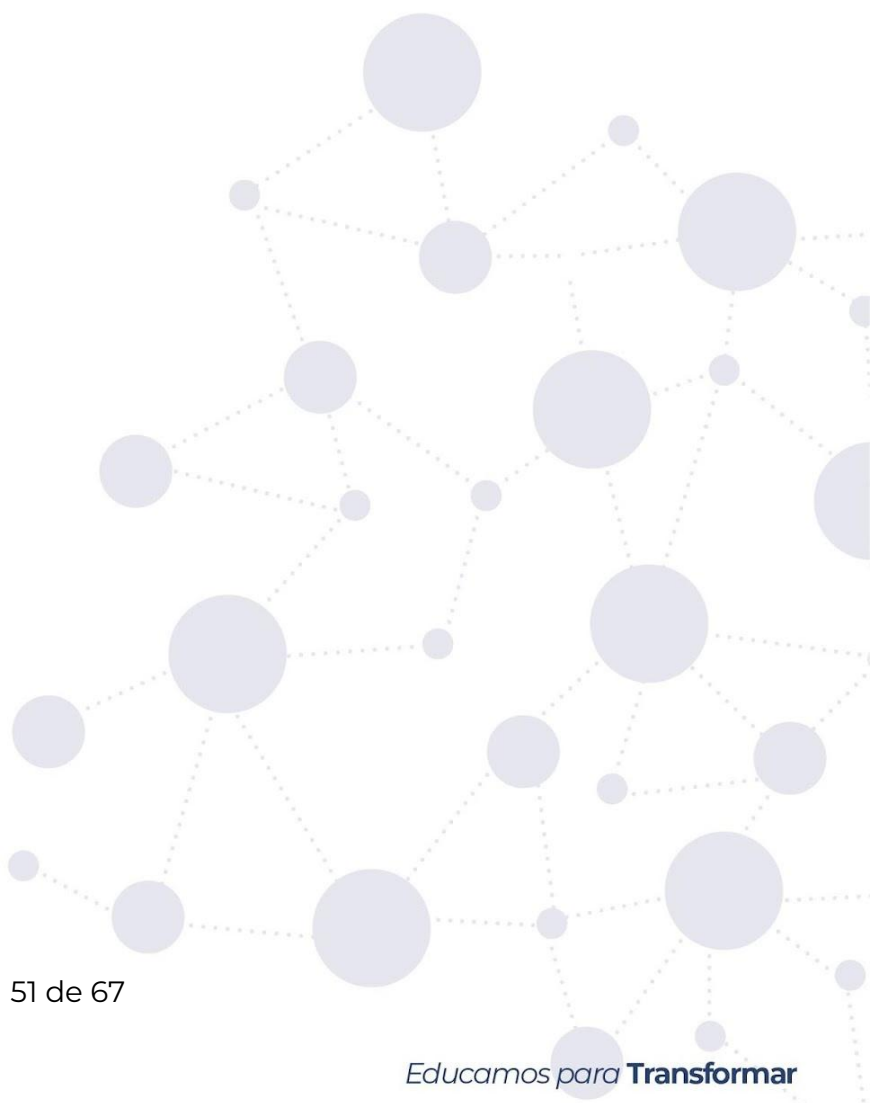
## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

---

### 4.2 De control de acceso

El sistema debe controlar los permisos que tiene cada usuario para su accesibilidad de una manera correcta, de tal forma que pueda acceder la información que le corresponde de acuerdo a su rol.

### Historias de usuario







UNL

Universidad Nacional de Loja

Dirección de Tecnologías de Información

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

ID: H001 Sincronizar datos de postulación desde el MGT hacia el MGB			
Usuario / Rol	Usuario Administrador de Becas	Prioridad	Alta
		Puntos estimados	10
Descripción	Como Usuario Administrador de Becas, quiero sincronizar la información de los postulantes que se encuentra en el MGT hacia el MGB para poder gestionar postulaciones y becas.		
Nro	Escenario	Criterio de aceptación	
1	<ul style="list-style-type: none"> <li>Poder sincronizar las postulaciones desde el MGT hacia el MGB.</li> <li>Poder visualizar las postulaciones del periodo sincronizado en la Etapa de Revisión..</li> </ul>	<ul style="list-style-type: none"> <li>Que exista un botón para sincronizar datos en el MGT.</li> <li>Que en el MGB se pueda visualizar a los postulantes en la Etapa de Revisión.</li> <li>Que la información de los postulantes permanezca guardada en el MGB luego de la sincronización.</li> </ul>	

ID: H002 Actualizar estados de postulaciones			
Usuario / Rol	Usuario Administrador de Becas	Prioridad	
		Puntos estimados	
Descripción	Como Usuario Administrador de Becas, quiero guardar el registro de postulantes que aprueban las etapas de Revisión y Selección, para a futuro poder consultar el estado de los postulantes y becarios por cada una de las etapas.		
Nro	Escenario	Criterio de aceptación	
1	<ul style="list-style-type: none"> <li>Poder aprobar y rechazar</li> </ul>	<ul style="list-style-type: none"> <li>Que exista un botón para actualizar el estado de postulantes y</li> </ul>	



UNL

Universidad Nacional de Loja

Dirección de Tecnologías de Información

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	postulantes en las Etapas de Revisión y Selección. <ul style="list-style-type: none"> <li>• Poder suspender, terminar y reactivar Becados.</li> </ul>	becarios. <ul style="list-style-type: none"> <li>• Que exista un modal que me pida confirmación para actualizar estado.</li> <li>• Que en el modal exista un campo que permita ingresar los motivos de aprobación, rechazo de postulantes y suspensión, terminación de becados.</li> </ul>
--	---	--

ID: H003 Descargar documentos de postulación		
<b>Usuario / Rol</b>	Usuario Administrador de Becas	<b>Prioridad</b>
		<b>Puntos estimados</b>
<b>Descripción</b>	Como Usuario Administrador de Becas, quiero descargar los documentos que los postulantes subieron en el MGT durante la Etapa de Postulación para poder revisar la documentación adjuntada y decidir si es pertinente la aprobación de la Etapa de Revisión.	
Nro	Escenario	Criterio de aceptación
1	<ul style="list-style-type: none"> <li>• Poder descargar la documentación de los postulantes en las Etapas de Revisión y Selección.</li> </ul>	<ul style="list-style-type: none"> <li>• Que exista un botón para descargar documentos en la Etapa de Revisión.</li> <li>• Que pueda descargar todos los documentos adjuntos de cada postulante.</li> </ul>

ID: H004 Ver trámite en el MGT		
<b>Usuario / Rol</b>	Usuario Administrador de Becas	<b>Prioridad</b>
		<b>Puntos estimados</b>



**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

<b>Descripción</b>	Como Usuario Administrador de Becas, quiero tener un enlace directo a la información del trámite de cada Postulante para poder revisar el trámite asociado..	
<b>Nro</b>	<b>Escenario</b>	<b>Criterio de aceptación</b>
1	<ul style="list-style-type: none"><li>• Poder abrir el trámite asociado de cada uno de los postulantes en las Etapas de Revisión y Selección.</li></ul>	<ul style="list-style-type: none"><li>• Que exista un botón en las Etapas de Revisión y Selección, que redirija al trámite del Postulante aceptado..</li></ul>

ID: H005 Finalizar Etapas		
<b>Usuario / Rol</b>	Usuario Administrador de Becas	<b>Prioridad</b>
		<b>Puntos estimados</b>
<b>Descripción</b>	Como Usuario Administrador de Becas, quiero poder finalizar las Etapas de Revisión y Selección, para evitar accidentalmente cambiar la información de Etapas pasadas.	
<b>Nro</b>	<b>Escenario</b>	<b>Criterio de aceptación</b>
1	<ul style="list-style-type: none"><li>• Poder finalizar las Etapas de Revisión y Selección respectivamente.</li><li>• Prohibir la actualización de un postulante que se encuentre en una Etapa finalizada</li></ul>	<ul style="list-style-type: none"><li>• Que exista un botón en las Etapas de Revisión y Selección, que finalice la Etapa correspondiente.</li><li>• Los datos de las Etapas finalizadas no pueden ser alterados por los usuarios del módulo.</li></ul>

**UNL**Universidad  
Nacional  
de LojaDirección de  
Tecnologías de Información

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

ID: H006 Actualizar Estados de Becas		
<b>Usuario / Rol</b>	Usuario Administrador de Becas	<b>Prioridad</b>
		<b>Puntos estimados</b>
<b>Descripción</b>	Como Usuario Administrador de Becas, quiero suspender y terminar Becados en la Etapa de Adjudicación, notificando de dicho cambio a los mismos, para evitar pagos que no corresponden a estudiantes que pierden los beneficios de su beca.	
Nro	Escenario	Criterio de aceptación
1	<ul style="list-style-type: none"><li>• Poder suspender y terminar Becas en la Etapa de Adjudicación.</li><li>• Notificar a los estudiantes sobre esta actualización.</li></ul>	<ul style="list-style-type: none"><li>• Debe existir un botón que permita suspender y terminar becas.</li><li>• Se debe mostrar un modal que pide confirmación para realizar dicha acción.</li><li>• Debe existir un campo para ingresar detalles específicos en caso de ser necesario.</li></ul>

ID: H007 Visualizar Postulaciones y Becas		
<b>Usuario / Rol</b>	Usuario Administrador de Becas	<b>Prioridad</b>
		<b>Puntos estimados</b>
<b>Descripción</b>	Como Usuario Administrador de Becas, quiero visualizar la información de los postulantes y becarios, para administrar sus estados por Etapa.	
Nro	Escenario	Criterio de aceptación
1	<ul style="list-style-type: none"><li>• Al visitar las Etapas de</li></ul>	<ul style="list-style-type: none"><li>• En la Etapa de Revisión y Selección se debe presentar el listado de</li></ul>



UNL

Universidad Nacional de Loja

Dirección de Tecnologías de Información

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	<p>Revisión, Selección deben listarse los datos de los Postulantes pertenecientes a la correspondiente Etapa.</p> <ul style="list-style-type: none"> <li>• Al visitar la Etapa de Adjudicación deben listarse los datos de los Becados pertenecientes a esta Etapa.</li> <li>• Al aplicar filtros el listado debe mostrar únicamente los registros comunes al o los filtros aplicados.</li> <li>• Al ordenar por las columnas habilitadas, el listado debe actualizarse.</li> <li>• Al ingresar un número de identificación o alguno de los nombres de un estudiante, el listado debe mostrar solo los registros que coinciden con los criterios de búsqueda.</li> </ul>	<p>Postulantes que se encuentren en cada Etapa respectivamente.</p> <ul style="list-style-type: none"> <li>• En la Etapa de Adjudicación se debe presentar el listado de Becados que se encuentren en esa Etapa.</li> <li>• Se debe poder filtrar, ordenar y paginar cada listado.</li> <li>• Se debe poder buscar registros específicos por su número de identificación o sus nombres.</li> </ul>
--	--	--

<b>ID: H008</b>	<b>Generar reportes</b>		
<b>Usuario / Rol</b>	Usuario Administrador de Becas	<b>Prioridad</b>	
		<b>Puntos estimados</b>	
<b>Descripción</b>	Como Usuario Administrador de Becas, quiero generar reportes de cada una de las Etapas de acuerdo a los filtros		



UNL

Universidad Nacional de Loja

Dirección de Tecnologías de Información

### ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	que aplique, para enviar listado de becados activos al Departamento Financiero, enviar listado de becados pendientes de pago de los meses ya transcurridos al Departamento Financiero o para proporcionar información de sobre Postulaciones y Becas en caso me sea requerida.	
Nro	Escenario	Criterio de aceptación
1	<ul style="list-style-type: none"> <li>Al visitar cada una de las Etapas debe mostrarse un botón que permite descargar la información que se está visualizando.</li> <li>Al pulsar sobre el botón generar reporte, se debe descargar automáticamente el reporte.</li> <li>Poder aplicar filtros y presionar sobre el botón generar reporte para su descarga.</li> </ul>	<ul style="list-style-type: none"> <li>Las Etapas de Revisión, Selección y Adjudicación deben permitir descargar reportes de acuerdo a los filtros que se apliquen o no.</li> <li>En la Etapa de Concesión se debe poder elegir el mes correspondiente o el estado de pago.</li> <li>El reporte debe corresponder a la Etapa elegida.</li> <li>El reporte debe descargarse en formato excel.</li> </ul>

<b>ID: H009</b>	<b>Agregar mensajes frecuentes</b>		
<b>Usuario / Rol</b>	Usuario Administrador de Becas	<b>Prioridad</b>	
		<b>Puntos estimados</b>	
<b>Descripción</b>	Como Usuario Administrador de Becas, quiero guardar mensajes frecuentes para la Etapa de Revisión, Selección y Adjudicación, para que me facilite dar motivos comunes de aprobación o rechazo en las Etapas de Revisión y Selección, y dar motivos de suspensión o terminación en la Etapa de Adjudicación.		



UNL

Universidad Nacional de Loja

Dirección de Tecnologías de Información

### ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

Nro	Escenario	Criterio de aceptación
1	<ul style="list-style-type: none"> <li>En el apartado configuraciones, en las Configuraciones de cada Etapa, habrá un campo de texto y un botón que permiten agregar mensajes frecuentes.</li> </ul>	<ul style="list-style-type: none"> <li>Se debe poder agregar mensajes para cada una de las Etapas de Revisión, Selección y Adjudicación.</li> </ul>

ID: H010		Actualizar mensajes frecuentes	
Usuario / Rol	Usuario Administrador de Becas	Prioridad	
		Puntos estimados	
Descripción	Como Usuario Administrador de Becas, quiero actualizar mensajes frecuentes para la Etapa de Revisión y Selección, para agregar o cambiar información del mensaje frecuente seleccionado.		
Nro	Escenario	Criterio de aceptación	
1	<ul style="list-style-type: none"> <li>En el apartado configuraciones, en las Configuraciones de cada Etapa, al seleccionar un mensaje habrá un botón que permite actualizar mensajes frecuentes.</li> </ul>	<ul style="list-style-type: none"> <li>Se debe poder actualizar un mensaje frecuente para cada una de las Etapas de Revisión, Selección y Adjudicación.</li> </ul>	

**UNL**Universidad  
Nacional  
de LojaDirección de  
Tecnologías de Información

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

ID: H011		Eliminar mensajes frecuentes	
Usuario / Rol	Usuario Administrador de Becas	Prioridad	
		Puntos estimados	
Descripción	Como Usuario Administrador de Becas, quiero eliminar mensajes frecuentes para la Etapa de Revisión y Selección, para deshacerme de mensajes escritos por error o limpiar mensajes en desuso.		
Nro	Escenario	Criterio de aceptación	
1	<ul style="list-style-type: none"><li>En el apartado configuraciones, en las Configuraciones de cada Etapa, al seleccionar un mensaje habrá un botón que permite eliminar mensajes frecuentes.</li></ul>	<ul style="list-style-type: none"><li>Se debe poder eliminar cada mensaje frecuente agregado para cada una de las Etapas de Revisión, Selección y Adjudicación.</li></ul>	

ID: H012		Marcar como favorito mensaje frecuente	
Usuario / Rol	Usuario Administrador de Becas	Prioridad	
		Puntos estimados	
Descripción	Como Usuario Administrador de Becas, quiero marcar como favorito un mensaje frecuente perteneciente a las Etapas de Revisión, Selección y Adjudicación respectivamente, para poder utilizarlo por defecto sino proporciono un motivo por cada acción.		
Nro	Escenario	Criterio de aceptación	





ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

1	<ul style="list-style-type: none"> <li>En el apartado configuraciones, en las Configuraciones de cada Etapa, al seleccionar un mensaje habrá un botón que permite marcar un mensaje como favorito.</li> </ul>	<ul style="list-style-type: none"> <li>Se debe poder marcar un mensaje frecuente como favorito para cada una de las Etapas de Revisión, Selección y Adjudicación.</li> <li>El mensaje favorito correspondiente debe enviarse en la notificación al estudiante.</li> </ul>
---	---	---

ID: H013 Actualizar montos de pago para los tipos de beca		
Usuario / Rol	Usuario Administrador de Becas	Prioridad
		Puntos estimados
Descripción	Como Usuario Administrador de Becas, quiero actualizar el valor de pago correspondiente por cada tipo de beca, para que a partir del momento de la actualización entre en vigencia el nuevo monto de pago requerido por cada tipo de beca.	
Nro	Escenario	Criterio de aceptación
1	<ul style="list-style-type: none"> <li>En el apartado configuraciones, en las Configuraciones de Becas, existirá un campo de texto que permite ingresar un nuevo valor de pago y un botón que permite actualizar los valores de pago de las becas..</li> </ul>	<ul style="list-style-type: none"> <li>El monto de pago de cada tipo de beca debe ser parametrizable.</li> <li>Al actualizar el monto de pago los registros de pago futuros deben utilizar el nuevo monto de pago.</li> <li>Los registros de pago existentes hasta antes de la actualización deben mantener el valor anterior.</li> </ul>

**UNL**Universidad  
Nacional  
de LojaDirección de  
Tecnologías de Información

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

ID: H014		Visualizar estadísticas generales de los becados	
Usuario / Rol	Usuario Administrador de Becas	Prioridad	
		Puntos estimados	
Descripción	Como Usuario Administrador de Becas, quiero visualizar estadísticas generales acerca de las postulaciones y becas para proveer cifras estadísticas de manera ágil y oportuna.		
Nro	Escenario	Criterio de aceptación	
1	<ul style="list-style-type: none"><li>En el apartado tablero, se mostrarán las estadísticas generales relacionadas a las postulaciones y becas.</li></ul>	<ul style="list-style-type: none"><li>El tablero debe mostrar el número de becas existentes, el número de postulantes, el número de becas por su tipo, el balance general de pagos, el número de postulantes rechazados y aceptados, el número de becados suspendidos y terminados.</li><li>Debe presentar gráficas del número de becas por facultad y gráfica comparativa entre becas por periodo académico.</li><li>Si no existen datos deben mostrarse las estadísticas en 0.</li></ul>	

ID: H015		Iniciar sesión	
Usuario / Rol	Usuario Administrador de Becas, Usuario de Tesorería, Usuario de Departamento Financiero	Prioridad	
		Puntos estimados	
Descripción	Como Usuarios del MGB, queremos iniciar sesión con nuestras credenciales institucionales para poder realizar nuestras actividades administrativas dentro del módulo.		



UNL

Universidad Nacional de Loja

Dirección de Tecnologías de Información

### ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

Nro	Escenario	Criterio de aceptación
1	<ul style="list-style-type: none"> <li>Los correos de los servidores públicos institucionales tendrán acceso al MGB.</li> </ul>	<ul style="list-style-type: none"> <li>Las cuentas de los servidores públicos que se encargan de los procesos que intervienen en la Gestión de Becas, deben tener acceso al MGB.</li> <li>Cada usuario debe poder visualizar la información que le concierne.</li> <li>Cada usuario debe tener sus privilegios de acuerdo al rol definido dentro del MGB.</li> </ul>

ID: H016		Notificar resultados de postulación	
<b>Usuario / Rol</b>	Usuario Administrador de Becas	<b>Prioridad</b>	
		<b>Puntos estimados</b>	
<b>Descripción</b>	Como Usuario Administrador de Becas, quiero notificar a los postulantes los resultados de su postulación, y al Usuario de Tesorería notificar la existencia de nuevos becados, para que los estudiantes conozcan los resultados de su postulación y el Usuario de Tesorería sepa que puede empezar el proceso de registro de nuevos becados.		
Nro	Escenario	Criterio de aceptación	
1	<ul style="list-style-type: none"> <li>En el correo de un postulante llegará la notificación de los resultados de su postulación.</li> <li>En el correo del Usuario de Tesorería llegará una notificación de nuevos becados.</li> <li>En el MGB aparecerá una notificación para el Usuario</li> </ul>	<ul style="list-style-type: none"> <li>Al finalizar la Etapa de Selección de cada Periodo Académico de Postulación, los postulantes deben recibir una notificación en el correo electrónico institucional usado en la Etapa de Postulación, con los resultados obtenidos.</li> <li>Al finalizar la Etapa de Selección de cada Periodo Académico de Postulación. El Usuario de Tesorería recibirá una notificación vía correo electrónico y dentro del MGB.</li> </ul>	



UNL

Universidad Nacional de Loja

Dirección de Tecnologías de Información

ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	de Tesorería.	
--	---------------	--

ID: H017 Guardar registro de becados en el sistema financiero		
Usuario / Rol	Usuario de Tesorería	Prioridad
		Puntos estimados
Descripción	Como Usuario de Tesorería, quiero guardar la confirmación de registro de los nuevos becados en el sistema financiero para llevar el control de los becados ya registrados así como crearles los registros de pago para el periodo académico vigente.	
Nro	Escenario	Criterio de aceptación
1	<ul style="list-style-type: none"> <li>Al iniciar sesión con el rol Usuario Tesorería, podrá visualizar el listado de nuevos becados.</li> <li>Al guardar la confirmación de registro, se crearán los registros de pago de los nuevos becados.</li> <li>Al finalizar la Etapa de Confirmación de Registro se creará los registros de pagos de los becados activos que aún no han sido creados.</li> </ul>	<ul style="list-style-type: none"> <li>El Usuario de Tesorería debe ser capaz de visualizar el listado de nuevos becados.</li> <li>El usuario de Tesorería debe poder guardar en el MGB la confirmación de registro del sistema financiero.</li> <li>Al guardar la confirmación de registro se deben crear los registros de pago para el periodo académico en curso de los nuevos becados.</li> <li>Al finalizar la Etapa de Confirmación de Registro, el módulo debe crear los registros de pago de los becados activos que no posean registros de pago.</li> </ul>



UNL

Universidad Nacional de Loja

Dirección de Tecnologías de Información

ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

ID: H018		Registro de Pago de Becas	
<b>Usuario / Rol</b>	Usuario de Tesorería, Usuario de Departamento Financiero	<b>Prioridad</b>	
		<b>Puntos estimados</b>	
<b>Descripción</b>	Como Usuario de Departamento Financiero, quiero importar los registros de devengado de becas para llevar el control de los becados a los que ya se ha ordenado el pago.		
Nro	Escenario	Criterio de aceptación	
1	<ul style="list-style-type: none"> <li>Al iniciar sesión con el rol Usuario Departamento Financiero, podrá visualizar el listado de registros de pago pendiente.</li> <li>Al importar un archivo devengado, se actualizarán los registros de pago.</li> <li>Si al importar se detectan errores, el módulo mostrará en una tabla la lista de errores.</li> </ul>	<ul style="list-style-type: none"> <li>El Usuario de Departamento Financiero debe ser capaz de visualizar el listado de registros de pago pendiente.</li> <li>El usuario del Departamento Financiero debe poder importar los registros de devengado de acuerdo al formato acordado.</li> <li>Si el archivo a importar contiene errores el módulo debe ser capaz de indicar dichos errores.</li> </ul>	
<b>Descripción</b>	Como Usuario de Tesorería, quiero importar los registros de pago de becas para llevar el control de los becados a los que ya se ha realizado el pago.		
Nro	Escenario	Criterio de aceptación	
2	<ul style="list-style-type: none"> <li>Al iniciar sesión con el rol Usuario Tesorería, podrá</li> </ul>	<ul style="list-style-type: none"> <li>El Usuario de Tesorería debe ser capaz de visualizar el listado de registros de pago pendiente.</li> </ul>	



UNL

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información


## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

	<p>visualizar el listado de registros de pago pendiente.</p> <ul style="list-style-type: none"><li>● Al importar un archivo de pago, se actualizarán los registros de pago.</li><li>● Si al importar se detectan errores, el módulo mostrará en una tabla la lista de errores.</li></ul>	<ul style="list-style-type: none"><li>● El usuario de Tesorería debe poder importar los registros de pago de acuerdo al formato acordado.</li><li>● El usuario de Tesorería debe poder actualizar individualmente los registros de pago seleccionando un registro desde el listado.</li><li>● Si el archivo a importar contiene errores el módulo debe ser capaz de indicar dichos errores.</li></ul>
--	--	---



## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

### • Aprobaciones

Nombre y Apellido	Cargo / Rol	Departamento	Firma
Pablo F. Ordoñez Ordoñez, Mg.Sc.	Director de Tesis	Carrera de Ingeniería en Sistemas	 <p>CN=PABLO FERNANDO ORDONEZ ORDONEZ, SERIALNUMBER=29032118 5443, OU=ENTIDAD DE CERTIFICACION DE INFORMACION, O=SECURITY DATA S.A. 2, C=EC Director de Tesis</p>
Ing. Danny Muñoz	Especialista en Sistemas de Información	Dirección de Tecnologías de Información	 <p>Firmado electrónicamente por: <b>DANNY EMANUEL MUNOZ FLORES</b></p>
Ing. Máximo Álvarez	Analista en Sistemas de Información 1	Dirección de Tecnologías de Información	 <p>Firmado electrónicamente por: <b>MAXIMO ANDRES ALVAREZ PACHECO</b></p>
Lcda. Marlene Valdez Pardo	Cliente	Unidad de Bienestar Universitario	 <p>Firmado electrónicamente por: <b>CARMEN MARLENE VALDEZ PARDO</b></p>
Jackson Guzmán	Tesista	Carrera de Ingeniería en Sistemas	<p><b>Jackson Guzmán</b></p> <p>Digitally signed by Jackson Guzmán Date: 2022.03.11 15:46:14 -05'00'</p>

## **Anexo 4. Manual de Usuario para el uso del MGB**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Manual de Usuario<sup>6</sup>

---

<sup>6</sup> Versión original en: <https://drive.google.com/Manual-Usuario-MGB.pdf>





**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

[MANUAL DE USUARIO]

<<DIRECCIÓN DE BIENESTAR UNIVERSITARIO>>  
<<SUBDIRECCIÓN DE CONTABILIDAD FINANCIERA>>  
<<SUBDIRECCIÓN DE TESORERÍA FINANCIERA>>

# MÓDULO DE GESTIÓN DE BECAS - SIAAF

Versión 22.08

**Elaborado por:**

*Jackson Andrews Guzmán Tituana*

**Revisado por:**

*Danny Emanuel Muñoz Flores  
Johana Elizabeth Cueva Domínguez*

**Aprobado por:**

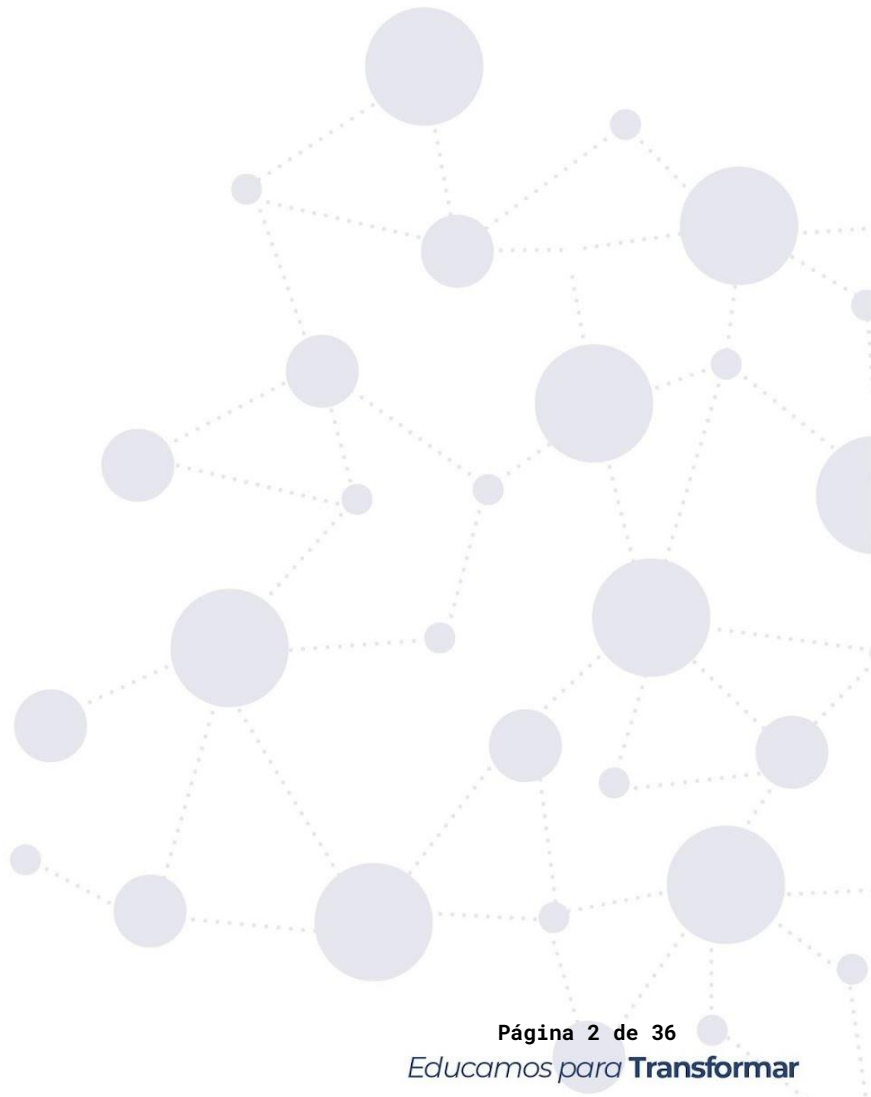
*Jhon Alexander Calderón Sanmartín*

Agosto 2022



## HISTORIAL DE CAMBIOS

Versión	Fecha	Responsable	Descripción de la modificación
22.08	29/08/2022	Tesista Jackson Guzmán	Documento Inicial





## ÍNDICE DE CONTENIDOS

<b>INTRODUCCIÓN</b>	<b>4</b>
<b>OBJETIVO</b>	<b>5</b>
<b>DIRIGIDO A</b>	<b>5</b>
<b>CAPÍTULO I</b>	<b>6</b>
<b>1. GENERALIDADES</b>	<b>7</b>
1.1. Diagrama General de Procesos	7
1.2. Ingreso al Sistema	8
1.3. Navegación del Módulo	9
1.4. Filtrado de Registros	9
1.5. Número de Registros a Mostrar	10
1.6. Búsqueda de Registros	10
1.7. Ordenación de Registros	10
1.8. Descargar Reporte para Importación de Devengado	11
<b>CAPÍTULO II</b>	<b>12</b>
<b>1. ETAPA DE REVISIÓN Y SELECCIÓN</b>	<b>13</b>
1.1. Calificar ficha socioeconómica	14
1.2. Revisar documentación de postulante	15
1.3. Aprobar Postulantes	16
1.4. Rechazar Postulantes	17
1.5. Ver trámite de postulación	17
1.6. Generar Reporte	18
1.7. Finalizar Etapas de Revisión y Selección	18
<b>3. ETAPA DE NOTIFICACIÓN</b>	<b>19</b>
<b>4. ETAPA DE ADJUDICACIÓN</b>	<b>20</b>
4.1. Suspender Becas	20
4.2. Terminar Becas	21
4.2. Reactivar Becas	21
<b>5. ETAPA DE CONCESIÓN</b>	<b>22</b>
5.1. Generar Reporte Detallado de Concesión	22
5.2. Sincronizar Becas Activas de Periodos Académicos Previos	22
<b>6. CONFIGURACIONES</b>	<b>23</b>
6.1. Agregar Mensajes Frecuentes	23
6.2. Actualizar Mensajes Frecuentes	24
6.3. Eliminar Mensajes Frecuentes	25
<b>CAPÍTULO III</b>	<b>26</b>
<b>1. ETAPA DE ADJUDICACIÓN</b>	<b>27</b>

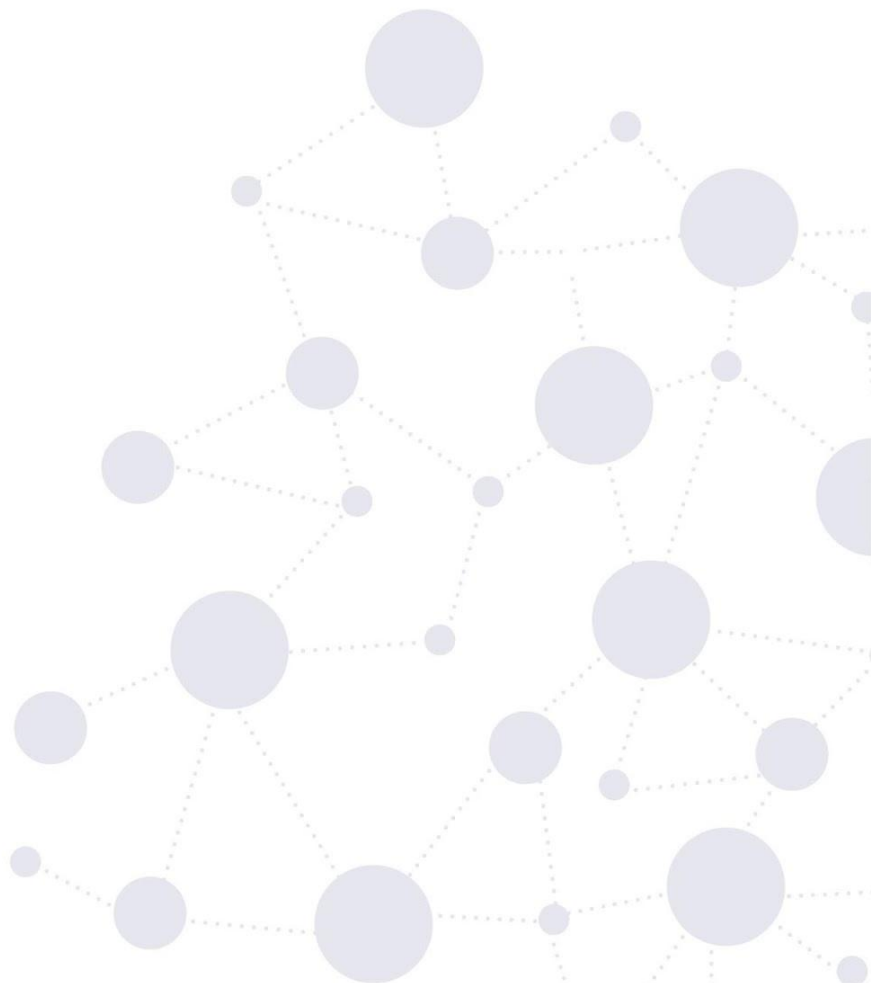


**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

1.1. Confirmar Registro de Becado en el Sistema Financiero	27
1.2. Armar Reporte de Importación de Pago	27
<b>2. ETAPA DE CONCESIÓN</b>	<b>28</b>
2.1. Importar Registros de Devengado o Pago	28
2.2. Actualización Manual de Registros de Concesión	31
<b>CAPÍTULO IV</b>	<b>33</b>
<b>1. ETAPA DE CONCESIÓN</b>	<b>34</b>
1.1. Armar Reporte de Importación de Devengado	34
<b>RECOMENDACIONES:</b>	<b>35</b>
<b>FIRMAS DE RESPONSABILIDAD</b>	<b>35</b>





UNL

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

## INTRODUCCIÓN

El Sistema de Información Académico, Administrativo y Financiero (SIAAF) es una plataforma que ha permitido automatizar diversos procesos de la Universidad Nacional de Loja, de manera integrada, flexible y estandarizada. De forma paulatina se han ido incorporando nuevas funcionalidades, ello con el fin de implementar mejoras que permitan optimizar procesos en los diferentes ámbitos como: académico, administrativo y financiero.

En cuanto al ámbito académico, se ha incorporado el Módulo de Gestión de Becas, el cual se enfoca en coadyuvar en los procesos de seguimiento y control de becas, ello con el fin de brindar al personal involucrado en la Gestión de Becas una herramienta que les permita llevar el registro de la información relacionada con los procesos antes descritos. Además a futuro pretende mejorar la comunicación existente con los estudiantes en cuanto a los beneficios de sus becas.

El presente manual está dividido en capítulos en donde se explica las acciones que debe realizar cada usuario en relación con el proceso de gestión de becas. Cada capítulo está relacionado a procesos comunes que debe cumplir el usuario involucrado de tal forma que este proceso se cumpla y pueda dar continuidad a la siguiente etapa si la misma es finalizada, por tal motivo es recomendable que cada usuario involucrado en el proceso sea cauteloso en revisar las acciones a realizar con la finalidad de no cometer errores que requieran accionar técnico.

Se debe considerar las notificaciones que se presentan y estar pendiente en los estados que se encuentran en cada etapa de tal forma que no se generen inconsistencias.

### OBJETIVO

Guiar en los procesos a seguir a través del **Módulo de Gestión de Becas (MGB)** en las Etapas de Revisión, Selección, Adjudicación y Concesión de Becas, siguiendo los lineamientos emitidos en los reglamentos y estatutos de la Universidad de Loja.

### DIRIGIDO A

Este documento es una guía para servidores universitarios con responsabilidades como: sección becas de la unidad de bienestar universitario, departamento financiero y subdirección de tesorería de la Universidad Nacional de Loja.



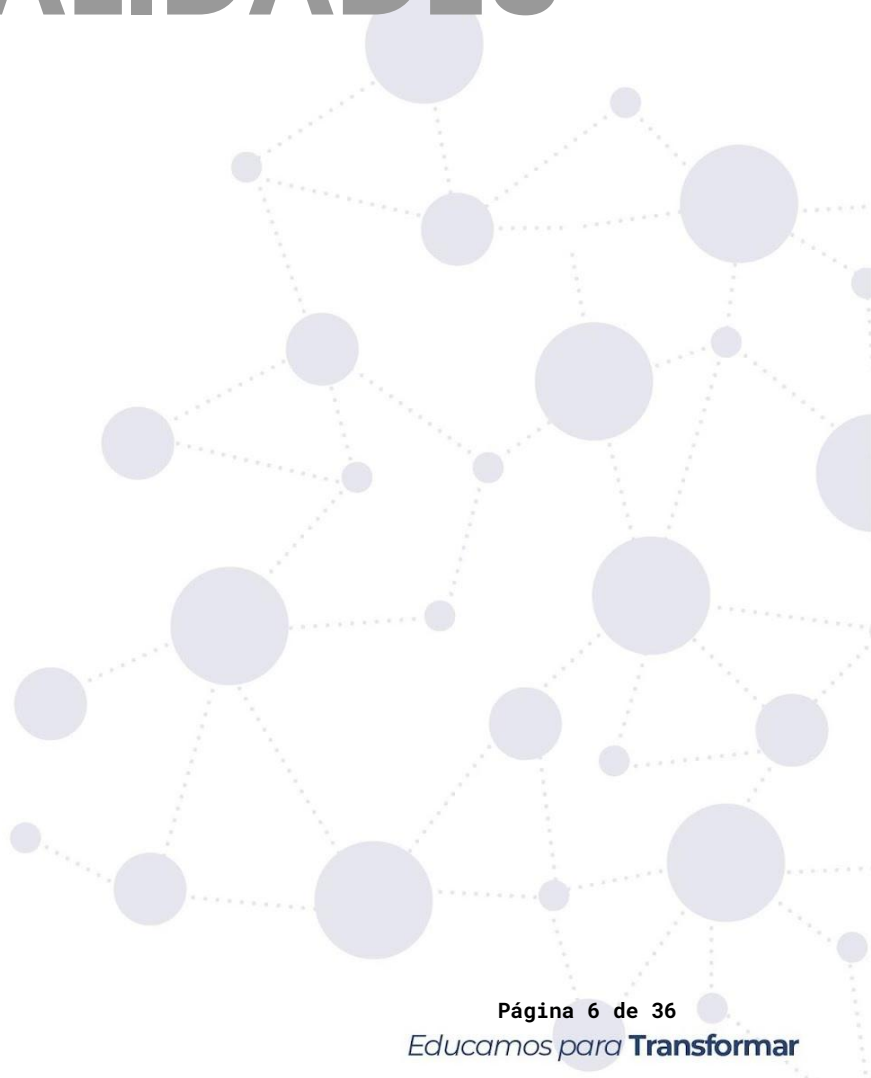
**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

# CAPÍTULO I

# GENERALIDADES



# 1. GENERALIDADES

## 1.1. Diagrama General de Procesos

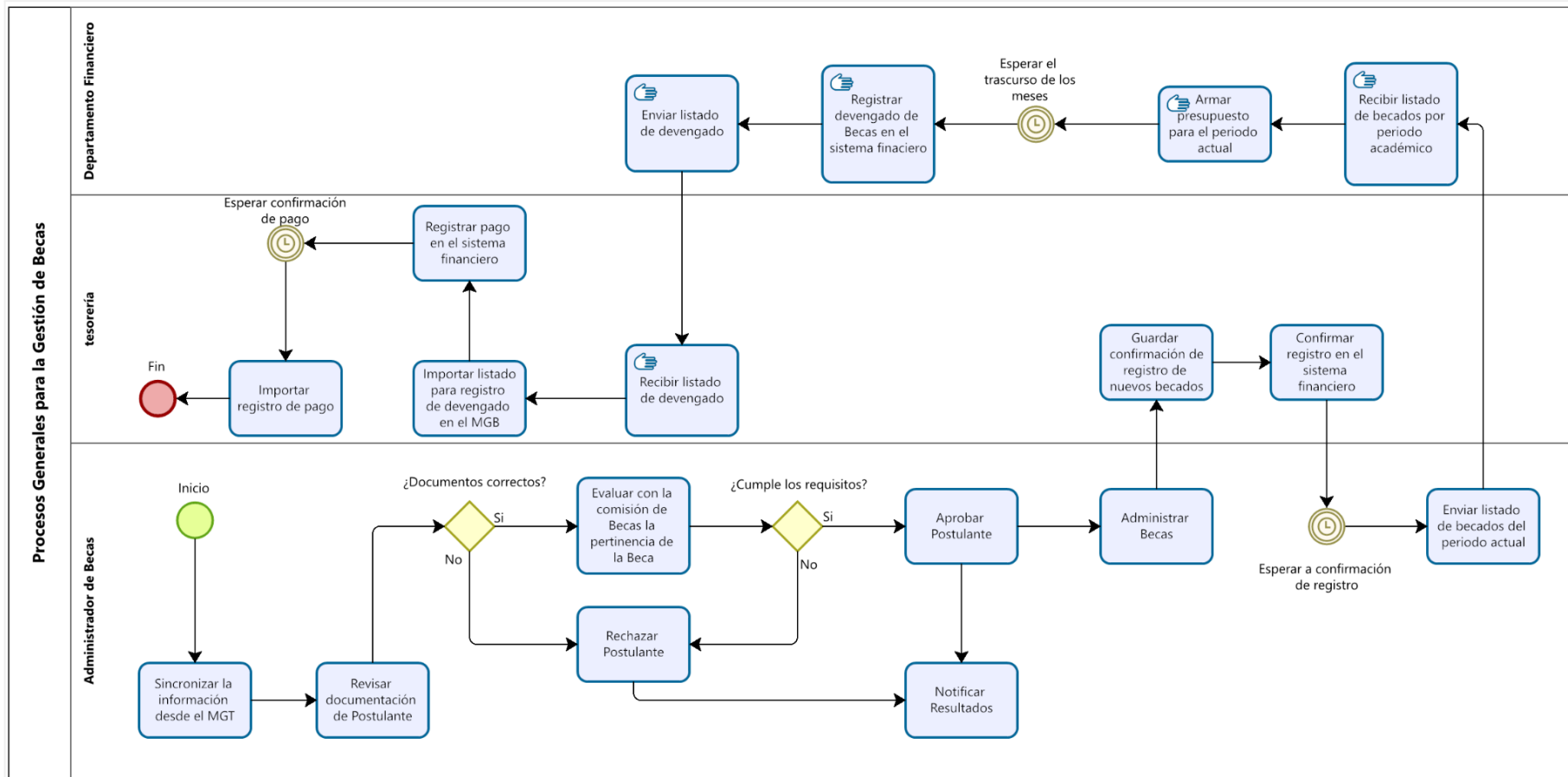


Figura 1: Diagrama general de procesos para la Gestión de Becas.

## 1.2. Ingreso al Sistema

**Paso 1:** Los servidores universitarios deben abrir el navegador de su preferencia y en la barra de direcciones ingresar la dirección <https://siaaf.unl.edu.ec> y digitar las credenciales de acceso (correo institucional y clave del SIAAF).



Sistema de Información Académico Administrativo Financiero - SIAAF

UNL Universidad Nacional de Loja

INICIAR SESIÓN

Correo electrónico institucional

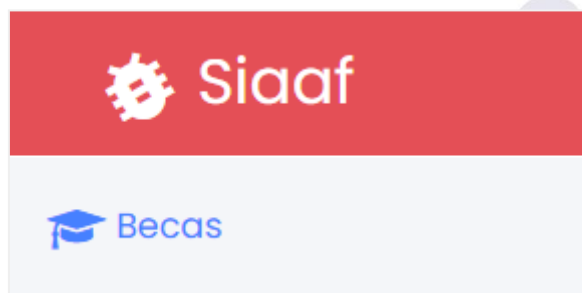
Contraseña

Ingresar

Servicio de Autenticación Centralizada Recuperar contraseña

**Figura 2:** Ingreso al SIAAF. [Captura de pantalla]

Paso 2: En el menú lateral izquierdo seleccione la opción [Becas], enlace que lo redirige al Módulo de Gestión de Becas.



**Figura 3:** Ingreso al Módulo de Gestión de Becas. [Captura de pantalla]



### 1.3. Navegación del Módulo

Todos los usuarios que accedan al Módulo de Gestión de Becas en la parte izquierda podrán encontrar el menú de navegación hacia las diferentes Etapas de acuerdo al rol de Usuario que posean.

#### 1.3.1. Responsabilidades de acuerdo a los roles

**Administrador de Becas.-** Responsable de las Etapas de Revisión, Selección, Adjudicación, Concesión y el apartado de Configuraciones.

Etapa de Revisión:

- Revisar documentación de postulantes.
- Asignar calificación a las fichas socioeconómicas de los postulantes.
- Aprobar o Rechazar postulantes de acuerdo a la documentación revisada.
- Finalizar la etapa para continuar con el proceso de selección de los postulantes aprobados.

Etapa de Selección:

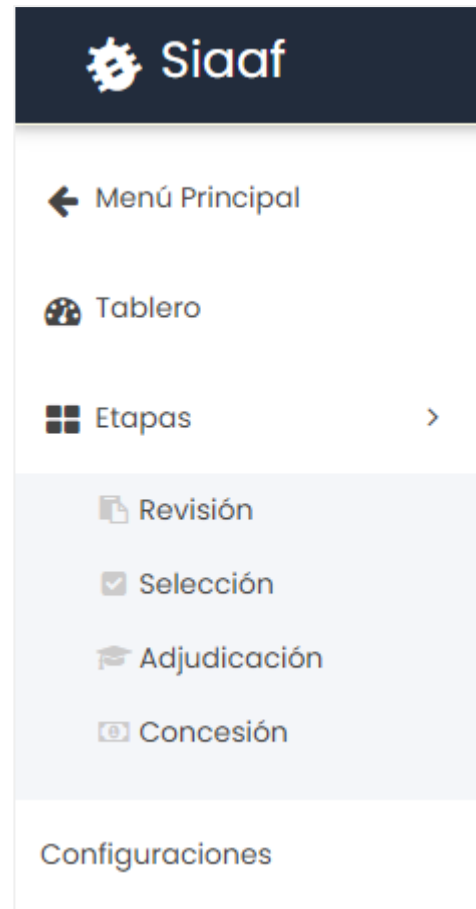
- Rechazar postulantes.
- Finalizar la etapa para continuar con el proceso de adjudicación de los postulantes aprobados y notificar los resultados de la Etapa de Postulación.

Etapa de Adjudicación:

- Actualizar Estado de becas activas a suspendidas o terminadas según sea el caso.

Etapa de Concesión:

- Generar Reporte aplicando únicamente el filtro de Periodo Académico adecuarlo según se requiera y enviarlo a Contabilidad Financiera para que pueda armar el presupuesto anual de Concesión de Becas.



**Figura 3:** Navegación del módulo.  
[Captura de pantalla]



- Descargar Reporte para Concesión mes a mes o según se requiera, adecuarlo y enviarlo periódicamente a Tesorería Financiera y Contabilidad Financiera solicitando el devengado y pago correspondiente de las Becas Activas.
- Sincronizar Becas Activas actualizando las becas en la Etapa de Adjudicación según se lo requiera<sup>1</sup> y actualizar los montos para los registros de pago si de acuerdo al reglamento cambió el valor de un tipo de beca. Este proceso se realizará una única vez por periodo académico, una vez esté activo un nuevo periodo académico.

#### Configuraciones:

- Configurar los montos para el Registro de Pago si de acuerdo a un nuevo reglamento los montos para los tipos de becas cambiaron su valor, caso contrario omitir esta responsabilidad hasta que se lo requiera.
- Administrar mensajes frecuentes, agregando, actualizando, cambiando a favorito o eliminado según se lo requiera para ayudarse durante la Actualización de Estado de las Etapas.

**Tesorería Financiera:** responsable de las Etapas de Adjudicación y Concesión

#### Etapa de Adjudicación:

- Confirmar registro en el sistema financiero de las nuevas postulaciones.
- Actualizar Estado de Concesión de los Postulantes importando los registros de pago o actualizando manualmente.

**Contabilidad Financiera:** responsable de la Etapa de Adjudicación.

#### Etapa de Adjudicación:

- Constatar y verificar la información de los reportes recibidos por parte del usuario Administrador de Becas y generar reporte para ordenar el devengado de becas al

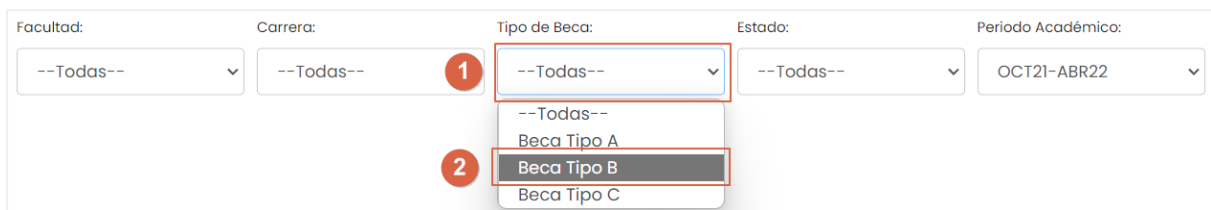
---

<sup>1</sup> Actualizar Estado de las Becas Activas en la Etapa de Adjudicación

usuario de Tesorería Financiera completando con el número de expediente.

## 1.4. Filtrado de Registros

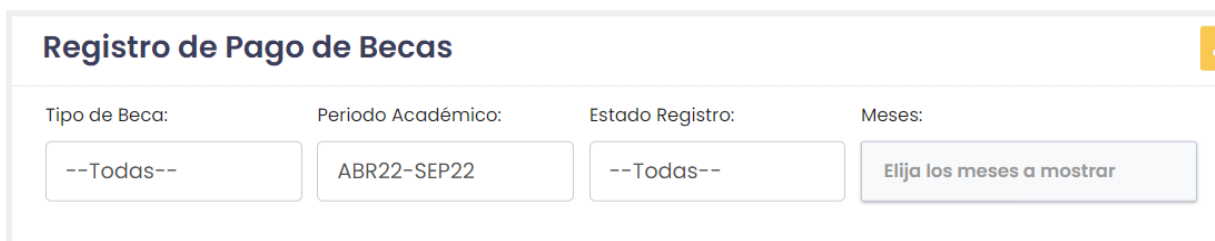
Todas las Etapas cuentan con una sección de filtros aplicables, para aplicar un filtro haga clic sobre el recuadro del filtro deseado, elija una de las opciones e inmediatamente la tabla actualizará la información mostrada de acuerdo al filtro aplicado.



The screenshot shows a filter interface with five dropdown menus: 'Facultad:', 'Carrera:', 'Tipo de Beca:', 'Estado:', and 'Período Académico:'. The 'Tipo de Beca:' dropdown is open, showing options: '--Todas--', 'Beca Tipo A', 'Beca Tipo B', and 'Beca Tipo C'. Red circles with numbers 1 and 2 highlight the dropdown menu and the 'Beca Tipo B' option respectively.

**Figura 4:** Filtros aplicables a las Etapas de Revisión, Selección y Adjudicación [Captura de Pantalla]

En la **[Etapa de Concesión]** para el listado de registros de pago existe un filtro de meses del cual se puede elegir uno o varios meses tanto para visualizar la información como para descargar el reporte.

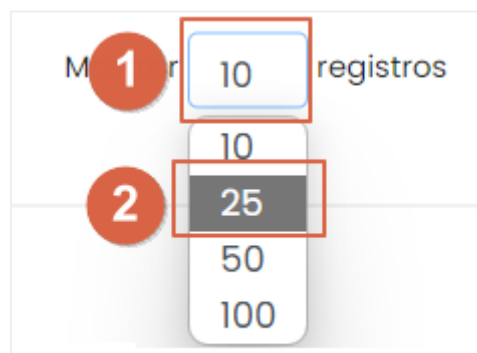


The screenshot shows the 'Registro de Pago de Becas' filter section. It includes four dropdown menus: 'Tipo de Beca:', 'Período Académico:', 'Estado Registro:', and 'Meses:'. The 'Meses:' dropdown is open, showing the option 'Elija los meses a mostrar'.

**Figura 5:** Filtros aplicables a la Etapa de Concesión [Captura de Pantalla]

## 1.5. Número de Registros a Mostrar

Es posible cambiar el número de registros a mostrar; por defecto se muestran 10 pero puede cambiarse a 25, 50 y 100 registros.

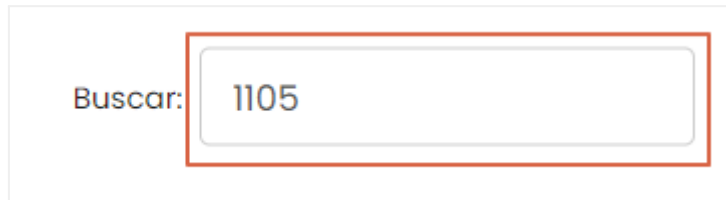


The screenshot shows a dropdown menu for the number of records to display. The options are 10, 25, 50, and 100. Red circles with numbers 1 and 2 highlight the dropdown menu and the '25' option respectively.

**Figura 6:** Cambiar el número de registros a mostrar [Captura de Pantalla]

## 1.6. Búsqueda de Registros

Es posible buscar un registro en específico ingresando su número de identificación, primer nombre, segundo nombre, primer apellido o segundo apellido.

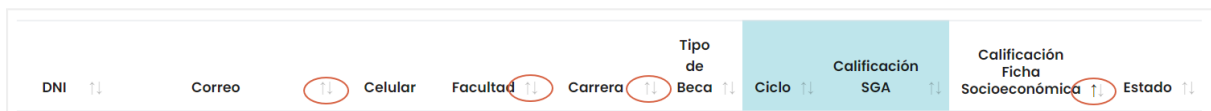


Buscar: 1105

**Figura 7:** Buscar registros por número de identificación o nombres [Captura de Pantalla]

## 1.7. Ordenación de Registros

En ciertas columnas de la tabla se presentan flechas que indican que esa columna puede ser ordenada en orden ascendente o descendente, al presionar sobre ellas.



DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
-----	--------	---------	----------	---------	--------------	-------	------------------	-----------------------------------	--------

**Figura 8:** Ordenar resultados en base a cierta columna [Captura de Pantalla]

## 1.8. Descargar Reporte para Concesión

Todos los usuarios tienen permiso para descargar el Reporte de Concesión. Esta acción permite descargar un reporte preparado con la información de los registros de pago filtrados de acuerdo a cómo se visualizan en ese preciso instante. Para descargar el Reporte de Concesión diríjase a la **[Etapa de Concesión]** desde el Menú de Becas, elija de entre las distintas opciones de filtros: Tipo de beca, Periodo Académico, Estado de Registro y Meses. Una vez aplicado los filtros deseados [véase figura 9], haga clic en el botón **[Reporte para Concesión]** que se muestra en la figura 10.

### Registro de Pago de Becas

Tipo de Beca:	Período Académico:	Estado Registro:	Meses:
Beca Tipo A	ABR21-SEP21	Pendiente	may

Mostrar 10 registros

Becado

Buscar:

May

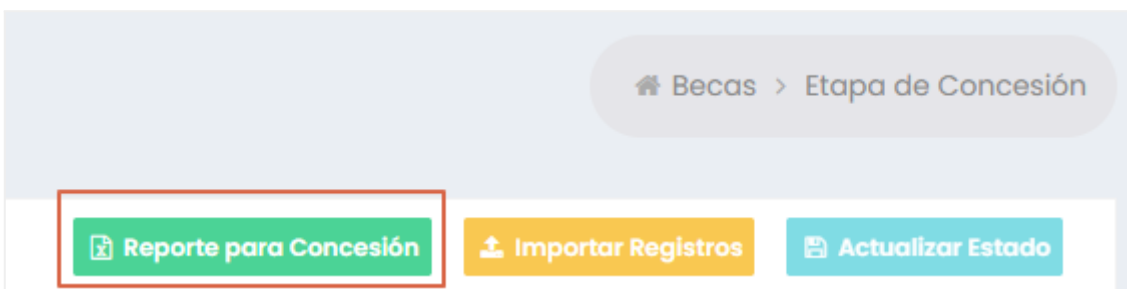
Jun

Jul

Ago

Sep

**Figura 9** Elegir los filtros deseados para descargar el reporte de concesión.



**Figura 10** Botón para descargar el reporte de importación de acuerdo a los filtros aplicados.

Al hacer clic en el botón **[Reporte para Concesión]**, se descargará un reporte [véase figura 45] que está preparado para completarlo con el **número de expediente** para poderlo usar durante la importación de registros de devengado y pago en el Módulo de Gestión de Becas.



**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

# CAPÍTULO II

# ADMINISTRACIÓN

# DE BECAS

# DIRECCIÓN DE

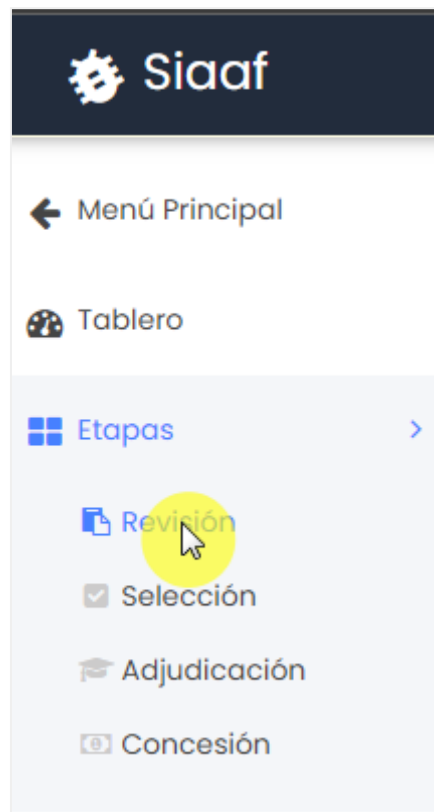
# BIENESTAR

# UNIVERSITARIO

## 1. ETAPA DE REVISIÓN Y SELECCIÓN

El Módulo de Gestión de Becas permite visualizar a los postulantes de los distintos periodos académicos de postulación en la **[Etapa de Revisión]**.

Para iniciar con el proceso de revisión de la documentación de los postulantes, debe acceder al Módulo de Gestión de Becas y seleccionar la opción **[Revisión]**, que se presenta en el menú **Etapas**:



**Figura 11:** Acceso a la Etapa de Revisión [Captura de Pantalla]

El Módulo de Gestión de Becas le presentará un listado de los **Postulantes a Beca para el periodo académico actual** [véase figura 12], se puede filtrar los resultados aplicando diversos filtros [véase figura 4], cambiar el número de registros a mostrar [véase figura 6], localizar algún registro en específico utilizando el buscador [véase figura 7] y ordenar los resultados en base a una columna [véase figura 8].





1859

**Listado de Postulaciones – Revisión de la Documentación** Actualizar Estado Descargar documentos Ver Trámite

Facultad: --Todas-- Carrera: --Todas-- Tipo de Beca: --Todas-- Estado: --Todas-- Período Académico: OCT21-ABR22

Mostrar 10 registros Buscar:

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica
Alanis Alanis Paula Escamilla	1173872852	berta27@unl.edu.ec	0908188897	FEIRNNR	INGENIERÍA AUTOMOTRIZ	Beca Tipo B	2	9.25	9
Alcántar Alcántar Nerea Rojo	1149407379	rafael52@unl.edu.ec	0942846946	FEAC	PEDAGOGÍA DE LA LENGUA Y LA LITERATURA	Beca Tipo B	2	8.24	7
Cabello Cabello Celia Morales	1199391991	reyes.nicolas@unl.edu.ec	0900006745	FSH	Psicología Clínica	Beca Tipo B	3	8.14	5
Carrasco Carrasco Nuria Villa	1199437383	zmoreno@unl.edu.ec	0929220071	FEAC	PSICOPEDAGOGÍA	Beca Tipo B	3	9.3	9
Carrero Carrero	1157961856	jan.lunque@unl.edu.ec	0974878003	FJSA	Trabajo Social	Beca Tipo B	3	9.59	7

Figura 12: Listado de Postulantes en la Etapa de Revisión [Captura de Pantalla]

## 1.1. Revisar documentación de postulante

Para revisar la documentación de los postulantes es importante asignarles el estado de revisión; para ello seleccione la cantidad de postulantes a revisarse, luego presione el botón **[Actualizar Estado]** y elija la opción **[Revisar]**, esto es útil para no perder de vista a los postulantes que se les está revisando la documentación.

Etapa de Revisión Becas > Etapa Revisión

**Listado de Postulaciones – Revisión de la Documentación** Actualizar Estado Descargar documentos Ver Trámite

Facultad: --Todas-- Carrera: --Todas-- Tipo de Beca: --Todas-- Estado: --Todas--

Mostrar 10 registros Buscar:

DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
1133054734	alvaro.pizarro@unl.edu.ec	0987756360	FEIRNNR	Ingeniería en Sistemas	Beca Tipo B	4	7.85	7.8	sin revisión
1147325226	arnau.sancho@unl.edu.ec	0993504501	FSH	MEDICINA	Beca Tipo B	4	8.49	9.1	sin revisión
1162471991	arnau39@unl.edu.ec	0926293585	FJSA	Derecho	Beca Tipo B	4	9.24	8.9	sin revisión

Figura 13: Asignar estado de revisión al postulante [Captura de Pantalla]

Con el estado actualizado a estado de revisión, se puede descargar la documentación del postulante eligiendo el postulante del cuál se desea obtener la documentación, y luego presionando sobre el botón **[Descargar documentos]**.



Etapa de Revisión Becas > Etapa Revisión

**Listado de Postulaciones – Revisión de la Documentación** Actualizar Estado Descargar documentos Ver Trámite

Facultad: --Todas-- Carrera: --Todas-- Tipo de Beca: --Todas-- Estado: --Todas-- Período Académico: OCT21-ABR22

Mostrar 10 registros Buscar:

DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
1133054734	alvaro.pizarro@unl.edu.ec	0987756360	FEIRNNR	Ingeniería en Sistemas	Beca Tipo B	4	7.85	7.8	revisando

Figura 14: Descargar documentación del postulante [Captura de Pantalla]

## 1.2. Calificar ficha socioeconómica

Para calificar la ficha socio-económica de un postulante, primero se debe seleccionar al postulante haciendo clic sobre la fila, luego se debe hacer clic sobre el botón de la columna **[Calificación Ficha Socioeconómica]** que indica un valor de 0 [véase figura 15].

DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
1133054734	alvaro.pizarro@unl.edu.ec	0987756360	FEIRNNR	Ingeniería en Sistemas	Beca Tipo B	4	7.85	0	sin revisión
1147325226	arnau.sancho@unl.edu.ec	0993504501	FSH	MEDICINA	Beca Tipo B	4	8.49	0	sin revisión

Figura 15: Seleccionar postulante a calificar [Captura de Pantalla]

Al hacer clic sobre el botón de la columna **[Calificación Ficha Socioeconómica]** se desplegará un componente con una entrada de texto en donde se debe ingresar la calificación de la ficha socioeconómica, y luego presionar sobre el botón de **[✓ validación ]**.

DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
1133054734	alvaro.pizarro@unl.edu.ec	0987756360	FEIRNNR	Ingeniería en Sistemas	Beca Tipo B	4	7.85	7.8	sin revisión

Figura 16: asignar calificación a la ficha socioeconómica [Captura de Pantalla]

El sistema registra la calificación dada y devuelve una mensaje de respuesta exitoso [véase figura 17].



1859

DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
1133054734	alvaro.pizarro@unl.edu.ec	0987756360	FEIRNRR	Ingeniería en Sistemas	Beca Tipo B	4	7.85	7.8	sin revisión
1147325226	arnau.sancho@unl.edu.ec	0993504501	FSH	MEDICINA	Beca Tipo B	4	8.49	0	sin revisión
1162471991	arnau39@unl.edu.ec	0926293585	FJSA	Derecho	Beca Tipo B	4	9.24	0	sin revisión
1136785078	asensio.pablo@unl.edu.ec	0987233830	FJSA	Derecho	Beca Tipo B	4	7.05	Ficha Socioeconómica calificada correctamente	

Figura 17: Confirmación de asignación de calificación a la ficha socioeconómica satisfactoria [Captura de Pantalla]

### 1.3. Aprobar Postulantes

Para Aprobar Postulantes en la [Etapa de Revisión], primero seleccione a los postulantes que se desea aprobar, luego presione el botón [Actualizar Estado] y elegir la opción [Aprobar]; esto desplegará un modal en el que se muestra un cuadro selector para elegir un mensaje frecuente de aprobación; por defecto en el área de texto se muestra el mensaje elegido como favorito para aprobar postulantes, se puede reemplazar el texto con un mensaje más específico si es necesario. En la parte inferior se muestra el listado de postulantes a aprobarse y se solicita confirmación presionando el botón [Aprobar].

**Aprobar la etapa de revisión de la Beca** ✕

**ⓘ Por defecto se muestra el mensaje configurado como favorito para esta acción. Sin embargo puede elegir de entre la lista de mensaje configurados o proporcionar un mensaje personalizado editando el cuadro inferior**

Mensajes frecuentes para aprobar becas

Bienvenido a la comunidad de Becados

Los mensajes ayudan a los postulantes a entender los motivos

Mensaje de notificación:

Bienvenido a la comunidad de Becados

El mensaje que se proporcione se enviará a los siguientes estudiantes:

ALVARADO GONZALEZ MAYURY SILVANA

Cancelar

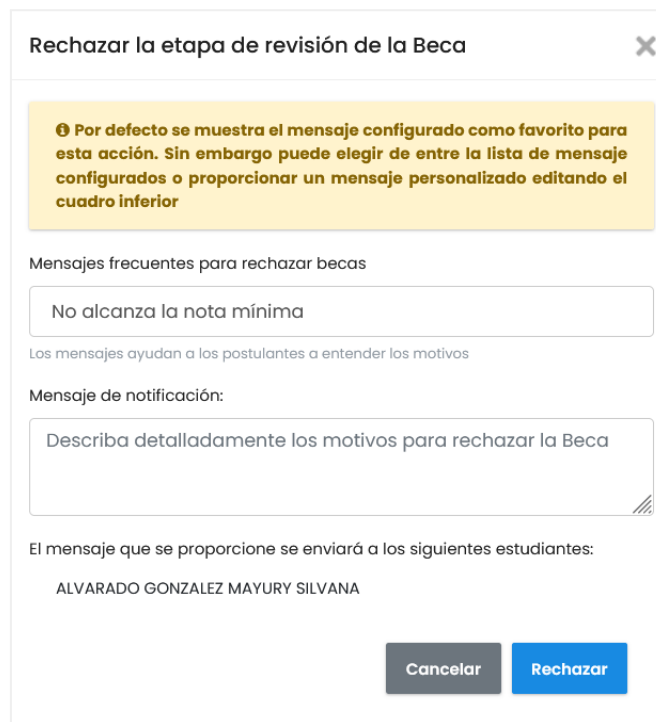
Aprobar

Figura 18: Aprobar la Etapa de Revisión de los Postulantes [Captura de Pantalla]

**Nota:** Se debe asignar una calificación a la ficha socioeconómica a los postulantes para poder aprobar o rechazar. Los postulantes que sean aprobados en la **[Etapa de Revisión]** pasarán a la **[Etapa de Selección]** con el mismo estado.

## 1.4. Rechazar Postulantes

Para Rechazar Postulantes en la **[Etapa de Revisión]**, primero se debe seleccionar a los postulantes que se desea rechazar, luego presionar el botón **[Actualizar Estado]** y elegir la opción **[Rechazar]**; esto desplegará un modal en el que se muestra un cuadro selector de mensajes frecuentes para rechazar; por defecto en el área de texto se muestra el mensaje elegido como favorito para rechazar postulantes, se puede reemplazar el texto con un mensaje más específico si es necesario y se muestran los correos a los que se enviará la notificación al finalizar la **[Etapa de Revisión]**. Se solicita confirmación presionando el botón **[Rechazar]**.



**Figura 19:** Rechazar la Etapa de Revisión de los Postulantes [Captura de Pantalla]

**Nota:** Se debe asignar una calificación a la ficha socioeconómica a los postulantes para poder aprobar o rechazar. Los postulantes que sean aprobados en la **[Etapa de Revisión]** pasarán a la **[Etapa de Selección]** con el mismo estado.

## 1.5. Ver trámite de postulación

Esta acción permite visualizar el trámite del estudiante en el **MGT**. Para esto se debe seleccionar un postulante y presionar el botón de **[Ver Trámite]**, se abrirá una nueva pestaña del navegador con la información del trámite de postulación.

**Listado de Postulaciones – Revisión de la Documentación** Actualizar Estado Descargar documentos Ver Trámite

Facultad: --Todas-- Carrera: --Todas-- Tipo de Beca: --Todas-- Estado: --Todas-- Período Académico: OCT21-ABR22

Mostrar 10 registros

DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Estado
1115272365	carrasco.miguelangel@unl.edu.ec	0962028391	FEAC	PEDAGOGÍA DE LAS CIENCIAS EXPERIMENTALES (QUÍMICA Y BIOLOGÍA)	Beca Tipo B	4	7.87	0	revisando

**Figura 20:** Ver trámite del postulante [Captura de Pantalla]

## 1.6. Generar Reporte

Para generar reportes se puede hacer uso de los filtros [véase figura 4] que se consideren necesarios y luego presionar el botón **[Generar Reporte]**.

**Listado de Postulaciones – Revisión de la Documentación** Actualizar Estado Descargar documentos Ver Trámite

Facultad: --Todas-- Carrera: --Todas-- Tipo de Beca: Beca Tipo A Estado: sin revisión Período Académico: OCT21-ABR22

Mostrar 10 registros

Postulante	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica
<input type="checkbox"/> Polanco Polanco Alejandro Villaseñor	1166253128	roberto.lerma@unl.edu.ec	0996748807	FEAC	EDUCACIÓN INICIAL	Beca Tipo A	7	7.45	0
<input type="checkbox"/> Quintana Quintana Adrián Yáñez	1149859673	paula.henriquez@unl.edu.ec	0958569788	FEAC	EDUCACIÓN INICIAL	Beca Tipo A	7	9.94	0

Mostrando registros del 1 al 10 de un total de 29 registros (filtrado de un total de 96 registros)

Anterior 1 2 3 Siguiente

Estados Disponibles: sin revisar revisar aprobar rechazar Generar Reporte Finalizar Etapa

**Figura 21:** Generar reporte de Postulantes y Becados [Captura de Pantalla]

## 1.7. Finalizar Etapas de Revisión y Selección

Finalizar Etapa permite archivar los registros de postulación para evitar una modificación accidental cuando se desee consultar a futuro. Se pueden finalizar las **[Etapas de Revisión y Selección]**, para ello se debe hacer clic sobre el botón



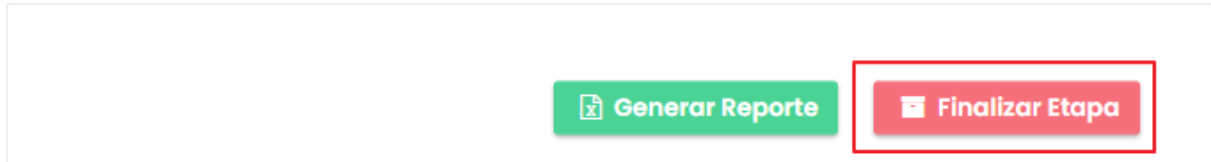
UNL

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

1859 **[Finalizar Etapa]** ubicado en la parte inferior derecha junto al botón Generar Reporte.

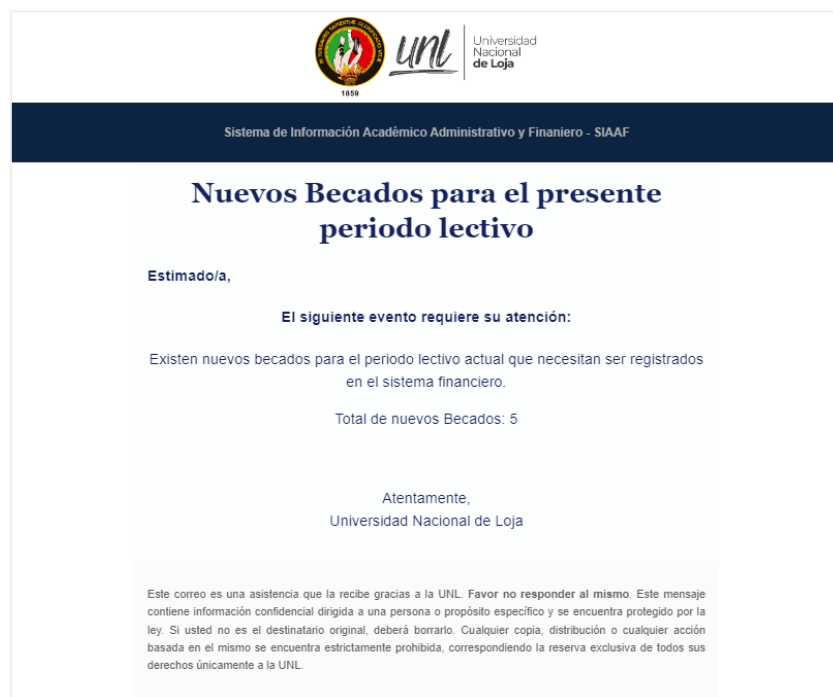
**Nota:** Para poder finalizar una etapa es indispensable que se haya aprobado o rechazado, según sea el caso, a todos los postulantes del periodo académico vigente.



**Figura 22:** Finalizar las Etapas de Revisión y Selección [Captura de Pantalla]

### 3. ETAPA DE NOTIFICACIÓN

La **[Etapa de Adjudicación]** se realiza automáticamente justo después de finalizar la **[Etapa de Selección]**, notificando a Tesorería de la existencia de nuevos Becados a registrar para el periodo en curso; así como también a cada uno de los postulantes con el respectivo mensaje **[Aprobación]** o **[Rechazo]** proporcionado durante las Etapas de **[Revisión]** y **[Selección]**



**Figura 23:** Notificación al correo electrónico de tesorería [Captura de Pantalla]



**Figura 24:** Notificación que recibe un postulante al momento de aprobarse su Beca [Captura de Pantalla]

## 4. ETAPA DE ADJUDICACIÓN

### 4.1. Suspende Becas

La suspensión de Becas se puede realizar en cualquier momento y suspenderá todos los registros de pago pendiente a partir de la suspensión. Esta acción permite cancelar los beneficios de la beca para un estudiante de acuerdo a las consideraciones establecidas en el Reglamento de Becas. Para actualizar el estado del Becado, primeramente se debe seleccionarlo, presione el botón **[Actualizar Estado]** y elija la opción **[Suspender]**. Esta acción notifica al estudiante.

**Nota:** Si por error se suspende a un becado o es necesario reactivar al becado elija la opción Activar, presente en el botón desplegable **[Actualizar Estado]**.



Etapa de Adjudicación Becas > Etapa Adjudicación

### Listado de Becas

Facultad: --Todas-- Carrera: --Todas-- Tipo de Beca: --Todas-- Estado: --Todas-- Período Académico: --Todos--

Mostrar 10 registros Buscar:

Becado	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	
1 <input type="checkbox"/>	Gallegos Gallegos Ismael Soliz	1160390625	psolorio@unl.edu.ec	0941431498	FEIRNR	Ingeniería en Electrónica y Telecomunicaciones	Beca Tipo A	5	9,72	7

2 Actualizar Estado

3 Activar

Suspender

Terminar

Figura 25: Suspender Beca [Captura de Pantalla]

## 4.2. Terminar Becas

La Terminación de Becas se debería realizar al final de cada periodo académico para que al momento de realizar la Sincronización de Becas de Periodos Previos no se creen los registros de pago del siguiente periodo académico. Esta acción permite terminar definitivamente los beneficios de una beca para un estudiante cuando este finaliza sus estudios. Para actualizar el estado del Becado, primeramente seleccione el registro correspondiente y presione el botón **[Actualizar Estado]**, elija la opción **[Terminar]**. Esta acción notifica al estudiante.

Listado de Becas 2 Actualizar Estado

Facultad: --Todas-- Carrera: --Todas-- Tipo de Beca: --Todas-- Estado: --Todas-- Período Académico: --Todos--

Mostrar 10 registros 3 Activar

Buscar:

Becado	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	
1 <input type="checkbox"/>	Gallegos Gallegos Ismael Soliz	1160390625	psolorio@unl.edu.ec	0941431498	FEIRNR	Ingeniería en Electrónica y Telecomunicaciones	Beca Tipo A	5	9,72	7

3 Terminar

Figura 26: Terminar Beca [Captura de Pantalla]

## 4.2. Reactivar Becas

Esta acción permite reactivar los beneficios de una beca para un estudiante cuando por error se cambió el estado a **[Suspendido]** o **[Terminado]**. Para actualizar el estado del Becado, primeramente seleccione el registro correspondiente y presione el botón **[Actualizar Estado]**, elija la opción **[Activar]**. Esta acción notifica al estudiante.



Etapa de Adjudicación

Becas > Etapa Adjudicación

**Listado de Becas** Confirmar Registro Actualizar Estado

Facultad: --Todas-- Programa de estudio: --Todas-- Tipo de Beca: --Todos-- Estado: Suspender Período Académico: OCT21-ABR22

Mostrar 10 registros Buscar: DNI / apellidos / nombres

Becado	Idad	Carrera	Tipo de Beca	Ciclo	Calificación SGA	Calificación Ficha Socioeconómica	Calificación Promedio	Estado	Registro en el ESIGEF
ALULIMA GUAMAN JHULEIDY DELIA	15	DERECHO	Beca Tipo A	7	9.25	8	8.62	SUSPENDIDO	REGISTRADO

Figura 27: Reactivar una Beca [Captura de Pantalla]

## 5. ETAPA DE CONCESIÓN

### 5.1. Generar Reporte Detallado de Concesión

Esta acción genera un reporte con el detalle de estado y valor de pago de cada mes. Para descargar el Reporte Detallado de Concesión aplique los filtros deseados [véase figura 5] y luego haga clic en el botón **[Gestionar Reporte]**.

Universidad Nacional de Loja											
Reporte de Devengado y Pagos de Concesiones											
Tipo de Beca			--Todos--								
Período Académico			ABR22-SEP22								
Estado Concesión			pagado								
Filtro											
Fecha Impresión			2022-08-07 22H:14M:35S								
Total Registros			13								
Nro	Estudiante	DNI	Correo Electrónico	Tipo de Beca	Estado de la Beca	Estados del mes may	Estados del mes jun	Estados del mes jul	Estados del mes ago	Estados del mes sep	Mes pagado
1	BALCAZAR VILLAFUERTE STEFFANNY GISELA	1105390833	steffanny.balcazar@unl.edu.ec	Beca Tipo C	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
2	ACARO PINTA HENRY JAVIER	1105400293	henry.j.acaro@unl.edu.ec	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
3	AGUIRRE CABRERA JAIME DANIEL	1105194383	jaime.aguirre@unl.edu.ec	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
4	AGUIRRE MEDINA REBECA FERNANDA	1150021770	rebeca.f.aguirre@unl.edu.ec	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
5	ANGAMARCA TERAN JORGE ANDRES	1105705444	d1candresangamarca@gmail.com	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
6	ANAZCO GALLARDO JONATHAN ALBERTO	0794878016	jonathan.anazco@unl.edu.ec	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
7	APOLO PINTADO KATHERINE ELIZABETH	11060866042	eli_ap99@outlook.com	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
8	ARCINIEGA ROJAS MISHHELL ALEJANDRA	1150975348	mishell.arcniega@unl.edu.ec	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
9	ARCOS LEON EDISON JAIR	2350094039	edison.arcos@unl.edu.ec	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
10	ARMIJOS ARMIJOS MANUEL FABIAN	1105343642	marmijosarmijos@gmail.com	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
11	AYORA MERINO STEVEN EDUARDO	1150660924	steven.ayora@unl.edu.ec	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
12	CABRERA CUENCA MARI CRUZ	1105853103	mari.cabrera@unl.edu.ec	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45
13	CABRERA CONDOLO JOSE MANUEL	1150838637	jose.m.cabrera.c@unl.edu.ec	Beca Tipo A	Activo	pagado	pagado	pendiente	pendiente	pendiente	45

Figura 28: Reporte de Devengado y Pago de Concesiones [Captura de Pantalla]

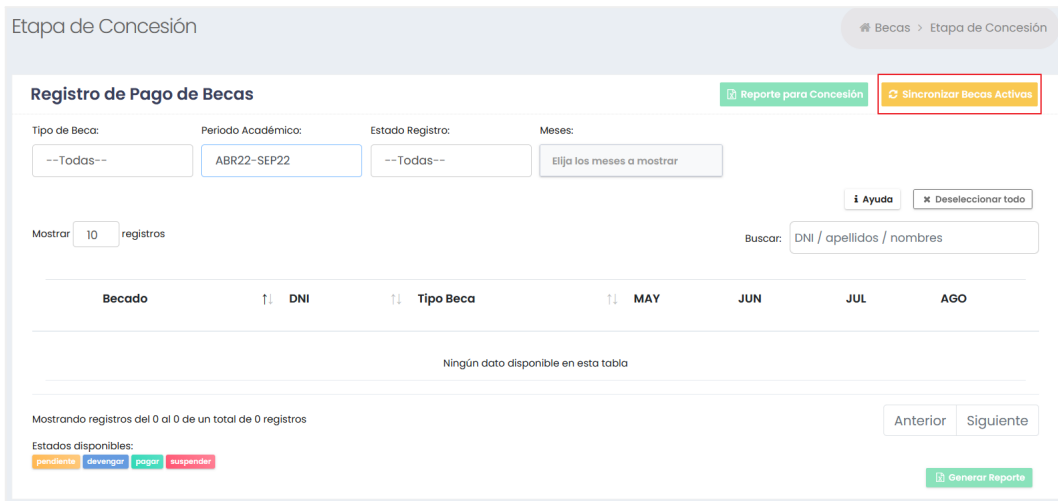
### 5.2. Sincronizar Becas Activas de Periodos Académicos Previos

Esta acción sincroniza las becas activas de los periodos académicos anteriores al periodo actual, creando sus correspondientes registros de pago para cada Beca. Para ellos es importante que el periodo académico haya concluido, que exista un



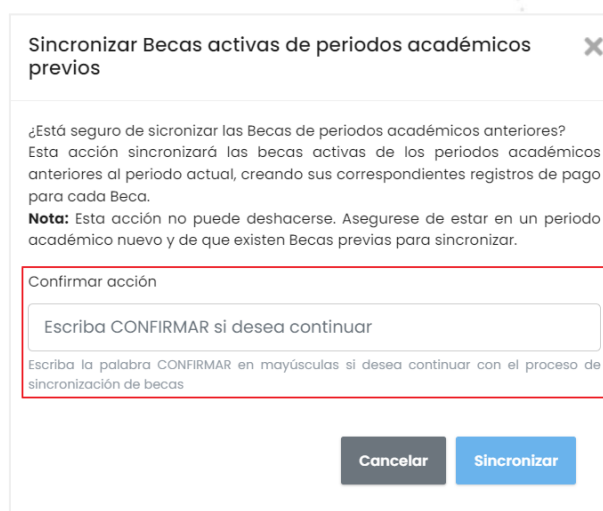


1859 nuevo periodo académico, que se hayan suspendido o terminado las becas que deben llegar a su fin y de ser necesario actualizar el monto de pago para los tipos de beca. Para sincronizar las becas activas de periodos académicos previos, elija de entre los filtros de la **[Etapa de Concesión]**, el periodo académico actual [véase figura 5] y haga clic sobre el botón **[Sincronizar becas activas de periodos académicos previos]**.



**Figura 29:** Botón para sincronizar becas activas de periodos académicos previos [Captura de Pantalla]

Esto abrirá un modal [véase figura 30] en el que se solicita la confirmación para realizar dicha acción ya que no podrá deshacerse y que sirve que se ejecute por error. Para continuar con la acción escriba la palabra **CONFIRMAR** en el recuadro de texto para habilitar el botón y luego presione sobre el botón **[Sincronizar]**. Se mostrará un mensaje de éxito y se actualizará el sitio para volver a cargar la información sincronizada en el nuevo periodo académico.

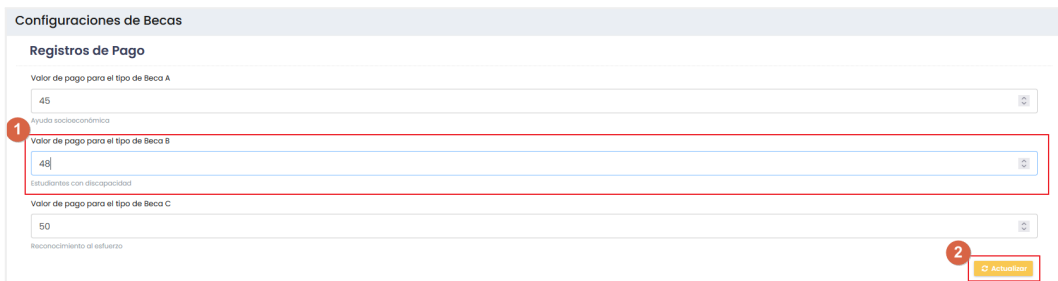


**Figura 30** Ventana modal para la sincronización de Becas activas de periodos académicos previos [captura de pantalla].

## 6. CONFIGURACIONES

### 6.1. Actualizar Montos de Pago para los Tipos de Becas

La actualización de los Montos de Pago se debe realizar sólo si se emite un nuevo reglamento en el que se indica que uno o varios Tipos de Beca cambiaron su monto de concesión. Para actualizar el valor de pago de un tipo de Beca, diríjase desde el menú de Becas al apartado **[Configuraciones]**, desplácese hasta la sección **[Registros de Pago]** ingrese el nuevo valor en el o los tipos de beca que desea actualizar, presione el botón **[Actualizar]** [véase figura 31]. Esto abrirá una ventana modal que le solicitará la confirmación para actualizar el valor de pago, haga clic en el botón **[Aceptar]** para continuar [véase figura 32].



Configuraciones de Becas

Registros de Pago

Valor de pago para el tipo de Beca A  
45

Ayuda socioeconómica

Valor de pago para el tipo de Beca B  
48

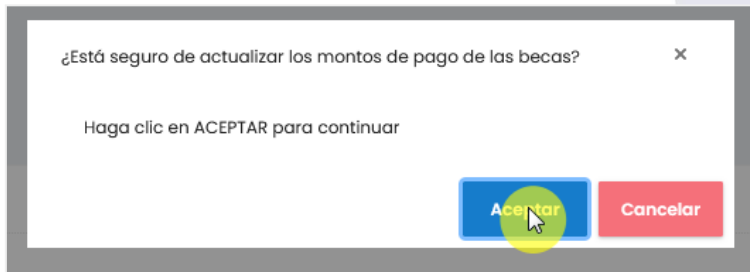
Estudiantes con discapacidad

Valor de pago para el tipo de Beca C  
50

Reconocimiento al esfuerzo

Actualizar

Figura 31 Ingresar un nuevo valor de pago para el Tipo de Beca B [captura de pantalla].



¿Está seguro de actualizar los montos de pago de las becas?

Haga clic en ACEPTAR para continuar

Aceptar Cancelar

Figura 32 Modal para confirmar la actualización del monto de pago [captura de pantalla].

### 6.2. Agregar Mensajes Frecuentes

Estos mensajes se utilizan para agilizar el trabajo de aprobación, rechazo, suspensión o terminación según corresponda de postulantes y becados. Para agregar nuevos mensajes frecuentes, diríjase desde el menú de Becas al apartado **[Configuraciones]**, desplácese hasta la sección **[Mensajes Frecuentes]** y escriba el nuevo mensaje en el **campo de texto** que se indica en el paso 1 [véase figura 33], elija el **tipo de mensaje** que desea agregar, según se indica en el paso 2 de la figura 33, escriba un **orden deseado** en el paso 3, (esto puede ser un valor numérico a partir del 0). Si desea que el mensaje que está agregando se utilice como mensaje frecuente y no tener que escribir o elegir durante la fase de Aprobación, Rechazo, Suspensión o Terminación; marque **Sí** al recuadro que

1859 indica **marcar como favorito** en el paso 4, caso contrario deje sin marcar y presione el botón **[Agregar]**.

Mensaje frecuente

1 Escriba un nuevo mensaje

Estos mensajes ayudarán a comprender el motivo de actualización de estado a los postulantes y becados

2 Tipo de Mensaje

- Aprobar Beca
- Rechazar Beca
- Suspender Beca
- Terminar Beca

Cada mensaje frecuente agregado debe pertenecer a un tipo, elija uno para el mensaje

3 Orden del mensaje

Orden del mensaje

Escriba el orden para el tipo de mensaje

4 Mensaje Favorito

Sí

Marca el mensaje como favorito de entre el mismo tipo al que pertenece

5

Agregar Actualizar Eliminar

**Figura 33:** Agregar un nuevo mensaje frecuente [Captura de Pantalla]

Estos mensajes se utilizan para notificar a los estudiantes del cambio de estado de su postulación o beca según corresponda [véase figura 24].

### 6.3. Actualizar Mensajes Frecuentes

Para actualizar un mensaje frecuente, diríjase desde el menú de becas al apartado **[Configuraciones]**, desplácese hasta la sección **[Mensajes Frecuentes]** y **seleccione desde la tabla el mensaje frecuente** que desea actualizar como se muestra en el paso 1 de la [véase figura 34]. Una vez seleccionado, el cuadro de texto resaltará en amarillo para indicarle que se ha cargado la información correspondiente al mensaje seleccionado y que **puede editarse** según se indica en el paso 2 de la [véase figura 34], tal como se hace para agregar un nuevo mensaje frecuente, solo que esta vez se debe presionar el botón **[Actualizar]**.

Orden	Mensaje Frecuente	Tipo
<input type="checkbox"/>	Bienvenido a la comunidad de Becados	APROBAR
<input type="checkbox"/>	No cumple con los requisitos mínimos	RECHAZAR
<input type="checkbox"/>	Se encontraba libando en las instalaciones de la institución	SUSPENDER
<input checked="" type="checkbox"/>	Su beca ha concluido a partir de ahora ya no recibirá beneficios económicos	TERMINAR

Mostrando registros del 1 al 4 de un total de 4 registros Fila seleccionada

Anterior 1 Siguiente

Mensaje frecuente

Su beca ha concluido a partir de ahora ya no recibirá beneficios económicos

Estos mensajes ayudarán a comprender el motivo de actualización de estado a los postulantes y becados

Tipo de Mensaje

- Aprobar Beca
- Rechazar Beca
- Suspender Beca
- Terminar Beca

Cada mensaje frecuente agregado debe pertenecer a un tipo, elija uno para el mensaje

Orden del mensaje

1

Escriba el orden para el tipo de mensaje

Mensaje Favorito

Sí

Marca el mensaje como favorito de entre el mismo tipo al que pertenece

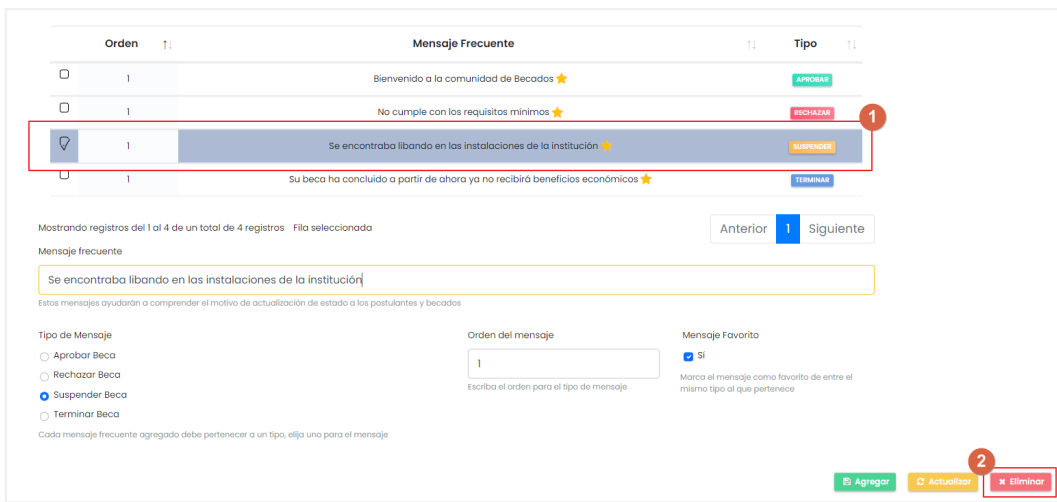
3

Agregar Actualizar Eliminar

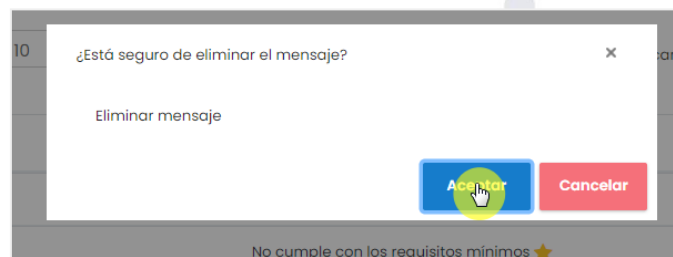
**Figura 34:** Actualizar mensaje usado para la terminación de Becas [Captura de Pantalla]

## 6.4. Eliminar Mensajes Frecuentes

Para eliminar un mensaje frecuente, diríjase desde el menú de becas al apartado **[Configuraciones]**, desplácese hasta la sección **[Mensajes Frecuentes]** y **seleccione desde la tabla el mensaje frecuente** que desea eliminar como se muestra en el paso 1 de la [véase figura 35]. Una vez seleccionado, el cuadro de texto resaltará en amarillo para indicarle que se ha cargado la información correspondiente al mensaje seleccionado, presione el botón **[Eliminar]**. Se abrirá un modal [véase figura 36] solicitando la confirmación para eliminar el mensaje seleccionado.



**Figura 35:** Eliminar mensaje usado para suspender Becas [Captura de Pantalla]



**Figura 36:** Modal de confirmación al presionar sobre el botón Eliminar [Captura de Pantalla]



**unl**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

# CAPÍTULO III

# ADJUDICACIÓN Y

# CONCESIÓN DE

# BECAS

# DIRECCIÓN

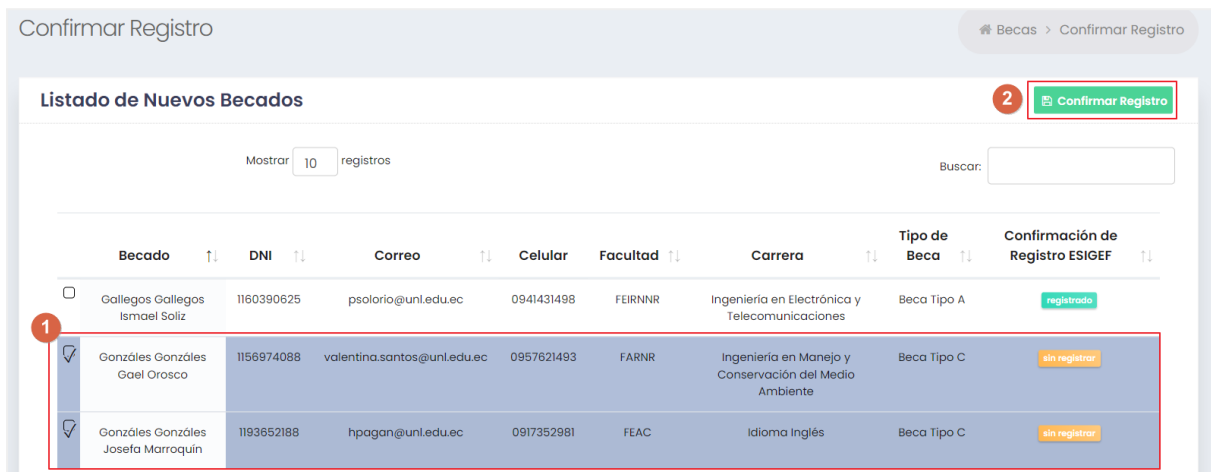
# FINANCIERA

## 1. ETAPA DE ADJUDICACIÓN

### 1.1. Confirmar Registro de Becado en el Sistema Financiero

Esta acción permite guardar en el **MGB**, la confirmación de registro de los becados en el Sistema Financiero. **Esta acción es muy importante ya que crea los registros de pago para la concesión de las becas.**

Para confirmar el registro de las Becas en el sistema Financiero, diríjase a la **[Etapa de Adjudicación]** desde el Menú de Becas, seleccione todos los becados que ya se hayan registrado, y presione el botón **[Confirmar Registro]**.



The screenshot shows a web interface titled 'Confirmar Registro'. At the top right, there is a breadcrumb 'Becas > Confirmar Registro'. Below the title, there is a 'Listado de Nuevos Becados' section with a 'Confirmar Registro' button (marked with a red '2'). A search bar and a 'Mostrar 10 registros' dropdown are also present. The main content is a table with columns: 'Becado', 'DNI', 'Correo', 'Celular', 'Facultad', 'Carrera', 'Tipo de Beca', and 'Confirmación de Registro ESIGEF'. Three rows are visible, with the second and third rows highlighted in blue and having a 'sin registrar' button. The first row has a 'registrado' button. A red box highlights the first two rows, and a red circle with '1' is next to the first row's checkbox.

Becado	DNI	Correo	Celular	Facultad	Carrera	Tipo de Beca	Confirmación de Registro ESIGEF
<input type="checkbox"/> Gallegos Gallegos Ismael Soliz	1160390625	psolorio@unl.edu.ec	0941431498	FEIRNR	Ingeniería en Electrónica y Telecomunicaciones	Beca Tipo A	registrado
<input checked="" type="checkbox"/> Gonzáles Gonzáles Gael OroSCO	1156974088	valentina.santos@unl.edu.ec	0957621493	FARNR	Ingeniería en Manejo y Conservación del Medio Ambiente	Beca Tipo C	sin registrar
<input checked="" type="checkbox"/> Gonzáles Gonzáles Josefa Marroquín	1193652188	hpagan@unl.edu.ec	0917352981	FEAC	Idioma Inglés	Beca Tipo C	sin registrar

Figura 37: Guardar confirmación de registro de los becados [Captura de Pantalla]

### 1.2. Armar Reporte de Importación de Pago

Para armar un Reporte de Importación de Pago es necesario tener el [Reporte para Concesión](#) en el que se debe completar la columna **NRO\_EXPEDIENTE** (en caso que los registros a importar no tengan el estado previo de DEVENGADO) del archivo excel [véase Figura 38], y opcionalmente la columna **FECHA\_TRANSACCION** (cuando la fecha de transacción de PAGO seá distinta a la fecha en la que se va a importar el archivo) siguiendo la estructura: **año-mes-día hora:minutos:segundos**.



Universidad Nacional de Loja					
Reporte para importar registros de devengado o pago de concesiones					
Tipo de Beca	Beca Tipo A				
Periodo Académico	ABR21-SEP21				
Estado Concesión	pendiente				
Filtro					
Fecha Impresión	2022-08-04 18H:17M:01S				
Total Registros	14				
NRO_IDENTIFICACION	BECADO	NRO_EXPEDIENTE	MESES	CREDITO	FECHA_TRANSACCION
1150975348	ARCINIEGA ROJAS MISHHELL ALEJANDRA		may	45	
0704878016	AÑAZCO GALLARDO JONATHAN ALBERTO		may	45	
2350094039	ARCOS LEON EDISON JAIR		may	45	
1105705444	ANGAMARCA TERAN JORGE ANDRES		may	45	
1105194383	AGUIRRE CABRERA JAIME DANIEL		may	45	
1105400293	ACARO PINTA HENRY JAVIER		may	45	
1150838637	CABRERA CONDOLO JOSE MANUEL		may	45	
1106086042	APOLO PINTADO KATHERINE ELIZABETH		may	45	
1105853103	CABRERA CUENCA MARI CRUZ		may	45	
1104818339	BECERRA MONTALVAN SANTIAGO FERNANDO		may	45	
1105343642	ARMIJOS ARMIJOS MANUEL FABIAN		may	45	
1106059775	CAILLAGUA MONTAÑO ANGHELA LEONELA		may	45	
1150660924	AYORA MERINO STEVEN EDUARDO		may	45	
1150021770	AGUIRRE MEDINA REBECA FERNANDA		may	45	

Figura 38 Reporte para importar registros de devengado o pago de concesiones.

**Nota:** Cuando se realice la importación del archivo, el módulo revisará si la columna **FECHA\_TRANSACCION** tiene una fecha válida y la utilizará, caso contrario si este campo no es provisto, se completará con la fecha y hora actual. Antes de completar el campo **FECHA\_TRANSACCION** asegúrese de haber seleccionado el **formato Texto** para todas las celdas de la columna que contengan información de registros.

## 2. ETAPA DE CONCESIÓN

### 2.1. Importar Registros de Devengado o Pago

Esta acción permite importar registros de devengado o pago según sea el caso, desde el archivo excel descargado en la [sección 1.8](#). Una vez preparado el archivo a importar [véase [sección 1.2](#)], diríjase a la **[Etapa de Concesión]** desde el Menú de Becas y presione el botón **[Importar Registros]** como se muestra en la figura 39.

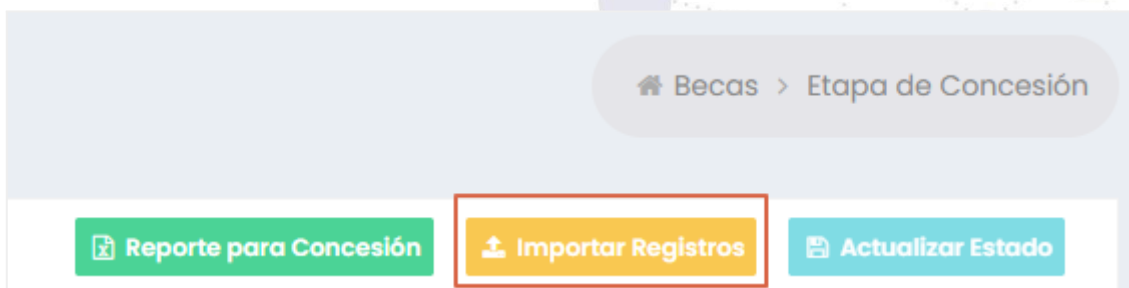


Figura 39: Botón para Importar Registros de Devengado o Pago [Captura de Pantalla]

Esta acción redirige a una vista para importar el archivo, primero se debe seleccionar el archivo presionando en el botón **[Explorar]**, esto abrirá el explorador





El reporte de errores agrega una columna adicional denominada **ERRORES** en la que se detallan los errores detectados durante la importación.

Universidad Nacional de Loja						
Reporte con errores durante el proceso de importación						
Periodo Académico		OCT20-ABR21				
Fecha Impresión		2022-08-04 19H:59M:08S				
Total Registros		4				
NRO_IDENTIFICACION	BECADO	NRO_EXPEDIENTE	MESES	CREDITO	FECHA_TRANSACCION	ERRORES
1150669924	AYORA MERINO STEVEN EDUARDO		nov	45		Expediente Invalido: debe tener 23 caracteres
1150021770	AGUIRRE MEDINA REBECA FERNANDA		nov	45		Expediente Invalido: debe tener 23 caracteres
1106086042	APOLO PINTADO KATHERINE ELIZABETH		nov	45		Expediente Invalido: debe tener 23 caracteres
1105400293	JACARO PINTA HENRY JAVIER	1700000000000098	nov	45		Registro tiene un pago previo que se encuentra en el mismo estado que el registro a importar

**Figura 42:** Reporte con el detalle de errores presentados durante la importación [Captura de Pantalla]

La validación del archivo de importación se realiza en el siguiente orden:

1. Registros que no contienen los campos requeridos: **NRO\_IDENTIFICACION**, **NRO\_EXPEDIENTE**, **MESES**, **CRÉDITO**.
2. Existencia de por lo menos un **MES**; **CRÉDITO** no negativo o con caracteres inválidos, **NRO\_IDENTIFICACION** que sea una Cédula o RUC válido.
3. El módulo valida que no existan registros duplicados; es decir, registros que contengan el mismo mes de pago para el mismo beneficiario.
4. El módulo valida que se exista el registro de pago correspondiente para cada mes a importar.
5. El módulo valida la correlación entre el valor de asignación de pago y la cantidad de meses a registrar. Ej: para los meses Nov, Dic → su valor debe ser de 90\$ si por cada mes le corresponden 45\$.
6. El módulo valida que no exista un registro de pago previo en el mismo estado al que se intenta actualizar. Ej: registro a importar Nov 45\$ periodo OCT21-ABR22, registro pago previo Nov 45\$ periodo OCT21-ABR22.

**Nota:** Cuando se importa un **DEVENGADO** el campo **NRO\_EXPEDIENTE** es requerido, a diferencia de cuando se importan registros de **PAGO** que deja de ser requerido siempre y cuando el registro tenga un registro previo en estado **DEVENGADO**, ya que el campo **NRO\_EXPEDIENTE** ya fue registrado durante la importación de **DEVENGADO**.

## 2.2. Actualización Manual de Registros de Concesión

Esta acción permite actualizar manualmente los registros de devengado o pago según sea el caso, seleccionando los registros deseados directamente desde la tabla. Para actualizar manualmente los registros de devengado o pago, diríjase a la **[Etapa de Concesión]** desde el Menú de Becas seleccione los registros que desea actualizar y presione el botón **[Actualizar Estado]** [véase figura 43]. Se abrirá un modal en el que se puede elegir los meses que se desea actualizar (paso 1), el estado al que se va a actualizar (paso 2) e ingresar el número de expediente correspondiente al registro (paso 3) finalmente haga clic en el botón **[Aceptar]** para confirmar [véase figura 44].

**Registro de Pago de Becas** Importar Registros Actualizar Estado

Tipo de Beca: --Todas-- Período Académico: OCT20-ABR21 Estado Registro: --Todas-- Meses: Elija los meses a mostrar

Mostrar 10 registros Ayuda Deseleccionar todo

Buscar: DNI / apellidos / nombres

Becado	DNI	Tipo Beca	OCT	NOV	DIC	ENE	FEB
<input type="checkbox"/> ACARO PINTA HENRY JAVIER	1105400293	Beca Tipo A	2022-08-04 DEVENGADO \$ 45	2022-08-04 DEVENGADO \$ 45	2022-08-04 DEVENGADO \$ 45	2022-08-04 DEVENGADO \$ 45	2022-08-04 DEVENGADO \$ 45
<input type="checkbox"/> AGUIRRE CABRERA JAIME DANIEL	1105194383	Beca Tipo A	2022-08-04 DEVENGADO \$ 45	2022-08-04 DEVENGADO \$ 45	2022-08-03 PENDIENTE \$ 45	2022-08-03 PENDIENTE \$ 45	2022-08-03 PENDIENTE \$ 45
<input checked="" type="checkbox"/> AGUIRRE MEDINA REBECA FERNANDA	1150021770	Beca Tipo A	2022-08-04 DEVENGADO \$ 45	2022-08-03 PENDIENTE \$ 45	2022-08-03 PENDIENTE \$ 45	2022-08-03 PENDIENTE \$ 45	2022-08-03 PENDIENTE \$ 45

**Figura 43:** Actualizar estado de la concesión manualmente [Captura de Pantalla]

**Actualizar el estado de la Concesión de Beca**

Se actualizará el estado de la concesión de los registros seleccionados.

Seleccione los meses que desea actualizar:

OCT  
 NOV  
 DIC  
 ENE  
 FEB

Seleccione el estado de la concesión:

pendiente

Ingrese el nro de expediente:

Nro. Expediente

Solo se puede completar este campo cuando se elige a un becado

Cancelar Aceptar

**Figura 44:** Elegir detalles de la actualización manual de la concesión [Captura de Pantalla]

## 2.3. Armar Reporte de Importación de Devengado

Para armar un Reporte de Importación de Devengado debe completar la columna **NRO\_EXPEDIENTE** del archivo excel [véase figura 32], y opcionalmente la columna **FECHA\_TRANSACCION** (cuando la fecha de transacción de devengado sea distinta a la fecha en la que se va a importar el archivo) siguiendo la estructura: **año-mes-día hora:minutos:segundos**.

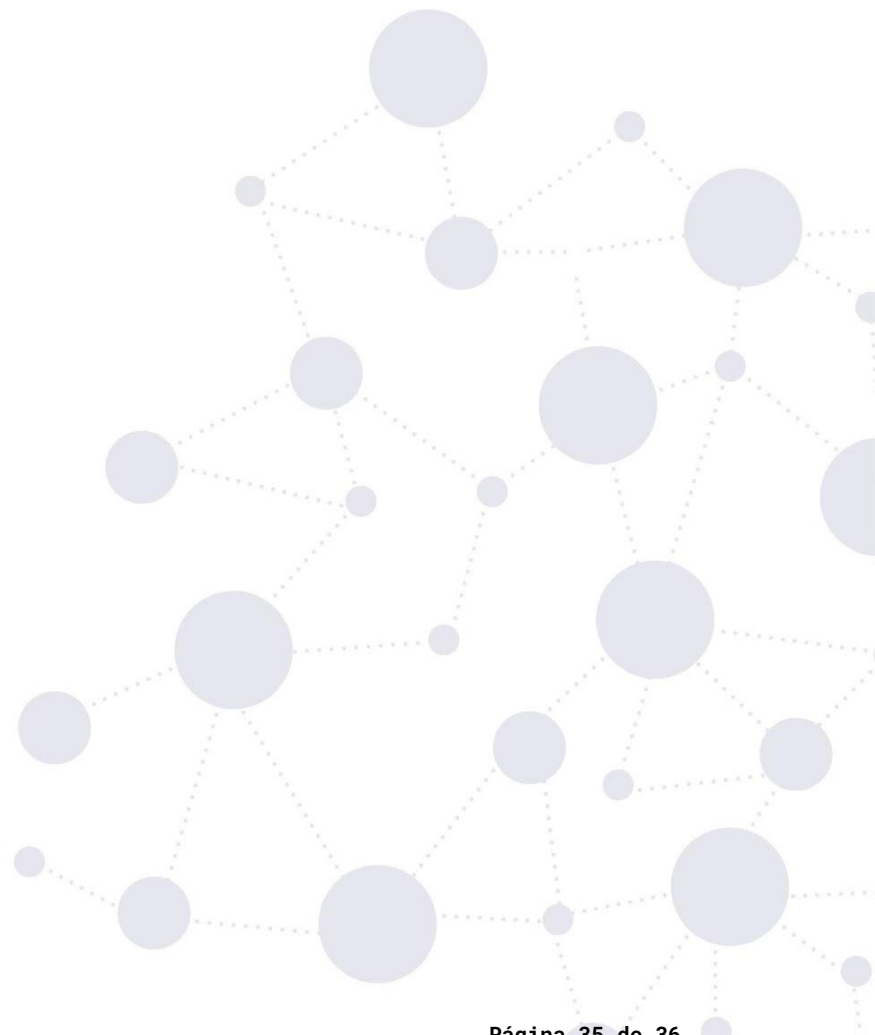


1859

Universidad Nacional de Loja					
Reporte para importar registros de devengado o pago de concesiones					
7	<b>Tipo de Beca</b>		Beca Tipo A		
8	<b>Periodo Académico</b>		ABR21-SEP21		
9	<b>Estado Concesión</b>		pendiente		
10	<b>Filtro</b>				
11	<b>Fecha Impresión</b>		2022-08-04 18H:17M:01S		
12	<b>Total Registros</b>		14		
NRO_IDENTIFICACION	BECADO	NRO_EXPEDIENTE	MESES	CREDITO	FECHA_TRANSACCION
1150975348	ARCINIEGA ROJAS MISHHELL ALEJANDRA		may	45	
0704878016	AÑAZCO GALLARDO JONATHAN ALBERTO		may	45	
2350094039	ARCOS LEON EDISON JAIR		may	45	
1105705444	ANGAMARCA TERAN JORGE ANDRES		may	45	
1105194383	AGUIRRE CABRERA JAIME DANIEL		may	45	
1105400293	ACARO PINTA HENRY JAVIER		may	45	
1150838637	CABRERA CONDOLO JOSE MANUEL		may	45	
1106086042	APOLO PINTADO KATHERINE ELIZABETH		may	45	
1105853103	CABRERA CUENCA MARI CRUZ		may	45	
1104818339	BECERRA MONTALVAN SANTIAGO FERNANDO		may	45	
1105343642	ARMIJOS ARMIJOS MANUEL FABIAN		may	45	
1106059775	CAILLAGUA MONTAÑO ANGHELA LEONELA		may	45	
1150660924	AYORA MERINO STEVEN EDUARDO		may	45	
1150021770	AGUIRRE MEDINA REBECA FERNANDA		may	45	

Figura 45 Reporte para importar registros de devengado o pago de concesiones.

**Nota:** Cuando se realice la importación del archivo, el módulo revisará si la columna **FECHA\_TRANSACCION** tiene una fecha válida y la utilizará, caso contrario si este campo no es provisto, se completará con la fecha y hora actual.





## RECOMENDACIONES:

- Antes de finalizar las **[Etapas de Revisión y Selección]**, se recomienda revisar que no existan inconsistencias como fichas socioeconómicas con una calificación que no corresponde o postulantes en estado de revisión o sin revisar.
- La Subdirección de Tesorería será la responsable de registrar a los nuevos becados en el sistema financiero y guardar la confirmación de registro en el **MGB** ya que esto habilita a los nuevos becados a que puedan recibir los beneficios económicos de su beca.
- La Subdirección de Tesorería será responsable de registrar los devengados y los pagos, para llevar el control de la concesión al día.
- La Subdirección de Contabilidad debe proporcionar a Tesorería el reporte con los registros devengados para que pueda realizar la importación de los mismos.

## FIRMAS DE RESPONSABILIDAD

Acción	Funcionario	Firmas
Elaborado por:	Jackson Andrews Guzmán Tituana <b>Tesista de la Carrera de Ingeniería en Sistemas</b>	 Digitally signed by Jackson Guzmán Date: 2022.09.02 15:40:54 -05'00'
Revisado por:	Danny Emanuel Muñoz Flores <b>Especialista de Sistemas de información</b>	 Firmado electrónicamente por: <b>DANNY EMANUEL MUNOZ FLORES</b>
	Johana Elizabeth Cueva Domínguez <b>Analista de Sistemas Informáticos I</b>	 Firmado electrónicamente por: <b>JOHANA ELIZABETH CUEVA DOMINGUEZ</b>
Aprobado por:	Jhon Alexander Calderón Sanmartín <b>Director de Tecnologías de Información</b>	 Firmado electrónicamente por: <b>JHON ALEXANDER CALDERON SANMARTIN</b>

## **Anexo 5. Manual Técnico de Requisitos Básicos para la configuración del MGB**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Manual Técnico de Implantación<sup>7</sup>

---

<sup>7</sup> Versión original en: <https://drive.google.com/Manual-Tecnico-MGB.pdf>



**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

[MANUAL TÉCNICO]

<<DIRECCIÓN DE TECNOLOGÍAS DE INFORMACIÓN>>

# MÓDULO DE GESTIÓN DE BECAS - SIAAF

Versión 22.08

**Elaborado por:**

*Jackson Andrews Guzmán Tituana*

**Revisado por:**

*Danny Emanuel Muñoz Flores*

**Aprobado por:**

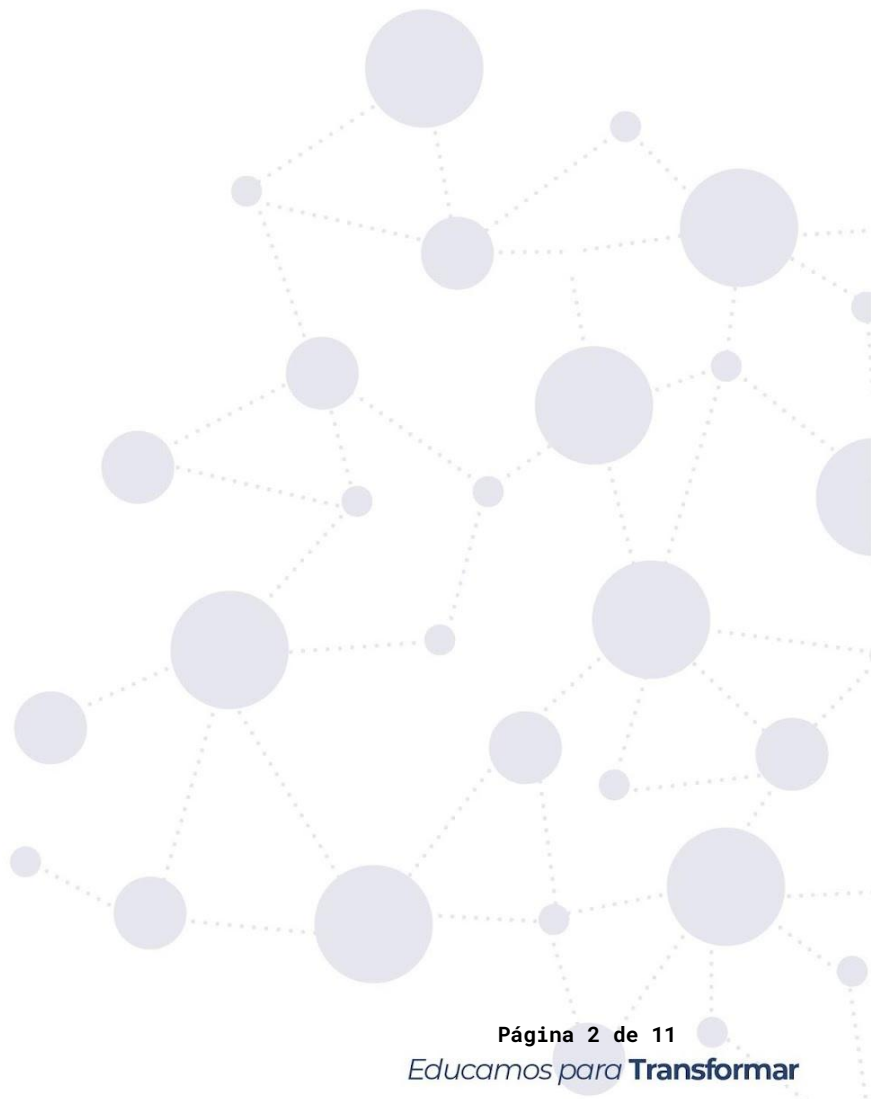
*Jhon Alexander Calderón Sanmartín*

Agosto 2022



## HISTORIAL DE CAMBIOS

<b>Versión</b>	<b>Fecha</b>	<b>Responsable</b>	<b>Descripción de la modificación</b>
22.08	29/08/2022	Tesista Jackson Guzmán	Documento Inicial





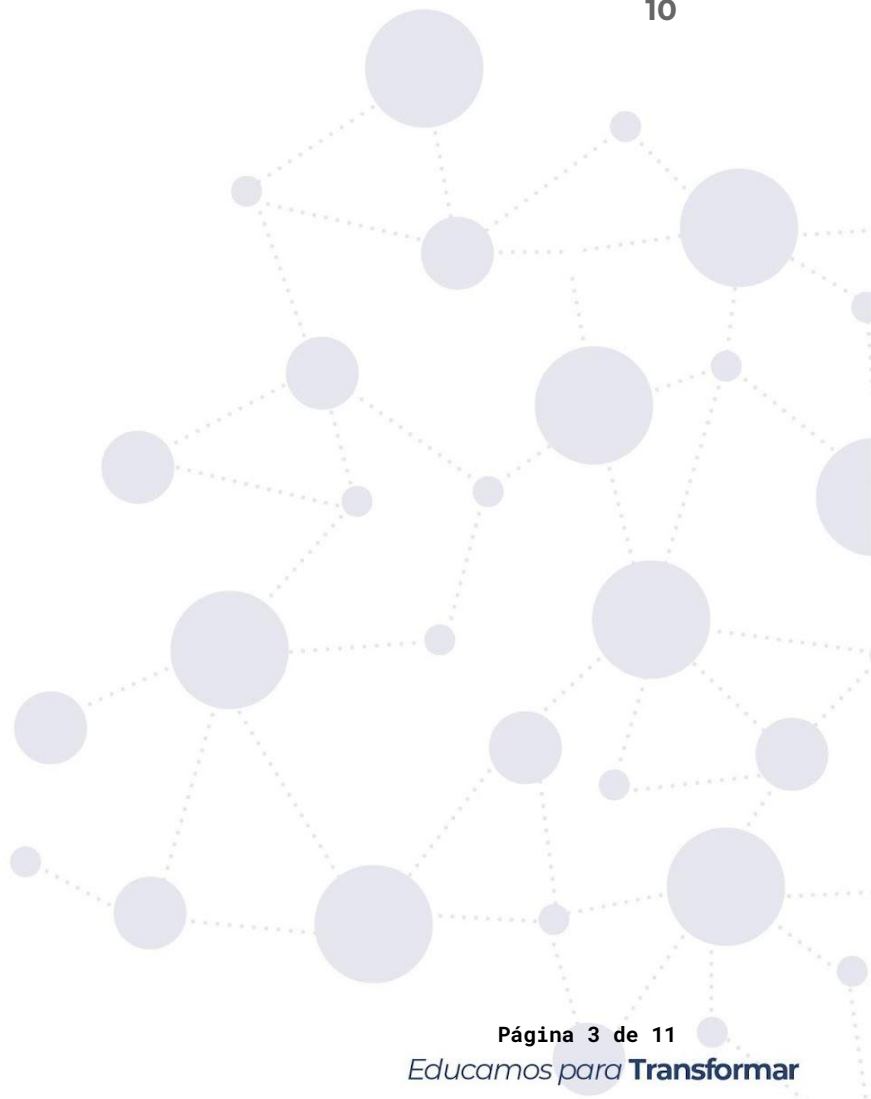
**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

## ÍNDICE DE CONTENIDOS

<b>INTRODUCCIÓN</b>	<b>4</b>
<b>OBJETIVO</b>	<b>4</b>
<b>DIRIGIDO A</b>	<b>4</b>
<b>CAPÍTULO I</b>	<b>5</b>
<b>1. GENERALIDADES</b>	<b>6</b>
1.1. Diagrama General de Procesos	6
<b>CAPÍTULO II</b>	<b>7</b>
<b>1. CONFIGURACIÓN DE BASE DE DATOS</b>	<b>8</b>
1.1. MIGRAR MODELOS DE LA APP A BASE DE DATOS	8
1.2. INSERTS A BASE DE DATOS	8
Registros del Modelo Parametrización y DetalleParametrización	8
Registros del Modelo Etapas	8
Registros del Modelo AuthPermission	8
Registros del Modelo TipoBeca	9
1.3. ADMINISTRAR CICLO MÁXIMO DE BECAS	9
1.4. ASIGNACIÓN DE ROLES DE USUARIO	9
<b>FIRMAS DE RESPONSABILIDAD</b>	<b>10</b>







UNL

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

## INTRODUCCIÓN

El Sistema de Información Académico, Administrativo y Financiero (SIAAF) es una plataforma que ha permitido automatizar diversos procesos de la Universidad Nacional de Loja (UNL), de manera integrada, flexible y estandarizada. De forma paulatina se han ido incorporando nuevas funcionalidades, ello con el fin de implementar mejoras que permitan optimizar procesos en los diferentes ámbitos como: académico, administrativo y financiero.

En cuanto al ámbito académico con el fin de brindar al personal involucrado en la Gestión de Becas, se ha incorporado el Módulo de Gestión de Becas (MGB), el cual se enfoca en coadyuvar en los procesos de seguimiento y control de becas. Así también a futuro se pretende mejorar la comunicación existente con los estudiantes en cuanto a los beneficios recibidos por las Becas.

El presente manual está dividido en capítulos en donde se explican las acciones que se deben realizar para configurar el MGB.

## OBJETIVO

Guiar en los procesos a seguir para la configuración del **Módulo de Gestión de Becas (MGB)** en el ambiente de producción del SIAAF.

## DIRIGIDO A

Este documento es una guía al personal técnico de DTI con responsabilidades como: administradores del SIAAF de la Universidad Nacional de Loja.



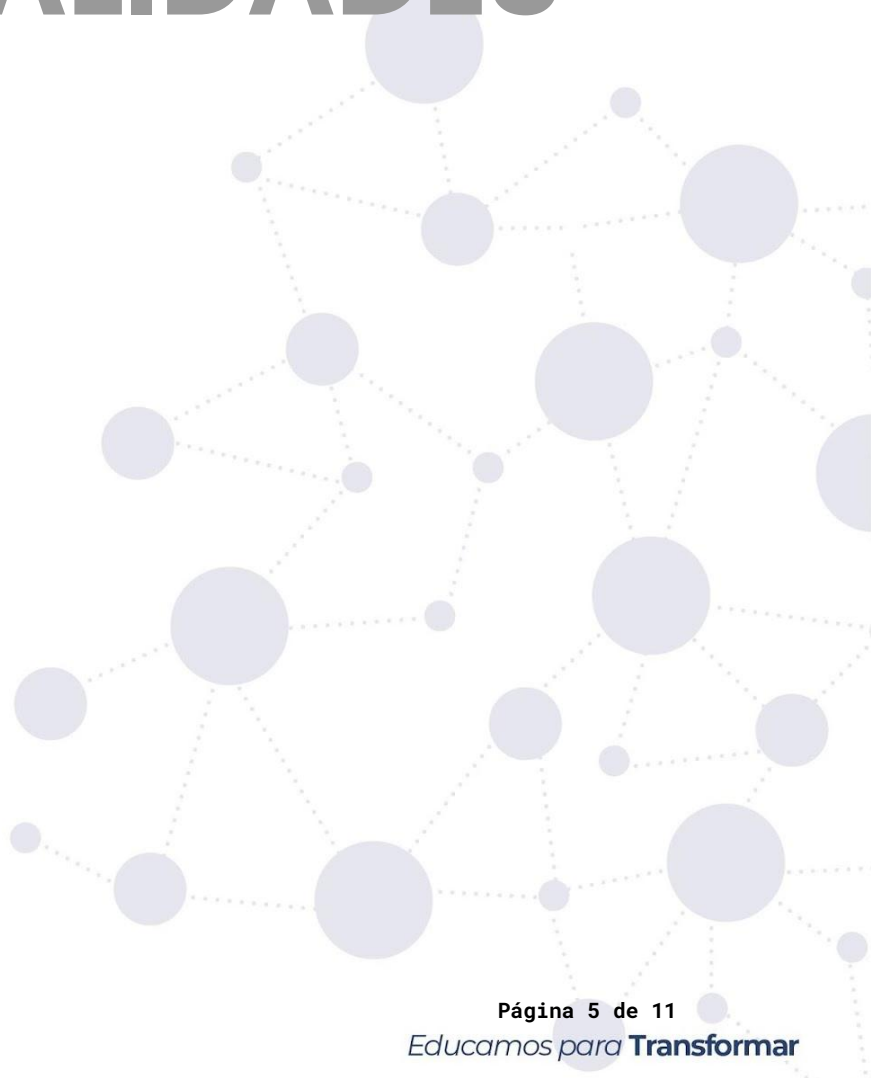
**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

# CAPÍTULO I

# GENERALIDADES



## 1. GENERALIDADES

### 1.1. Diagrama General de Procesos

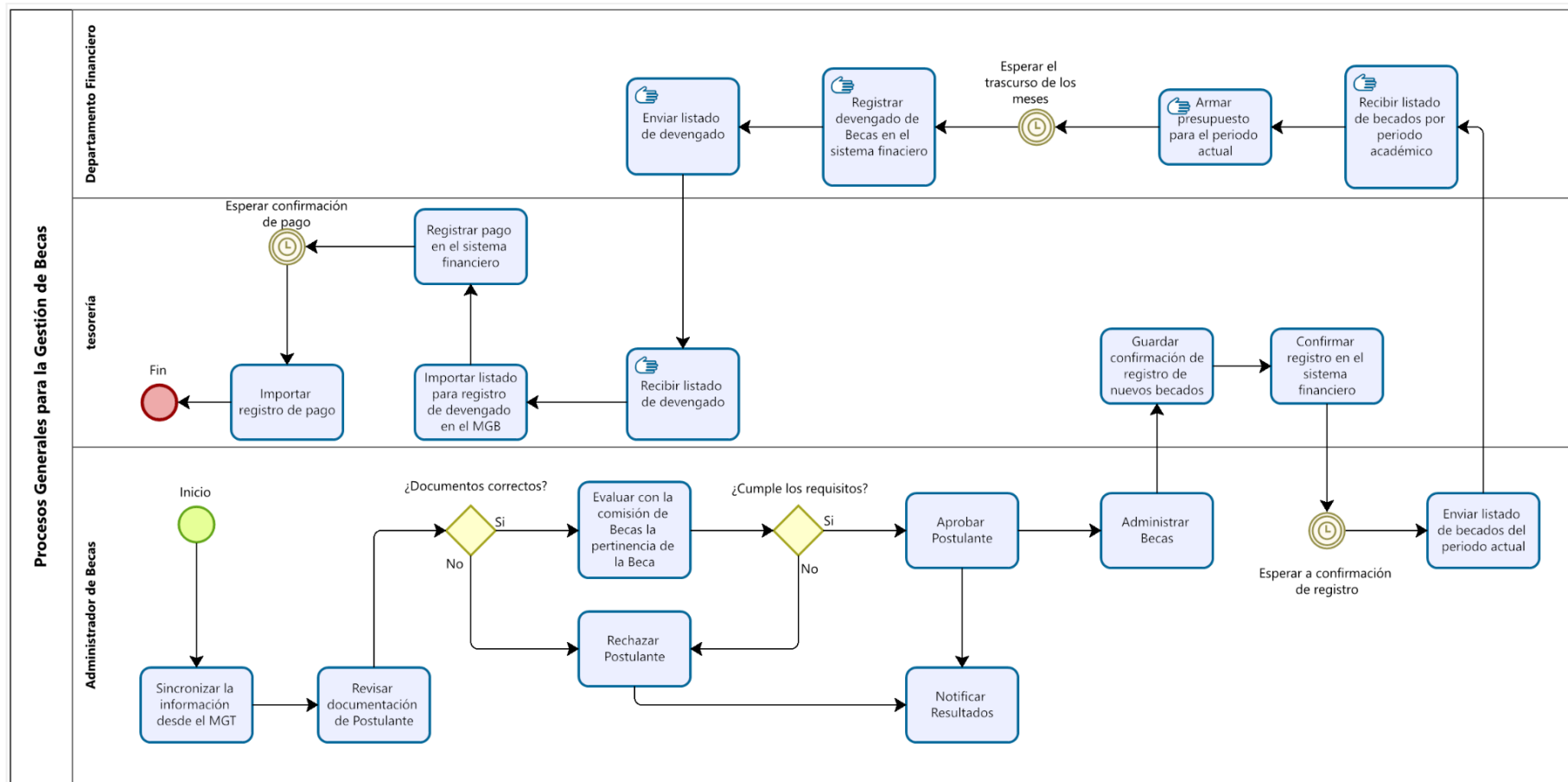


Figura 1.: Diagrama general de procesos para la Gestión de Becas.



**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

# CAPÍTULO II

# CONFIGURACIONES GENERALES DEL MÓDULO DE GESTIÓN DE BECAS

## 1. CONFIGURACIÓN DE BASE DE DATOS

### 1.1. MIGRAR MODELOS DE LA APP A BASE DE DATOS

El primer paso a realizarse en la configuración de la aplicación es migrar los modelos de la aplicación a la base de datos esto se puede realizar con el siguiente comando:

```
python manage.py makemigrations becas  
python manage.py migrate becas
```

**Figura 2.** Comandos de Django para ejecutar las migraciones de los modelos a la Base de Datos [comandos]

**Nota:** asegúrese de encontrarse dentro del directorio ~/siaaf

### 1.2. INSERTS A BASE DE DATOS

Para configurar el MGB se requiere insertar ciertos datos por defecto para que el módulo funcione correctamente.

#### Registros del Modelo Parametrización y DetalleParametrización

Archivo SQL: [enlace a archivo](#)

Dentro de este archivo sql se encuentra el registro de parametrización para las parametrizaciones del MGB y 2 registros detalleparametrización.

El registro con código EMAIL\_TESORERIA hace referencia al correo electrónico institucional del usuario de tesorería para que pueda recibir las notificaciones de nuevos becados adjudicados.

El registro con código URL\_MGT\_DETALLE\_PROCESO contiene parte de la ruta hacia un trámite del MGT.

El registro con código NOTA\_MINIMA\_APROBAR\_BECA se refiere a la nota mínima que el MGB considera una vez se finaliza la **[Etapa de Revisión]**.



## Registros del Modelo Etapas

Archivo SQL: [enlace a archivo](#)

Estos registros pertenecen a un modelo propio del módulo y es importante que se encuentren registrados en la base de datos para que las vistas de cada etapa funcionen correctamente.

## Registros del Modelo AuthPermission

Archivo SQL: [enlace a archivo](#)

Dentro del archivo se encuentran todos los permisos necesarios para cada grupo de usuario que pertenece al MGB. Estos permisos son importantes para el correcto funcionamiento del MGB. Luego de haber realizado los **INSERT** de este archivo se debe ejecutar el comando `python manage.py migrate becas` para que se asocien los permisos al grupo de usuarios correspondiente.

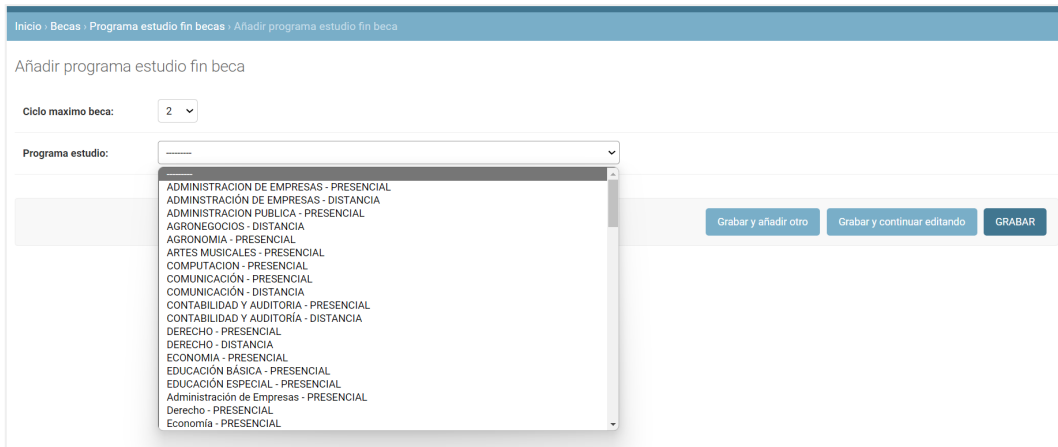
## Registros del Modelo TipoBeca

Archivo SQL: [enlace a archivo](#)

El archivo contiene los 3 tipos de becas vigentes hasta el momento, es importante completar con los montos de pago reales.

## 2. Configuraciones adicionales

### 2.1. ADMINISTRAR CICLO MÁXIMO DE BECAS



**Figura 3.** Vista del Administrador del SIAAF para agregar nuevos registros de programa\_estudio\_fin\_beca [captura de pantalla]

Este modelo se compone de 2 campos, el campo ciclo máximo beca hace referencia al ciclo en que las becas caducan, es decir, el ciclo máximo de gracia de las becas de acuerdo al programa de estudio al que pertenecen, por otro lado el campo programa estudio hace referencia a la carrera para la cual se quiere configurar el ciclo máximo de gracia para las becas de ese programa estudio.

### 2.2. ASIGNACIÓN DE ROLES DE USUARIO

Existen 3 roles para el uso del MGB y están distribuidos en 3 grupos distintos con permisos específicos para cada rol.

Los grupos creados para el módulo informático son:

- Administrador de Becas (administrador\_becas)
- Contabilidad Becas (contabilidad\_becas)
- Tesorería Becas (tesoreria\_becas)

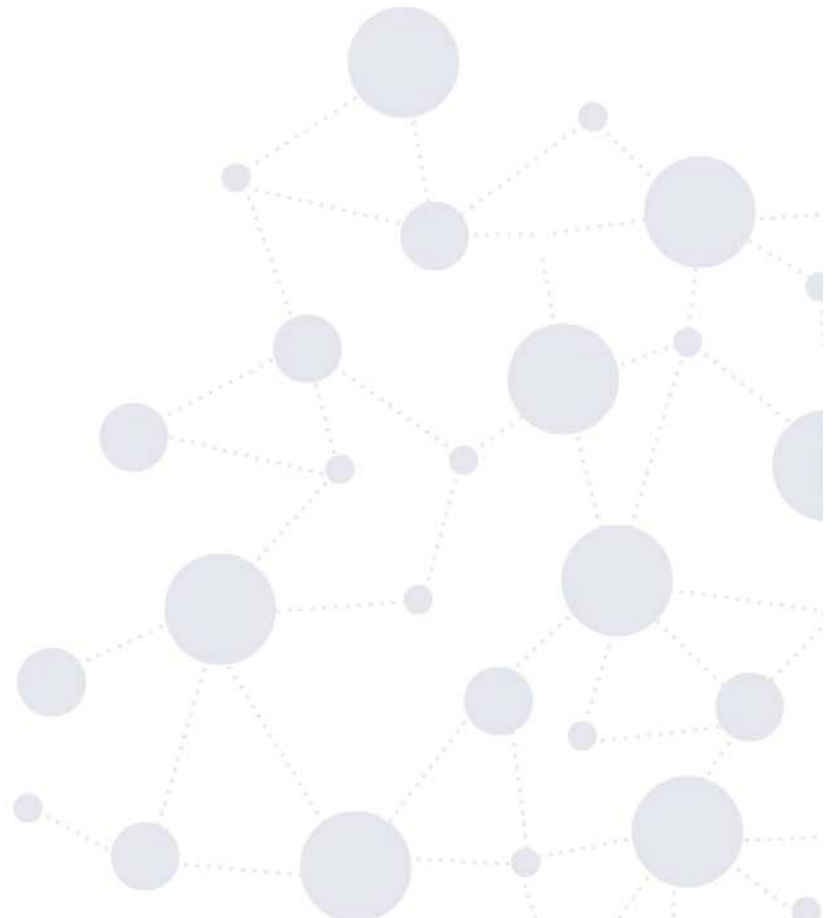
Estos grupos se encuentran descritos en el archivo apps.py del MGB.

Para la asignación de grupos de usuarios se debe localizar a las personas que desempeñan estos roles y asignarles el grupo correspondiente.



## FIRMAS DE RESPONSABILIDAD

Acción	Funcionario	Firmas
Elaborado por:	Jackson Andrews Guzmán Tituana <b>Tesista de la Carrera de Ingeniería en Sistemas</b>	 Digitally signed by Jackson Guzmán Date: 2022.09.02 15:39:41 -05'00'
Revisado por:	Danny Emanuel Muñoz Flores <b>Especialista de Sistemas de información</b>	 Firmado electrónicamente por: <b>DANNY EMANUEL MUNOZ FLORES</b>
	Máximo Andres Alvarez Pacheco <b>Analista de Sistemas de Información 1</b>	 Firmado electrónicamente por: <b>MAXIMO ANDRES ALVAREZ PACHECO</b>
Aprobado por:	Jhon Alexander Calderón Sanmartín <b>Director de Tecnologías de Información</b>	 Firmado electrónicamente por: <b>JHON ALEXANDER CALDERON SANMARTIN</b>





## **Anexo 6. Plan de Pruebas para el MGB**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Plan de Pruebas<sup>8</sup>

---

<sup>8</sup> Versión original en: <https://drive.google.com/Plan-Pruebas-MGB.pdf>



UNL

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

[PLAN DE PRUEBAS]

<<DIRECCIÓN DE BIENESTAR UNIVERSITARIO>>  
<<SUBDIRECCIÓN DE CONTABILIDAD FINANCIERA>>  
<<SUBDIRECCIÓN DE TESORERÍA FINANCIERA>>

# MÓDULO DE GESTIÓN DE BECAS - SIAAF

Versión 22.08

**Elaborado por:**

*Jackson Andrews Guzmán Tituana (tesista)*

**Revisado por:**

*Danny Emanuel Muñoz Flores*

*Máximo Andres Alvarez Pacheco*

*Johana Elizabeth Cueva Dominguez*

**Aprobado por:**

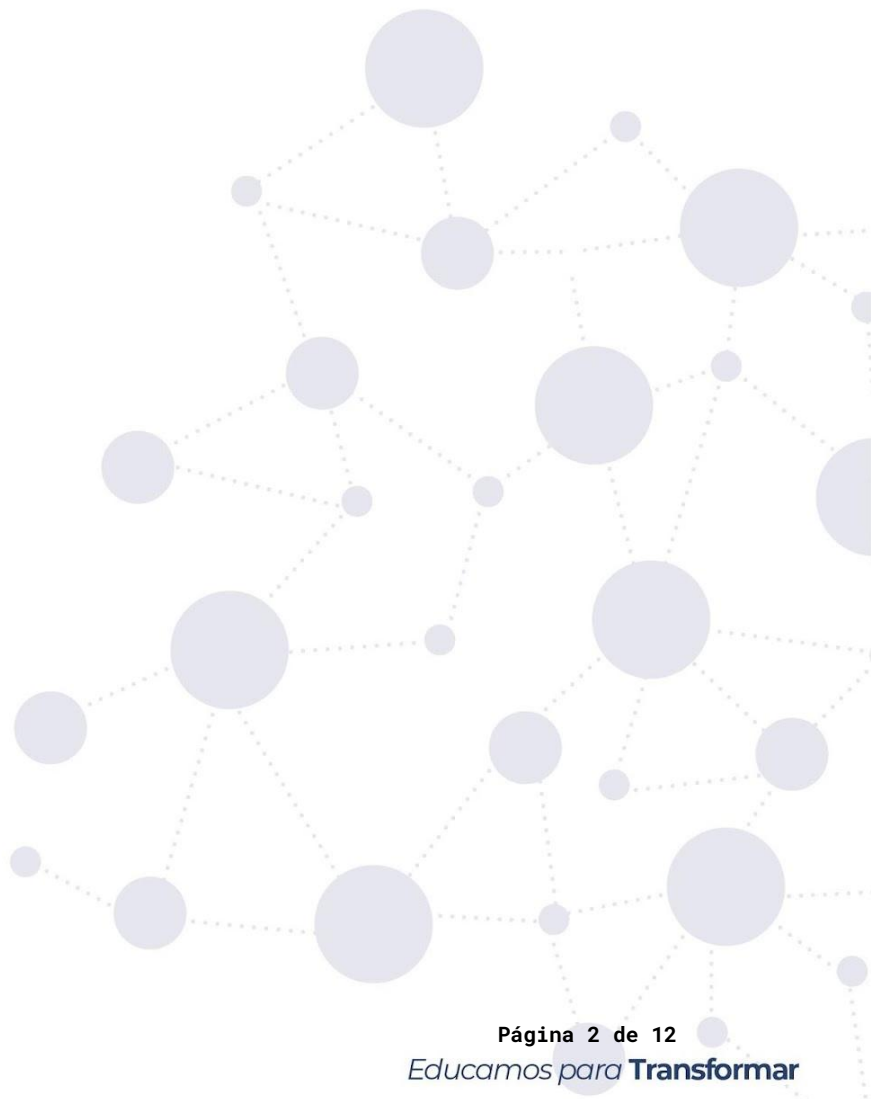
*Jhon Alexander Calderón Sanmartín*

Agosto 2022



## HISTORIAL DE CAMBIOS

<b>Versión</b>	<b>Fecha</b>	<b>Responsable</b>	<b>Descripción de la modificación</b>
22.08	29/08/2022	Tesista Jackson Andrews Guzmán T.	Documento Inicial





**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

## ÍNDICE DE CONTENIDOS

<b>INTRODUCCIÓN</b>	<b>4</b>
<b>REFERENCIAS</b>	<b>4</b>
<b>GLOSARIO</b>	<b>4</b>
<b>CONTEXTO</b>	<b>5</b>
2.1 Proyecto / Subproceso de Prueba	5
2.2 Elementos de Prueba	5
2.3 Alcance de la Prueba	5
2.4 Suposiciones y Restricciones	6
2.5 Partes Interesadas	6
<b>ESTRATEGIAS DE PRUEBA</b>	<b>7</b>
3.1 Subproceso de prueba	7
3.2 Técnicas de diseño de Pruebas	7
3.3 Criterio de Finalización y Prueba	7
3.4 Métricas	7
3.5 Requisitos del entorno de Pruebas	7
3.5.1 Ambiente de Pruebas	7
3.5.2 Herramientas de Pruebas	8
<b>PERSONAL</b>	<b>8</b>
4.1 Roles, Actividades y Responsabilidades	8
4.2 Necesidades de Entrenamiento	9
<b>CRONOGRAMA</b>	<b>9</b>
<b>FIRMAS DE RESPONSABILIDAD</b>	<b>10</b>
<b>ANEXOS</b>	<b>111</b>



## 1. INTRODUCCIÓN

Este documento tiene por propósito proporcionar la información y el marco requerido para planificar y desarrollar las actividades del proceso de pruebas del proyecto 2020-007 SIAAF - Implementación de Módulo de Gestión de Becas y la tesis titulada “Módulo para el Seguimiento y Control de Becas de la Unidad de Bienestar Universitario”, que permita la gestión de las becas a partir de la Etapa de Revisión, pasando hacia las Etapas de Selección, Adjudicación y Concesión. El módulo informático desarrollado en base a los requerimientos descritos en el documento de especificación de requisitos [véase [Documento de Especificación de Requisitos de Software para el Módulo de Gestión de Becas](#)], forma parte del Sistema de Información Académico Administrativo Financiero (SIAAF).

## REFERENCIAS

- [Especificación de Requisitos de Software del Proyecto](#)
- [ISO 29119](#)

## GLOSARIO

En este documento se utilizan los siguientes términos abreviados:

- **DTI:** Dirección de Tecnologías de Información
- **MGB:** Módulo de Gestión de Becas
- **MGT:** Módulo de Gestión de Trámites
- **SGA:** Sistema de Gestión Académica
- **UAT:** Pruebas de Aceptación del Usuario (del inglés User Acceptance Test)
- **QA:** Control de Calidad (del inglés de Quality Assurance)

En este documento se utilizan las siguientes definiciones:

- **Clientes Directos:** usuarios que harán uso del MGB
- **Clientes Indirectos:** personal pertenecientes al DTI que darán soporte y seguimiento al desarrollo del MGB
- **Clientes:** clientes directos e indirectos del MGB

## 2. CONTEXTO

### 2.1 Proyecto / Subproceso de Prueba

El Módulo de Gestión de Becas se compone de los siguientes apartados:

- Tablero de Estadísticas de Becas
- Etapa de Revisión
- Etapa de Selección
- Etapa de Adjudicación
- Etapa de Concesión
- Configuraciones

A continuación se muestran los módulos que se relacionan con el módulo desarrollado (MGB):

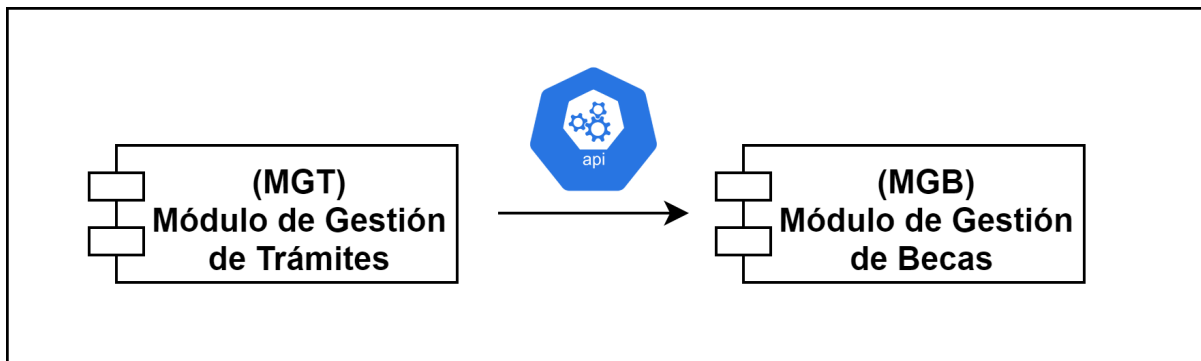


Figura 1. Esquema general de las conexiones entre el MGB y los recursos externos

### 2.2 Elementos de Prueba

Se realizarán pruebas a los siguientes elementos:

- Módulo de Gestión de Becas

### 2.3 Alcance de la Prueba

Las pruebas descritas en este documento se enfocarán principalmente en las funcionalidades desarrolladas para el Módulo de Gestión de Becas para probarse por parte del tesista. Incluye los factores de la calidad no funcionales como el rendimiento y la seguridad informática con las limitaciones definidas a continuación para cada tipo de prueba.

**Pruebas de Casos de Uso.-** Se aplica para validar que se cumplan con los requisitos de software especificados en el [documento de ERS](#) contemplados durante la fase de análisis. Estas pruebas contemplarán únicamente los casos de uso específicos presentes en el documento de ERS.



**Pruebas de Carga y Estabilidad.-** Se enfocarán principalmente en validar que el módulo sea capaz de resistir cuando todos los usuarios que tienen acceso están interactuando al mismo tiempo con los recursos del módulo.

**Pruebas de Seguridad.-** Se enfocarán en determinar si el módulo mantiene todas sus vistas protegidas y asegurar que no es posible saltarse el sistema de autenticación y por otro lado garantizar que cada grupo de usuarios pueda acceder únicamente a los recursos para los que está autorizado interactuar.

**Pruebas Unitarias.-** Se describen con el objetivo de asegurar que aquellas funciones más críticas para el módulo funcione como se espera y no generen inconsistencias con la información.

Finalmente las **Pruebas de Aceptación** estarán enfocadas en el acuerdo con los Clientes de que el software cumple todos los requisitos levantados durante la etapa de análisis para su construcción.

## 2.4 Suposiciones y Restricciones

Suposiciones:

- El ambiente de pruebas contará con datos reales de períodos académicos pasados con el fin de facilitar las pruebas de aceptación para los Clientes Directos.

Restricciones:

- Solo el equipo de QA de DTI tiene acceso al MGT para la validación de la integración con este módulo externo.
- Las Pruebas de Carga y Estabilidad no contemplan escenarios para determinar el número máximo de peticiones en las que el módulo deja de funcionar ya que esto perjudica al servidor y por ende al uso de los otros módulos del SIAAF durante su ejecución.
- Las Pruebas de Seguridad no abarcarán toda la amalgama de pruebas recomendadas por [OWASP](#) en este aspecto.
- Las Pruebas Unitarias no contemplan conexiones a bases de datos ya que para ello se requiere simular la base de datos de todo el proyecto SIAAF.
- Las Pruebas de Código no contemplan la solución de mejoras en el uso de características externas al módulo pero pertenecientes al SIAAF.
- Las Pruebas de Aceptación se realizan en un ambiente de pruebas del SIAAF controlado y simulado con datos reales.

## 2.5 Partes Interesadas

**Tabla 1.** Partes Interesadas en la ejecución del Plan de Pruebas.

Licda. Marlene Valdez	<ul style="list-style-type: none"><li>- Aprobación del Plan de Pruebas, el Cronograma de las Pruebas y los entregables.</li><li>- Realización de las pruebas de aceptación (UAT).</li><li>- Las UAT se darán en el ambiente de simulación de QA para el producto.</li></ul>
Ing. Juan Sebastián Varela	
Lcda. Mónica Raquel Mora	
Dirección de Tecnologías de Información (DTI)	<ul style="list-style-type: none"><li>- Revisión y aprobación del Plan de Pruebas y del Cronograma de Pruebas.</li></ul>

## 3. ESTRATEGIAS DE PRUEBA

### 3.1 Subproceso de prueba

Las pruebas para el Módulo de Gestión de Becas incluirán los siguientes subprocesos de prueba:

- Pruebas de Caso de Uso
- Pruebas de Carga y Estabilidad
- Pruebas de Seguridad
- Pruebas Unitarias
- Pruebas de Aceptación del Usuario

### 3.2 Técnicas de diseño de Pruebas

En este acápite se identifican las técnicas que se utilizarán para el diseño de las pruebas.

- Pruebas de Casos de Uso
- Pruebas Exploratorias

### 3.3 Criterio de Finalización y Prueba

Las pruebas deben alcanzar una cobertura de requisitos del 80% y todos los procedimientos de pruebas deben ejecutarse sin fallas de gravedad 1 (alta).



### 3.4 Métricas

Las siguientes métricas se recogerán durante el transcurso de la ejecución de las pruebas:

- Número de casos de prueba ejecutados.
- Número de incidentes por categoría.
- Número de casos de prueba re-ejecutados.
- Número de incidentes resueltos por categoría.

### 3.5 Requisitos del entorno de Pruebas

#### 3.5.1 Ambiente de Pruebas

**Tabla 2.** Descripción del ambiente de pruebas

<b>Ambiente</b>	<a href="#">Subdominio del servidor SIAAF para el aseguramiento de Calidad (QA)</a>
<b>Navegadores</b>	Chrome, Edge
<b>Sistemas Operativos</b>	Windows, Linux

#### 3.5.2 Herramientas de Pruebas

**Tabla 3.** Descripción de las herramientas de pruebas utilizadas.

<b>Herramienta</b>	<b>Función</b>
Django Test Case	Ejecución y Reporte de las pruebas automatizadas (Pruebas Unitarias)
Apache JMeter	Ejecución y Reporte de las pruebas automatizadas (Pruebas de Carga y Estabilidad, Pruebas de Seguridad)
SonarQube	Ejecución y Reporte de las pruebas automatizadas (Pruebas de Código, Pruebas de Seguridad)
Encuestas Google Forms	Obtener la retroalimentación de los Usuarios (Pruebas de Aceptación)

## 4. PERSONAL

### 4.1 Roles, Actividades y Responsabilidades

La matriz RACI (Responsible-Accountable-Consulted-Informed) a continuación ilustra qué rol está involucrado en qué actividad (es) y cuál es el nivel de participación.

R: Responsable;

A: Apoyo;

C: Consultado;

I: Informado;

**Tabla 4.** Descripción de Roles y Actividades para la ejecución de pruebas.

<b>Rol/Actividad</b>	Pruebas de Carga y Estabilidad	Pruebas de Seguridad	Pruebas Unitarias	Pruebas de Aceptación
Product Owner	I	I	I	A
Analista de QA	I	I	I	A
Tesista	R	R	R	R

### 4.2 Necesidades de Entrenamiento

Se necesita una introducción al módulo desarrollado para los que van a ejecutar las pruebas. Se estima que esta actividad puede tomar 1 hora.

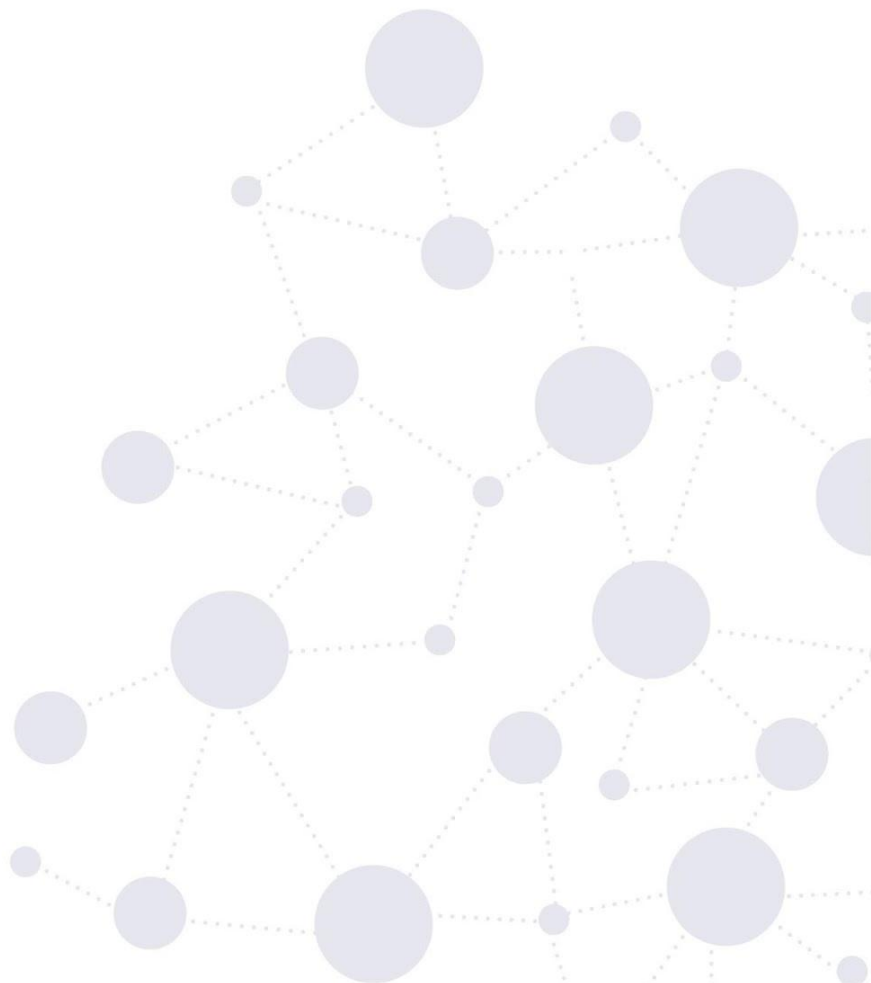


## 5. CRONOGRAMA

El cronograma general de las pruebas se muestra a continuación.

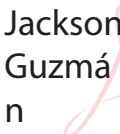




**Tabla 5.** Estimación del tiempo de ejecución de las pruebas descritas en este documento.

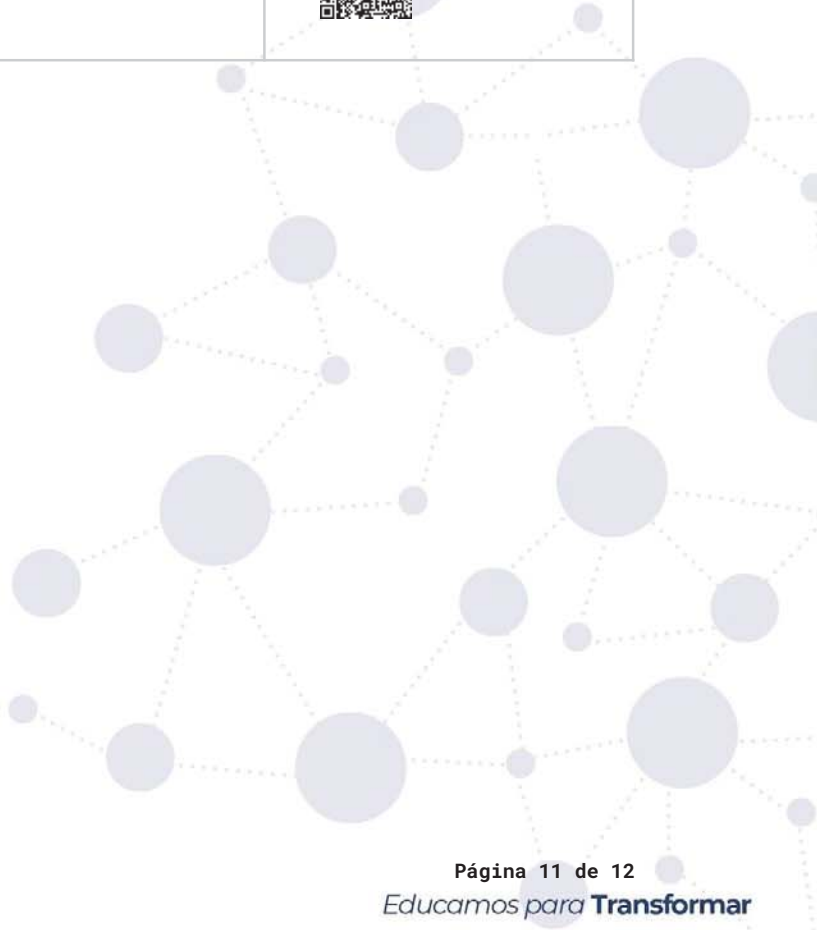
Actividad de Pruebas/Semana	1	2	3	4
Análisis y Diseño	X			
Entorno de Pruebas		X		
Ejecución		X	X	
Finalización				X





## FIRMAS DE RESPONSABILIDAD

Acción	Funcionario	Firmas
Elaborado por:	Jackson Andrews Guzmán Tituana <b>Tesista</b>	 Digitally signed by Jackson Guzmán Date: 2022.09.02 15:37:24 -05'00'
Revisado por:	Danny Emanuel Muñoz Flores <b>Especialista en Sistemas de Información</b>	 Firmado electrónicamente por: <b>DANNY EMANUEL MUNOZ FLORES</b>
	Máximo Andres Alvarez Pacheco <b>Analista de Sistemas de Información 1</b>	 Firmado electrónicamente por: <b>MAXIMO ANDRES ALVAREZ PACHECO</b>
	Johana Elizabeth Cueva Dominguez <b>Analista de Sistemas Informáticos 1</b>	 Firmado electrónicamente por: <b>JOHANA ELIZABETH CUEVA DOMINGUEZ</b>
Aprobado por:	Jhon Alexander Calderón Sanmartín <b>Director de Tecnologías de Información</b>	 Firmado electrónicamente por: <b>JHON ALEXANDER CALDERON SANMARTIN</b>





**UNL**

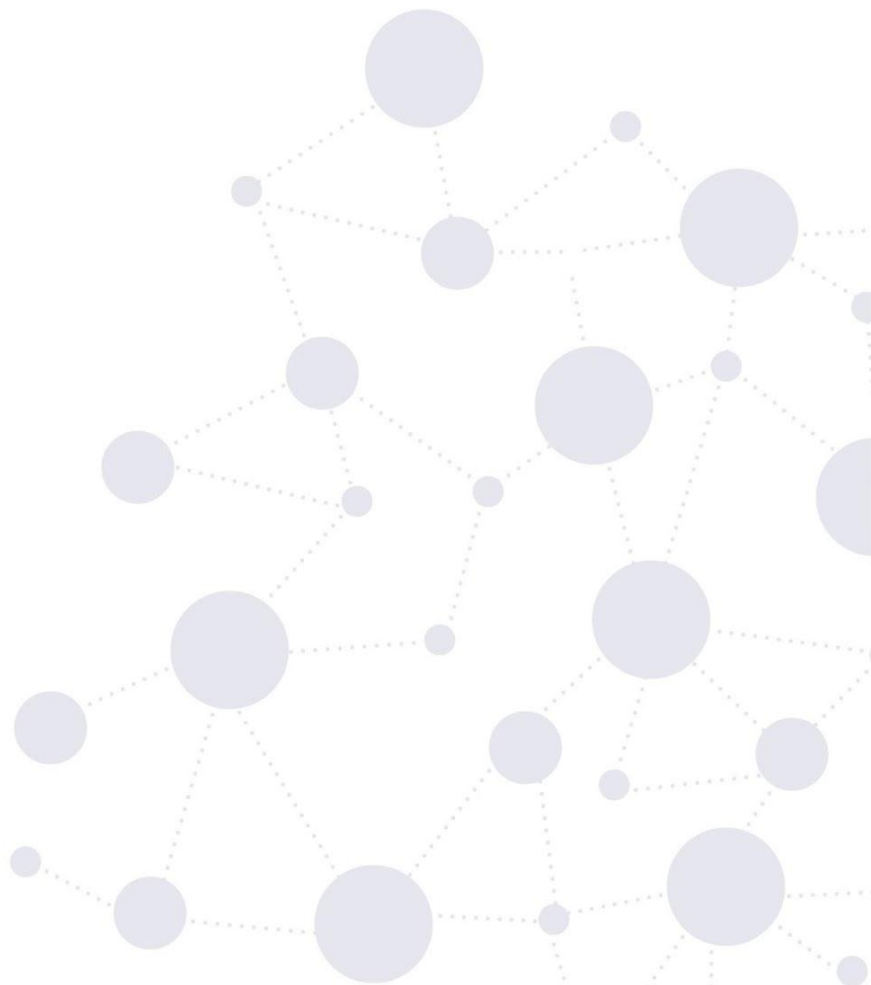
Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

## ANEXOS

### Documento de Especificación de Requisitos de Software para el Módulo de Gestión de Becas

📄 ERS\_IEEE\_830 para el MGB - firmado.pdf



## **Anexo 7. Ejecución del Plan de Pruebas para el MGB**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Ejecución del Plan de Pruebas<sup>9</sup>

---

<sup>9</sup> Versión original en: <https://drive.google.com/Ejecucion-Plan-Pruebas-MGB.pdf>



UNL

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

[EJECUCIÓN DEL PLAN DE PRUEBAS]

<<DIRECCIÓN DE BIENESTAR UNIVERSITARIO>>

<<SUBDIRECCIÓN DE CONTABILIDAD FINANCIERA>>

<<SUBDIRECCIÓN DE TESORERÍA FINANCIERA>>

# MÓDULO DE GESTIÓN DE BECAS - SIAAF

Versión 22.08

**Elaborado por:**

*Jackson Andrews Guzmán Tituana (tesista)*

**Revisado por:**

*Danny Emanuel Muñoz Flores*

*Máximo Andres Alvarez Pacheco*

*Johana Elizabeth Cueva Dominguez*

**Aprobado por:**

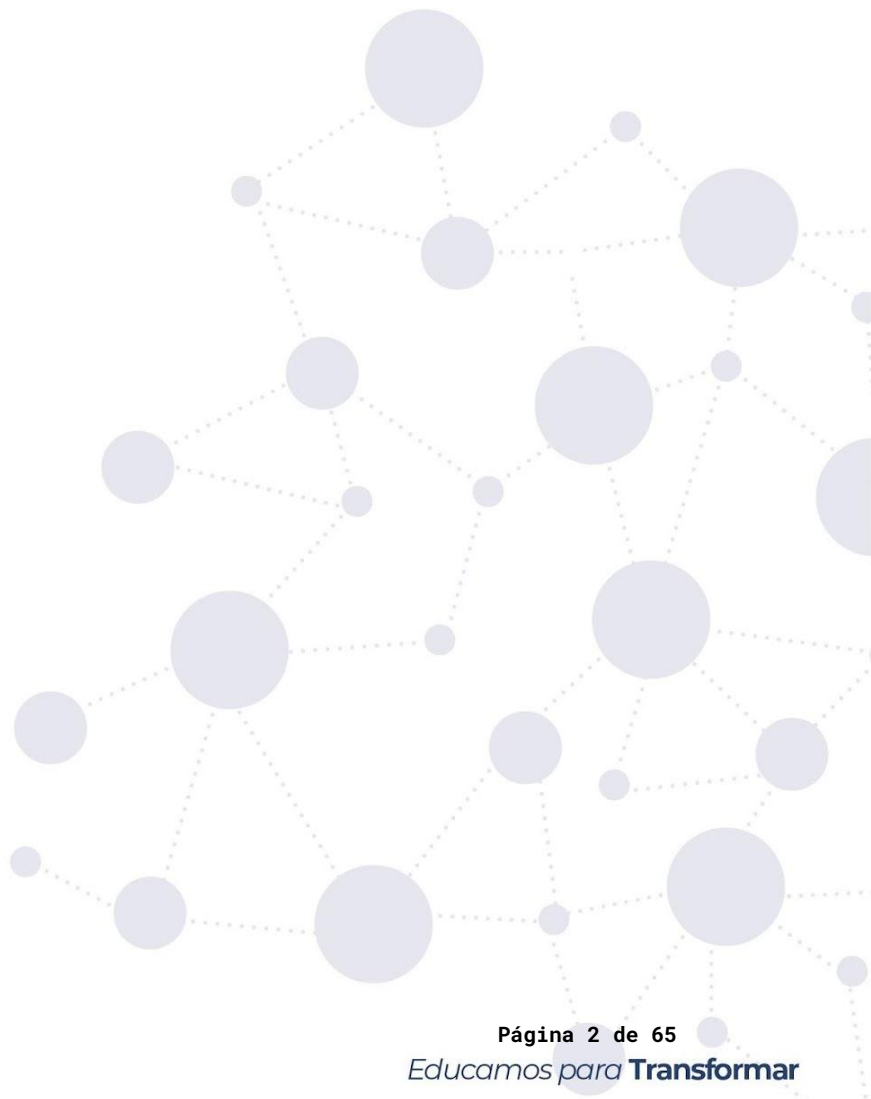
*Jhon Alexander Calderón Sanmartín*

Agosto 2022



## HISTORIAL DE CAMBIOS

Versión	Fecha	Responsable	Descripción de la modificación
22.08	29/08/2022	Tesista Jackson Andrews Guzmán T. Ing. Máximo Andres Alvarez Pacheco	Documento Inicial







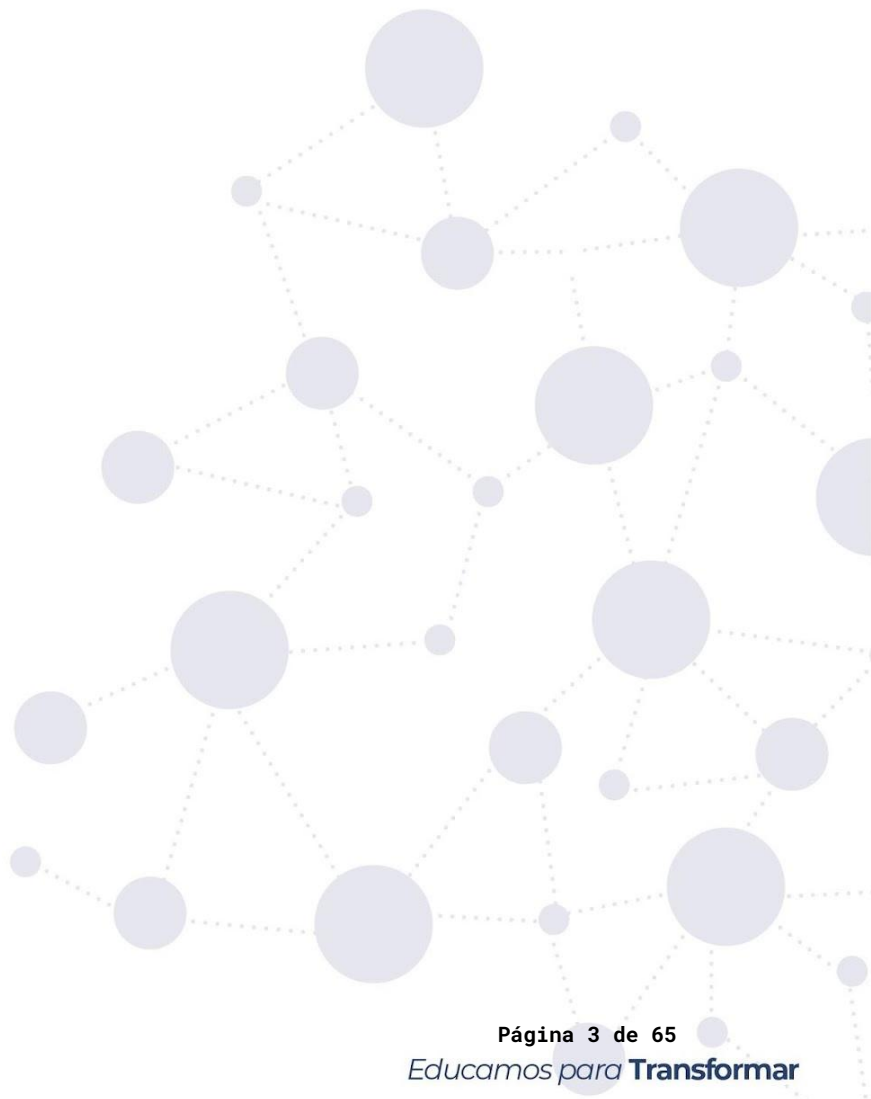
**UNL**

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

## ÍNDICE DE CONTENIDOS

<b>INTRODUCCIÓN</b>	<b>4</b>
<b>REFERENCIAS</b>	<b>5</b>
<b>GLOSARIO</b>	<b>5</b>
<b>CONTEXTO</b>	<b>6</b>
2.1 Proyecto / Subproceso de Prueba	6
2.2 Partes Interesadas	6
<b>PRUEBAS</b>	<b>7</b>
3.1 Pruebas de los Casos de Uso	7
3.1 Pruebas de Carga y Estabilidad	18
3.2 Pruebas de Seguridad	24
3.3 Pruebas Unitarias	25
3.4 Pruebas de Código	34
3.5 Pruebas de Aceptación del Usuario	38
<b>FIRMAS DE RESPONSABILIDAD</b>	<b>48</b>
<b>ANEXOS</b>	<b>49</b>





## 1. INTRODUCCIÓN

Este documento tiene por propósito proporcionar la información recopilada en el desarrollo de las actividades del proceso de pruebas para el proyecto titulado “Módulo para el Seguimiento y Control de Becas de la Unidad de Bienestar Universitario”, que permita la gestión de las becas a partir de la Etapa de Revisión, pasando hacia las Etapas de Selección, Adjudicación y Concesión. El módulo informático desarrollado en base a los requerimientos descritos en el documento de especificación de requisitos [véase [Documento de Especificación de Requisitos de Software para el Módulo de Gestión de Becas](#)], forma parte del Sistema de Información Académico Administrativo Financiero (SIAAF).

## REFERENCIAS

- [Especificación de Requisitos de Software del Proyecto](#)
- [Plan de Pruebas para el Módulo de Gestión de Becas](#)
- [ISO 29119](#)

## GLOSARIO

En este documento se utilizan los siguientes términos abreviados:

- **DTI:** Dirección de Tecnologías de Información
- **MGB:** Módulo de Gestión de Becas
- **MGT:** Módulo de Gestión de Trámites
- **SGA:** Sistema de Gestión Académica
- **UAT:** Pruebas de Aceptación del Usuario (del inglés User Acceptance Test)
- **QA:** Control de Calidad (del inglés Quality Assurance)
- **eSIGEF:** Sistema de Administración Financiera
- **WSL:** Subsistema de Windows para Linux (del inglés Windows Subsystem for Linux)

En este documento se utilizan las siguientes definiciones:

- **Clientes Directos:** usuarios que harán uso del MGB
- **Clientes Indirectos:** personal pertenecientes al DTI que darán soporte y seguimiento al desarrollo del MGB
- **Clientes:** conformado por los Clientes Directos e Indirectos del MGB

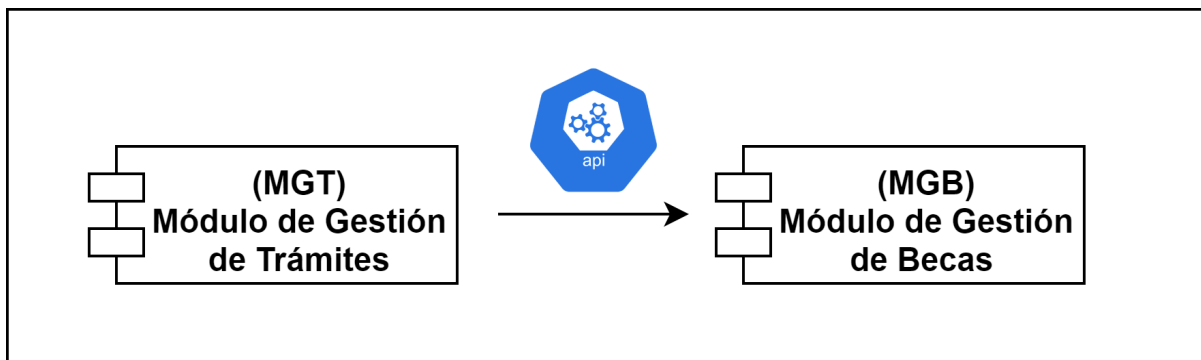
## 2. CONTEXTO

### 2.1 Proyecto / Subproceso de Prueba

El Módulo de Gestión de Becas se compone de los siguientes apartados:

- Tablero de Estadísticas de Becas
- Etapa de Revisión
- Etapa de Selección
- Etapa de Adjudicación
- Etapa de Concesión
- Configuraciones

A continuación se muestran los módulos que se relacionan con el módulo desarrollado (MGB):



**Figura 1.** Esquema general de las conexiones entre el MGB y los recursos externos [gráfico]

### 2.2 Partes Interesadas

**Tabla 1.** Partes Interesadas en la ejecución del Plan de Pruebas.

Lcda. Marlene Valdez	<ul style="list-style-type: none"> <li>- Participación en las Pruebas de Aceptación (UAT). Las UAT se darán en el ambiente de simulación de QA para el producto.</li> <li>- Aprobación de las Pruebas de Aceptación de Usuario (UAT).</li> </ul>
Ing. Juan Sebastián Varela	
Lcda. Mónica Raquel Mora	
Dirección de Tecnologías de Información (DTI)	<ul style="list-style-type: none"> <li>- Revisión y aprobación del Plan de Pruebas y del Cronograma de Pruebas.</li> </ul>

### 3. PRUEBAS

#### 3.1 Pruebas de los Casos de Uso

Para realizar estas pruebas se consideraron los casos de uso descritos en el [documento de ERS](#) perteneciente al MGB contemplados durante la fase de análisis. Las pruebas consisten en validar el cumplimiento de los Requisitos de Software que se analizaron en un inicio tanto para Requisitos Funcionales como no Funcionales, los resultados obtenidos se detallan a continuación [véase [Pruebas de Casos de Uso](#)].

**Tabla 2.** Casos de Prueba validados en relación a los Requisitos Funcionales del MGB.

CASOS DE USO	ROL	RESULTADOS ESPERADOS	QA
Sincronizar datos de postulación desde el MGT hacia el MGB	Administrador de Becas	El sistema permite registrar la información de los postulantes en el MGB luego de haber llamado al endpoint para registrar postulantes del MGB	OK
Actualizar estados de postulaciones	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la actualización fue exitosa luego de haber validado que se puede actualizar al estado Revisar independientemente de haber calificado o no la ficha socioeconómica	OK
Actualizar estados de postulaciones	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la actualización fue exitosa luego de haber validado que se puede actualizar al estado Revisar independientemente de haber calificado o no la ficha socioeconómica	OK
Actualizar estados de postulaciones	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la actualización fue exitosa luego de haber comprobado que la ficha socioeconómica se encuentra calificada	OK
Actualizar estados de postulaciones	Administrador de Becas	El sistema mostrará un mensaje de advertencia solicitando que se califique la ficha socioeconómica luego de haber validado que la ficha socioeconómica no se encuentra calificada	OK
Actualizar estados de postulaciones	Administrador de Becas	El sistema mostrará un mensaje de advertencia solicitando que se califique la ficha socioeconómica luego de haber validado que la ficha socioeconómica no se encuentra calificada	OK
Actualizar estados de postulaciones	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la actualización fue exitosa luego de haber comprobado que la ficha socioeconómica se encuentra calificada	OK
Actualizar estados de postulaciones	Administrador de Becas	El sistema mostrará un mensaje de éxito	OK



1859

postulaciones	de Becas	indicando que la actualización fue exitosa luego de haber validado que se puede actualizar al estado Revisar independientemente de haber calificado o no la ficha socioeconómica	
Actualizar estados de postulaciones	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la actualización fue exitosa luego de haber validado que se puede actualizar al estado Revisar independientemente de haber calificado o no la ficha socioeconómica	OK
Actualizar estados de postulaciones	Administrador de Becas	El sistema mostrará un mensaje de advertencia indicando que no se puede actualizar el estado de una etapa finalizada luego de haber validado que la etapa se encuentra finalizada	OK
Descargar documentos de postulación	Administrador de Becas	El sistema descargará los archivos comprimidos de los postulantes mostrando un mensaje de éxito confirmando la acción luego de haber llamado al endpoint del MGT para obtener los documentos	OK
Descargar documentos de postulación	Administrador de Becas	El sistema descargará los archivos comprimidos de los postulantes mostrando un mensaje de éxito confirmando la acción luego de haber llamado al endpoint del MGT para obtener los documentos	OK
Ver trámite en el MGT	Administrador de Becas	El sistema abrirá nuevas pestañas con un enlace que redirige al trámite correspondiente en el MGT luego de haber armado la ruta de acceso con la url del MGT y el código del proceso	OK
Ver trámite en el MGT	Administrador de Becas	El sistema abrirá nuevas pestañas con un enlace que redirige al trámite correspondiente en el MGT luego de haber armado la ruta de acceso con la url del MGT y el código del proceso	OK
Finalizar Etapas	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la etapa se finalizó correctamente luego de comprobar que la etapa no se encuentra finalizada	OK
Finalizar Etapas	Administrador de Becas	El sistema mostrará un mensaje de error indicando que no se puede finalizar la etapa mientras no se haya aprobado o rechazado a todos los postulantes luego de validar que existen postulantes que se encuentran en el estado de Revisión o Pendiente	OK
Finalizar Etapas	Administrador de Becas	El sistema no habilitará el botón de Finalizar Etapa luego de verificar que la	OK



1859

		etapa ya se encuentra finalizada	
Actualizar estados de Becas	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la actualización fue exitosa luego de verificar que independientemente del estado de la confirmación de registro en el sistema financiero se pueda Suspendir la Beca	OK
Actualizar estados de Becas	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la actualización fue exitosa luego de verificar que independientemente del estado de la confirmación de registro en el sistema financiero se pueda Suspendir la Beca	OK
Actualizar estados de Becas	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la actualización fue exitosa luego de verificar que independientemente del estado de la confirmación de registro en el sistema financiero se pueda Terminar la Beca	OK
Actualizar estados de Becas	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la actualización fue exitosa luego de verificar que independientemente del estado de la confirmación de registro en el sistema financiero se pueda Terminar la Beca	OK
Actualizar estados de Becas	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la actualización fue exitosa luego de verificar que independientemente del estado actual se pueda Reactivar la Beca	OK
Actualizar estados de Becas	Administrador de Becas	El sistema mostrará un mensaje de éxito indicando que la actualización fue exitosa luego de verificar que independientemente del estado actual se pueda Reactivar la Beca	OK
Actualizar estados de Becas	Administrador de Becas	El sistema mostrará un mensaje de advertencia indicando que no se puede actualizar al mismo estado y solicitando que se elija un estado distinto luego de comprobar que el Becado se encuentra en el mismo estado	OK
Visualizar Postulaciones y Becas	Administrador de Becas	El sistema mostrará los registros de postulaciones o becas disponibles para cada periodo académico luego de verificar que existan registros	OK
Visualizar Postulaciones y Becas	Tesorería Financiera	El sistema mostrará los registros de postulaciones o becas disponibles para cada periodo académico luego de verificar que existan registros	OK
Visualizar Postulaciones y Becas	Contabilidad Financiera	El sistema mostrará los registros de postulaciones o becas disponibles para	OK





1859

			cada periodo académico luego de verificar que existan registros	
Generar Reportes		Administrador de Becas	El sistema descarga el reporte en formato excel e indicará con un mensaje de éxito que la solicitud pudo completarse correctamente luego de haber obtenido los datos desde la Base de Datos en base a los filtros seleccionados y a armar el archivo blob	OK
Generar Reportes		Tesorería Financiera	El sistema descarga el reporte en formato excel e indicará con un mensaje de éxito que la solicitud pudo completarse correctamente luego de haber obtenido los datos desde la Base de Datos en base a los filtros seleccionados y a armar el archivo blob	OK
Generar Reportes		Contabilidad Financiera	El sistema descarga el reporte en formato excel e indicará con un mensaje de éxito que la solicitud pudo completarse correctamente luego de haber obtenido los datos desde la Base de Datos en base a los filtros seleccionados y a armar el archivo blob	OK
Generar Reportes		Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema deshabilitará el botón para Generar Reportes luego de verificar que la tabla se encuentra vacía	OK
Agregar frecuente	mensaje	Administrador de Becas	El sistema mostrará un mensaje de éxito y actualizará la página para mostrar el mensaje agregado luego de guardar correctamente el nuevo mensaje	OK
Actualizar frecuente	mensaje	Administrador de Becas	El sistema mostrará un mensaje de éxito y actualizará la página para mostrar el mensaje actualizado luego de verificar que se trata de una operación POST y verificar que datos se cambiaron para actualizar otros registros de ser necesario	OK
Eliminar frecuente	mensaje	Administrador de Becas	El sistema mostrará un mensaje de éxito y actualizará la página para mostrar que el mensaje fue eliminado luego de verificar el orden del mensaje y reordenar los otros mensajes para cubrir la posición faltante	OK
Marcar como favorito mensaje frecuente		Administrador de Becas	El sistema mostrará un mensaje de éxito y actualizará la página para indicar con una estrella que el mensaje frecuente ahora es el favorito luego de verificar si existe un mensaje del mismo tipo que es favorito para actualizarlo como no favorito y poder obtener un único favorito por tipo	OK



1859

Actualizar montos de pago para los tipos de becas	Administrador de Becas	El sistema mostrará un mensaje de éxito y actualizará la página para mostrar el nuevo monto de pago ingresado luego de verificar que el monto ingresado sea un monto válido	OK
Visualizar estadísticas generales de becados	Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema mostrará las estadísticas generales disponibles de postulaciones, becas y montos de pago luego de intentar obtener los datos de postulaciones, becas y registros de pagos desde la Base de Datos	OK
Iniciar Sesión	Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema debe permitir el acceso a los usuarios configurados previamente con los grupos de usuario y permisos correspondientes luego de verificar si tiene permitido acceder al sistema	OK
Notificar resultados de postulación	Administrador de Becas	El sistema debe guardar las notificaciones de los postulantes aprobados y rechazados con la razón proporcionada durante la Revisión y Selección para ser enviadas cuando sea oportuno	OK
Notificar resultados de postulación	Administrador de Becas	El sistema debe guardar la notificación para el usuario de Tesorería y enviarla al correo cuando sea oportuno, y debe mostrarse en las notificaciones del SIAAF luego de obtener el parámetro del correo del usuario de tesorería	OK
Guardar confirmación de registro de beca del sistema financiero	Tesorería Financiera	El sistema mostrará un mensaje de éxito indicando que la confirmación fue exitosa luego de verificar que el registro no se encuentre ya confirmado	OK
Guardar confirmación de registro de beca del sistema financiero	Tesorería Financiera	El sistema mostrará un mensaje de éxito indicando que los registros seleccionados ya han sido confirmados como registrados luego de verificar que los registros ya se encuentran confirmados	OK
Registrar Pago de Becas	Tesorería Financiera	El sistema mostrará un mensaje de éxito indicando que los registros fueron correctamente importados, mostrando el número de registros importados y una tabla con la información cargada luego de haber actualizado los registrados de pago leídos en el archivo excel y realizado las validaciones de los datos correspondientes	OK
Registrar Pago de Becas	Tesorería Financiera	El sistema mostrará mensajes de advertencia indicando que no se pudo procesar el archivo luego de verificar que no se encontró la columna NRO_IDENTIFICACION	OK
Registrar Pago de	Tesorería	El sistema mostrará mensajes de	OK





1859	Becas	Financiera	advertencia indicando que no se pudo procesar el archivo y mostrando un botón para descargar el reporte con errores luego de verificar que el archivo no cuenta con todos los datos requeridos para procesar la importación	
Registrar Becas	Pago de	Tesorería Financiera	El sistema mostrará un mensaje de éxito indicando que se actualizó correctamente luego de verificar que el mes, el estado y el nro expediente sean válidos	OK
Registrar Becas	Pago de	Tesorería Financiera	El sistema mostrará un mensaje de advertencia solicitando que se elija por lo menos un mes para concesionar luego de verificar que no se seleccionó un solo mes	OK
Registrar Becas	Pago de	Tesorería Financiera	El sistema mostrará un mensaje de un mensaje de advertencia mostrando que el campo nro de expediente no es válido luego de verificar que el registro seleccionado no tiene el campo nro expediente y no se ha ingresado un número de expediente válido	OK
Registrar Becas	Pago de	Tesorería Financiera	El sistema mostrará un mensaje de un mensaje de advertencia mostrando que el estado de concesión debe ser distinto al actual luego de verificar que el estado del registro es el mismo al que se intenta actualizar	OK
Registrar Becas	Pago de	Tesorería Financiera	El sistema mostrará un mensaje de éxito informando que la actualización se dio correctamente luego de verificar que el registro seleccionado ya cuenta con un nro de expediente registrado	OK

**Tabla 3.** Casos de Prueba ejecutado en relación a los Requisitos no Funcionales del MGB

CASOS DE PRUEBA	ROL	RESULTADOS ESPERADOS	QA
Las transacciones de lectura de información deben realizarse en menos de 2s.	Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema debe tardar menos de 2s en cargar la información solicitada luego de solicitar la información a la Base de Datos	OK
Las transacciones de lectura deben cargarse bajo demanda organizados en lotes de 10, 25 y 50 registros al mismo tiempo.	Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema debe mostrar la información de acuerdo al número de registros solicitados luego de obtener el número de registros deseados a mostrarse	OK
En el 90% de los casos las transacciones de guardado y	Administrador de Becas Tesorería	El sistema debe tardar menos de 2s en operaciones de guardado y actualización luego de guardar o actualizar registros en la	OK

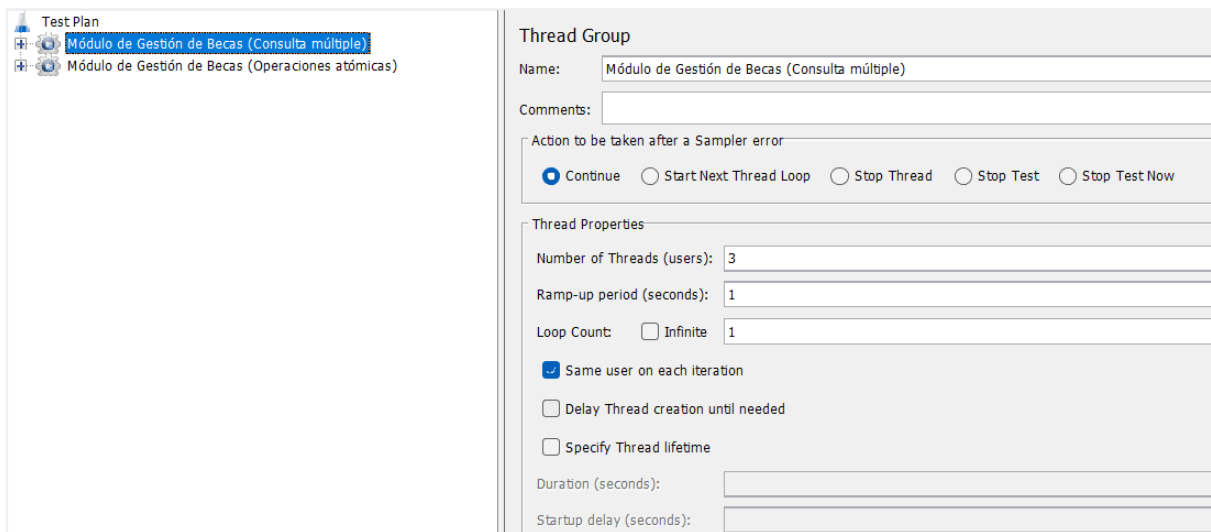


1859

modificación deben tardar 2s o menos.	Financiera Contabilidad Financiera	Base de Datos	
El sistema será capaz de permitir la conexión simultánea de 2 o más usuarios al mismo tiempo.	Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema debe permitir operar a los distintos usuarios al mismo tiempo luego de verificar el grupo de usuario al que pertenece	OK
El módulo informático permitirá únicamente el acceso a los administradores del SIAAF, al Responsable de la Gestión de Becas, al Responsable del Registro de Pagos y al Contador Financiero	Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema debe permitir ingresar únicamente a los usuarios del MGB luego de verificar el grupo de usuario al que pertenece	OK
El módulo informático hará la distinción entre los roles y privilegios que tiene cada usuario.	Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema debe permitir acceder a los distintos apartados únicamente a los usuarios correspondientes de acuerdo a los privilegios concedidos para cada rol de usuario del MGB luego de verificar los permisos del usuario	OK
La sesión del usuario utilizará el servicio de autenticación implementado para el SIAAF.	Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema debe permitir acceder a través de la autenticación del SIAAF	OK
La seguridad de la aplicación está cubierta por la arquitectura provista por el SIAAF.	Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema no debe permitir acceder a ninguna de las rutas del MGB luego de verificar que no existe token de sesión en las cookies	OK
El módulo informático está pensado para interactuar activamente con el MGT, por lo que se hará uso de apis para permitir la comunicación entre estos.	Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema debe recibir la información de postulación del MGT luego de verificar que se reciban todos los datos	OK
El Módulo puede accederse desde cualquier dispositivo con conexión a internet para su uso.	Administrador de Becas Tesorería Financiera Contabilidad Financiera	El sistema debe ser accesible desde cualquier dispositivo móvil con conexión a internet	OK

### 3.1 Pruebas de Carga y Estabilidad

Para la ejecución de las Pruebas de Carga y Estabilidad se realizó la siguiente configuración en Apache JMeter [véase figura 2], con una población de 3 usuarios, ya que se corresponde al número máximo de usuarios simultáneos que interactúan con el módulo. Las pruebas se dividen en 2 grupos de hilos, en el primer hilo están asociadas todas las peticiones que se pueden realizar consultas múltiples, y en el segundo grupo de hilos están peticiones atómicas es decir, aquellas que se pueden realizar una única vez.

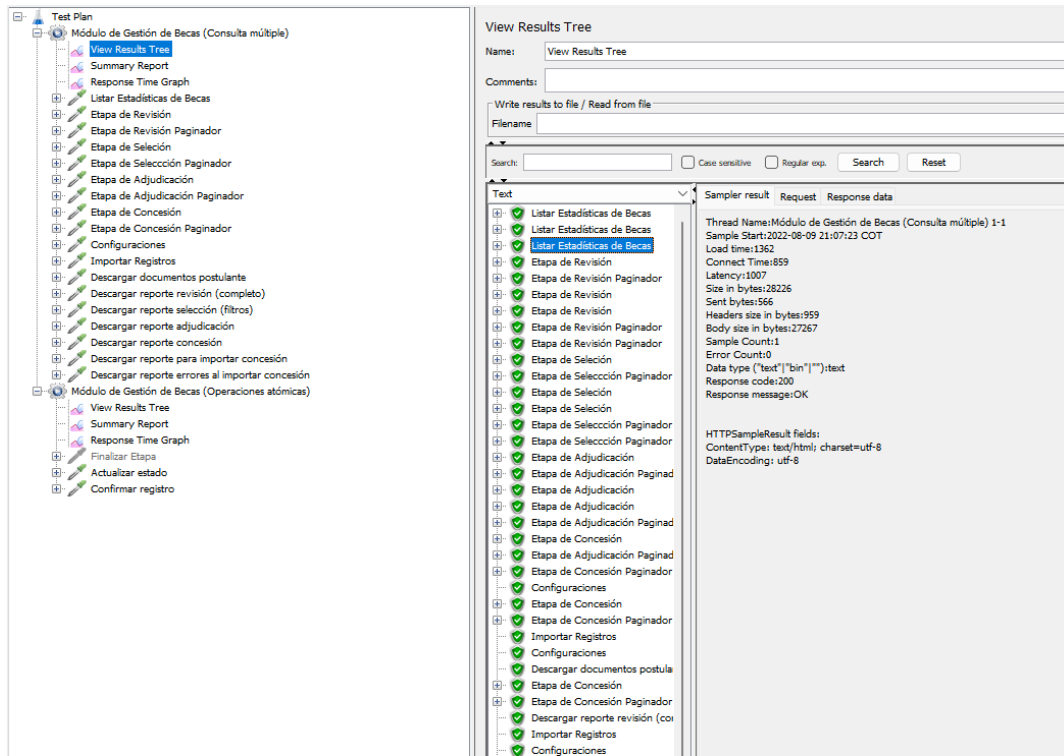


**Figura 2.** Configuración del Plan de Pruebas con Apache JMeter 5.5 [captura de pantalla]

Dentro del grupo de hilos de de consultas múltiples se encuentran peticiones de lectura GET [véase figura 3], como la carga de cada uno de los apartados del módulo así como sus correspondientes petición paginadora, Además están peticiones POST para la descarga de reportes de cada una de las etapas y la petición de importación de archivos.

En el hilo de consultas atómicas, se encuentran las peticiones de actualización de estado de las etapas [véase figura 3], la finalización de etapas y la confirmación de registro para su concesión; todas peticiones POST de escritura.

Para iniciar la solicitud de las peticiones desde Apache JMeter fue necesario configurar las cookies y las cabeceras, para que el servidor no interprete como una amenaza y permita realizar las consultas, ya que los recursos del módulo se encuentran protegidos y es necesario identificarse con el token de sesión como se muestra en la figura 27 y la Figura 28 de los Anexos. Tanto las cookies como las cabeceras se obtuvieron desde una sesión de usuario del navegador.



**Figura 3.** Ejecución de Pruebas de carga y estabilidad con JMeter [captura de pantalla]

Todas las peticiones de consulta múltiple se ejecutaron con éxito como se muestra en la figura 3. Los tiempos de respuesta obtenidos se presentan en la tabla 4 la cual organiza a los recursos consultados de MGB indicando el número de muestras simultáneas para cada recurso, el tiempo promedio, el tiempo mínimo, el tiempo máximo y la desviación estándar.

**Tabla 4.** Reporte resumen de los tiempos de respuesta del grupo de hilos de consultas múltiples exportado por Apache Jmeter [véase figura 32 para captura original]

Recursos	# Muestras	Promedio	Min	Max	Desv. Std
Listar Estadísticas de Becas	3	417	413	422	3.74
Etapa de Revisión	3	292	269	313	18.06
Etapa de Revisión Paginador	3	68	54	76	9.93
Etapa de Selección	3	372	272	484	86.92
Etapa de Selección Paginador	3	54	51	59	3.4
Etapa de Adjudicación	3	291	254	349	41.36
Etapa de Adjudicación Paginador	3	63	52	81	12.83
Etapa de Concesión	3	259	258	261	1.25
Etapa de Concesión Paginador	3	51	49	53	1.63
Configuraciones	3	197	164	223	24.8
Importar Registros	3	181	175	194	8.96



1859

Descargar documentos postulante	3	774	721	847	53.11
Descargar reporte revisión (completo)	3	90	75	111	15.17
Descargar reporte selección (filtros)	3	89	80	106	11.81
Descargar reporte adjudicación	3	142	132	157	10.66
Descargar reporte concesión	3	74	66	89	10.21
Descargar reporte para importar concesión	3	169	92	323	108.66
Descargar reporte errores al importar concesión	3	36	26	52	11.43
TOTAL	54	201	26	847	183.25

Tal como se indica en la tabla 4 la consulta con el tiempo mínimo es de 26 ms que corresponde al recurso de descarga de reporte de errores al importar concesión, con un máximo de 52 ms y un promedio de 36 ms; esto hace mucho sentido ya que este reporte tiende a ser pequeño porque incluye sólo aquellos registros que no pudieron procesarse. Por otro lado el recurso que ocupa más tiempo en ejecutarse es la descarga de documentos de postulante con 847 ms como tiempo máximo, un tiempo mínimo de 721 ms con un promedio de 774 ms, esto se debe a que para obtener los documentos del postulante debe consultarse un endpoint del Módulo de Gestión de Trámites el cuál retorna los archivos en formato blob que el MGB debe convertirlos en archivo descargable.

Todos los recursos del MGB no superan el segundo de respuesta considerando una interacción en simultáneo de todos los usuarios. Esto demuestra el cumplimiento de los requisitos no funcionales de rendimiento especificados en la ERS [véase Documento de Especificación de Requisitos, sección Requerimientos no Funcionales].

Observando la gráfica de la figura 4 se aprecia que solo 2 peticiones de las 18 en total en su promedio de tiempo de respuesta superan los 400 ms, esto quiere decir que a las 16 solicitudes equivalentes al 89% del total de solicitudes les toma menos de 400 ms de ejecución. Al 11% restante le toma menos de 600 ms en ejecutarse. La gráfica además muestra que el 56% de las solicitudes tienen incluso tiempos de respuesta muy por debajo de los 200 ms.



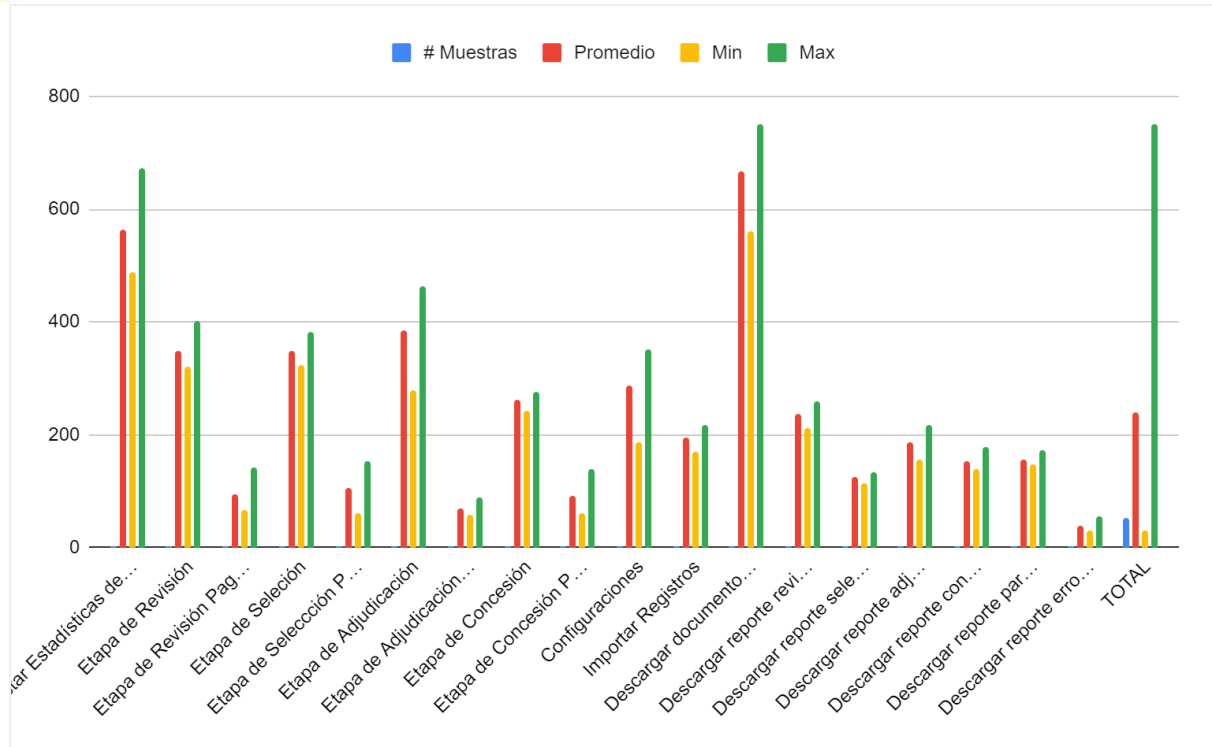


Figura 4. Gráfica del tiempo de respuesta de las peticiones del hilo de consultas múltiples [gráfico]

Para asegurar la estabilidad del MGB se creó un grupo de hilos específico con todas las peticiones GET manteniendo los parámetros de configuración de 3 usuarios simultáneamente cada segundo y habilitando el tiempo de ejecución de pruebas indefinido como se muestra en la figura 5.

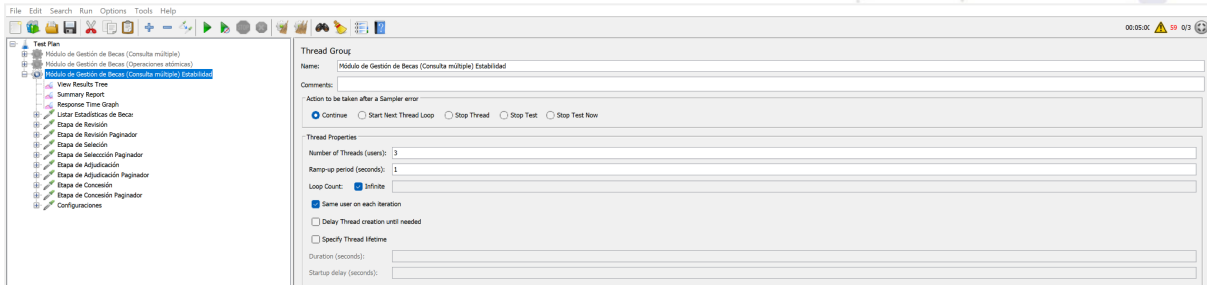


Figura 5. Configuración del grupo de hilos para las Pruebas de Estabilidad con Apache JMeter 5.5 [captura de pantalla]

Para esta prueba se dejó activa la ejecución por 5 minutos exactos, ejecutándose 425 muestras en promedio para cada recurso dando un total de 4257 peticiones a los recursos del módulo. Se obtuvo un tiempo de respuesta mínimo de 44 ms un tiempo máximo de 1108 ms y el tiempo promedio de ejecución es de 211 ms con una desviación estándar de 122 ms y un porcentaje de error del 0% para todos los casos [véase figura 6].

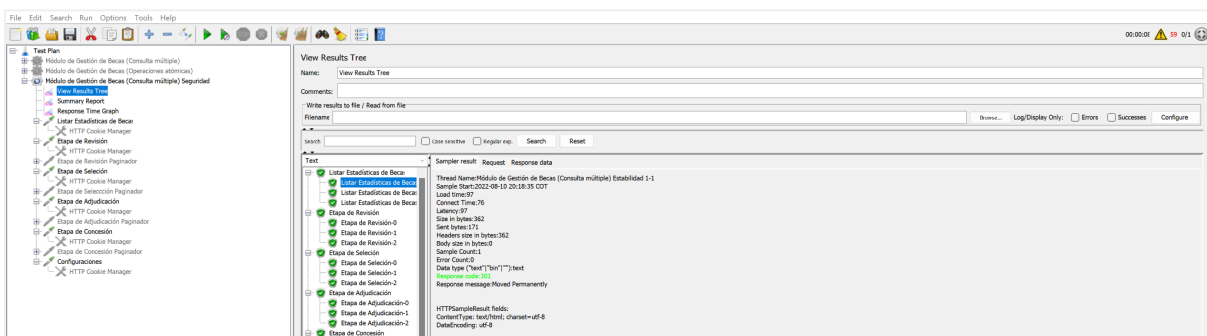


<b>Número de casos de prueba ejecutados:</b>	21
<b>Número de incidentes:</b>	0
<b>Número de casos de prueba re-ejecutados:</b>	21
<b>Número de incidentes resueltos:</b>	0

### 3.2 Pruebas de Seguridad

Para la ejecución de estas pruebas se utilizó el grupo de hilos que contiene las peticiones de las 6 vistas del MGB, desactivando las cookies para que no se envíe el token de sesión y poder validar si se puede acceder a alguna de las vistas del módulo sin haber iniciado sesión.

Todas las peticiones responden correctamente con el código 301 que indica el proceso de redirección a la vista de inicio de sesión como se muestra en la figura 8 en la que además se observa cómo se da el proceso de redirección en 3 fases, siendo el último la respuesta de la vista de inicio de sesión del SIAAF [véase figura 9]



**Figura 8.** Respuesta de redireccionamiento de la vista listar estadísticas Becas al intentar acceder sin token de sesión [captura de pantalla]



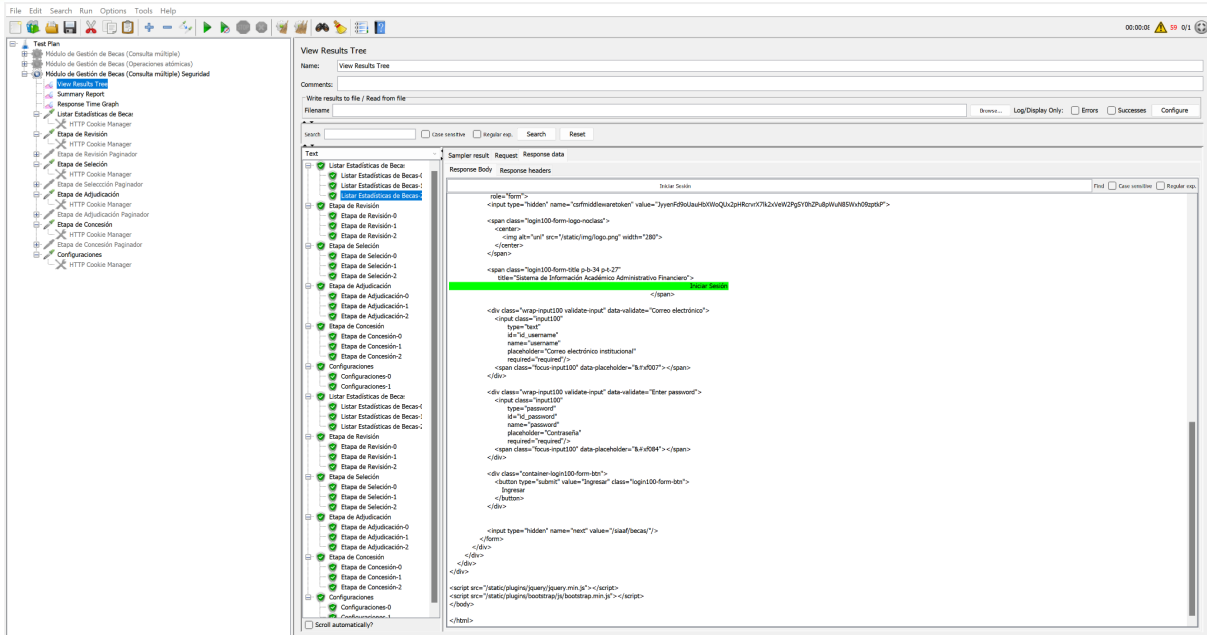


Figura 9. Respuesta 200 con la vista de inicio de sesión del SIAAF [captura de pantalla]

Para validar los permisos de acceso que tienen los diferentes grupos del módulo se intentó acceder a vistas en las que el usuario de Tesorería no tiene permisos como se muestra en la figura 10 [para una ejecución detallada de este intento de pruebas véase **Anexos sección Pruebas de Seguridad**].

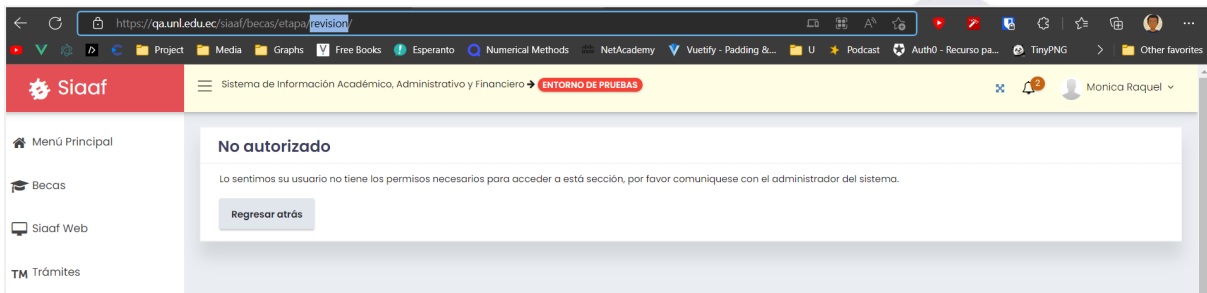


Figura 10. Intento de acceso a la Etapa de Revisión con el rol de Tesorería [captura de pantalla]

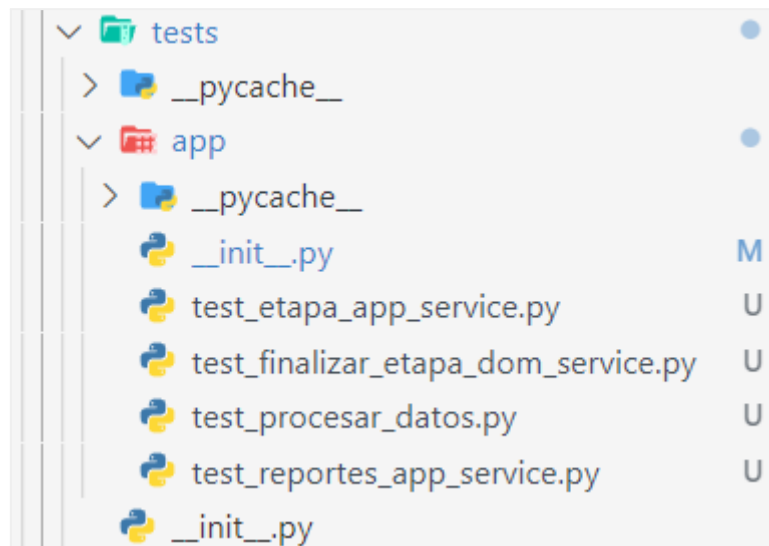
<b>Número de casos de prueba ejecutados:</b>	13
<b>Número de incidentes:</b>	0
<b>Número de casos de prueba re-ejecutados:</b>	6
<b>Número de incidentes resueltos:</b>	0

### 3.3 Pruebas Unitarias

En la ejecución de las Pruebas Unitarias se utilizó la librería **SimpleTestCase** incluida en Django que permite crear escenarios de prueba del código, para simular que las distintas combinaciones de entrada produzcan una salida

1859 deseada y corregir errores en caso de ser necesario. En esta validación se consideraron aquellas funcionalidades que son más críticas para el módulo, es decir, que de fallar podría producir errores del uso de varias características.

El código para las Pruebas Unitarias del módulo de Gestión de Becas fue organizado de acuerdo a las capas más importantes del módulo, dentro de un directorio tests en la aplicación de becas.



**Figura 11.** Estructura del código de Pruebas Unitarias para el Módulo de Gestión de Becas [captura de pantalla]

Para validar las distintas funciones que tiene el MGB se crearon Pruebas Unitarias específicas para cada función con algunas de las variaciones que se pueden presentar; por ejemplo para el caso de la función que obtiene los meses de un periodo académico se crearon 2 funciones test, la primera para validar que para los periodos ABR-SEP obtenga siempre los meses de mayo a agosto y la segunda para validar que para los periodos OCT-ABR se obtengan los meses de octubre a marzo [véase figura 12].

```
unit_tests.py

@classmethod
def test_get_meses_abr_sep(self):
    meses_obtenidos = EtapasAppService.get_listado_meses_por_periodo_academico(
        'ABR21-SEP21')
    meses_esperados = ['may', 'jun', 'jul', 'ago']
    print('Debería retornar los meses de Mayo a Agosto:')
    print('meses obtenidos: ', meses_obtenidos)
    print('meses esperados: ', meses_esperados)
    assert meses_obtenidos == meses_esperados

@classmethod
def test_get_meses_oct_mar(self):
    meses_obtenidos = EtapasAppService.get_listado_meses_por_periodo_academico(
        'OCT21-ABR22')
    meses_esperados = ['oct', 'nov', 'dic', 'ene', 'feb', 'mar']
    print('Debería retornar los meses de Octubre a Marzo:')
    print('meses obtenidos: ', meses_obtenidos)
    print('meses esperados: ', meses_esperados)
    assert meses_obtenidos == meses_esperados
```

**Figura 12.** Código de las Pruebas Unitarias al método `get_listado_meses_por_periodo_academico` [captura de pantalla]

Para ejecutar los tests se hizo uso del comando `python manage.py test app.becas.tests.app.TestEtapaAppService`, y para comprobar que efectivamente se obtiene el resultado esperado se hizo un `print` con una descripción del caso de prueba así como del resultado obtenido y el resultado esperado de cada test como se aprecia en la figura 13.

```
(siaafenv) jack_@Legion:~/UNL/siaaf$ python manage.py test
app.becas.tests.app.TestEtapaAppService
System check identified no issues (0 silenced).
Debería retornar los meses de Mayo a Agosto:
meses obtenidos: ['may', 'jun', 'jul', 'ago']
meses esperados: ['may', 'jun', 'jul', 'ago']
.Debería retornar los meses de Octubre a Marzo:
meses obtenidos: ['oct', 'nov', 'dic', 'ene', 'feb', 'mar']
meses esperados: ['oct', 'nov', 'dic', 'ene', 'feb', 'mar']
.
-----
Ran 2 tests in 0.001s

OK
```

**Figura 13.** Respuesta por consola de la ejecución de las Pruebas Unitarias al método `get_listado_meses_por_periodo_academico` [captura de pantalla]

Para la finalización de Etapas se creó código de Pruebas Unitarias [véase Figura 43, Figura 44, Figura 45 y Figura 46 de los anexos] para validar las funciones más importantes como la aprobación y rechazo de postulantes, la creación de instancias de becas periodo de la siguiente etapa y la verificación de que el estado de los postulantes sea permitido. Estas Pruebas Unitarias se ejecutaron correctamente validando que las funciones retornan lo esperado [véase figura 14].

```
(siaafenv) jack_@Legion:~/UNL/siaaf$ python manage.py test
app.becas.tests.app.TestFinalizarEtapaDomService
System check identified no issues (0 silenced).
Si el postulante fue aprobado debería retornar True
es_beca_aprobada: True
.Si el postulante fue rechazado debería retornar False
es_beca_aprobada: False
.Debería retornar un dict sin mensaje y false si el estado es APROBADO
el estado a comprobar es: 2
respuesta: (False, '', None)
.Debería retornar un dict con un mensaje de advertencia si el estado es EN_REVISION
el estado a comprobar es: 1
respuesta: (True, 'No se puede finalizar la etapa, complete la información para todos
los postulantes', 400)
.Debería retornar un dict con un mensaje de advertencia si el estado es PENDIENTE
el estado a comprobar es: 0
respuesta: (True, 'No se puede finalizar la etapa, complete la información para todos
los postulantes', 400)
.Debería retornar un dict sin mensaje y false si el estado es RECHAZADO
el estado a comprobar es: 3
respuesta: (False, '', None)
.Debería crear un nuevo registro de beca_periodo_academico_etapa con la siguiente etapa
beca_siguiete_etapa: ETAPA_SELECCION
.
-----
Ran 7 tests in 0.002s

OK
```

**Figura 14.** Respuesta por consola de la ejecución de 7 Pruebas Unitarias a los métodos de EtapaDomService [captura de pantalla]

Se crearon Pruebas Unitarias para validar la obtención de montos de pago para la Generación de Reportes de Concesión [véase figura 47] y los resultados de su ejecución se muestran en la figura 15.

```
(siaafenv) jack_@Legion:~/UNL/siaaf$ python manage.py test
app.becas.tests.app.TestReportesAppService
System check identified no issues (0 silenced).
Si el monto no existe debería retornar un monto de 0
monto obtenido: 0
.Si el monto existe debería retornar el monto guardado
monto obtenido: 45
```



```
-----  
Ran 2 tests in 0.001s
```

```
OK
```

**Figura 15.** Respuesta por consola de la ejecución de las Pruebas Unitarias a los métodos de ReportesAppService [captura de pantalla]

También se crearon Pruebas Unitarias para validar la Importación de Registros de Devengado y Pago en la Etapa de Concesión [véase figura 48, figura 49, figura 50, figura 51, figura 52 y figura 53], ya que es una de las funcionalidades más importantes del MGB y que contiene varias validaciones que deben cumplirse para una importación exitosa. La ejecución de estas Pruebas Unitarias presentó errores en ciertas funciones [véase figura 16], que permitieron corregir las funciones gracias a los caso de prueba que no fueron contemplados [véase figura 17].

```
(siaafenv) jack_@Legion:~/UNL/siaaf$ python manage.py test  
app.becas.tests.app.TestProcesarDatosImportacion  
System check identified no issues (0 silenced).  
----- Probando es_dni_valido -----  
Si recibe un número de cédula debería retornar True  
dni a validar: 1105740979  
respuesta: True  
.----- Probando str_a_list -----  
Si la celda contiene texto con meses separados por comas debería retornar una lista con  
los meses  
celda: ene, feb, mar  
respuesta: ['ene', 'feb', 'mar']  
.----- Probando str_a_list -----  
Si la celda está vacía debería retornar una lista vacía  
celda:  
respuesta: []  
F----- Probando es_dni_valido -----  
Si recibe un dni invalido debería retornar False  
dni a validar: 1234567  
respuesta: False  
.----- Probando es_expediente_valido -----  
Si el expediente contiene 1 carácter debería retornar False  
expediente a validar: 1  
respuesta: False  
.----- Probando es_expediente_valido -----  
Si el expediente contiene más de 23 caracteres debería retornar False  
expediente a validar: EXP12345678901234567890123  
respuesta: False  
.----- Probando es_expediente_valido -----  
Si el expediente contiene más de 1 caracter debería retornar True  
expediente a validar: EXP123  
respuesta: True  
.----- Probando hay_meses -----  
Si la lista contiene elementos vacíos debería retornar False
```



1859

```
texto en la lista: ['']
respuesta: True
F----- Probando hay_meses -----
Si la lista contiene meses debería retornar True
texto en la lista: ['oct', 'nov', 'dic']
respuesta: True
.----- Probando hay_meses -----
Si la lista está vacía debería retornar False
texto en la lista: []
respuesta: False
.Si el monto contiene caracteres no numéricos debería retornar False
E----- Probando es_monto_valido -----
Si el monto es 0 debería retornar False
monto a validar: 0
respuesta: False
.----- Probando es_monto_valido -----
Si el monto está vacío debería retornar False
E----- Probando es_monto_valido -----
Si el monto es un número entero debería retornar True
monto a validar: 100
respuesta: True
.----- Probando es_dni_valido -----
Si recibe un número de pasaporte debería retornar True
dni a validar: A3705348
respuesta: False
F----- Probando es_dni_valido -----
Si recibe un número de ruc debería retornar True
dni a validar: 1105740979001
respuesta: True
.----- Probando deberia_validarse_expediente -----
Si el tipo de registro es desconocido debería retornar True
tipo de registro a importar: desconocido
respuesta: True
.----- Probando deberia_validarse_expediente -----
Si el tipo de registro es devengado debería retornar True
tipo de registro a importar: devengado
respuesta: True
.----- Probando deberia_validarse_expediente -----
Si el tipo de registro es pagado debería retornar False
tipo de registro a importar: pagado
respuesta: False
.
=====
ERROR: test_monto_caracteres_no_numericos
(app.becas.tests.app.test_procesar_datos.TestProcesarDatosImportacion)
-----
Traceback (most recent call last):
  File "/home/jack/_UNL/siaaf/app/becas/tests/app/test_procesar_datos.py", line 109, in
test_monto_caracteres_no_numericos
    monto_valido = es_monto_valido(monto)
  File "/home/jack/_UNL/siaaf/app/becas/utils/procesar_datos.py", line 37, in
es_monto_valido
    monto = float(monto) or 0
```





1859

```
ValueError: could not convert string to float: '50a#'

=====
ERROR: test_monto_vacio
(app.becas.tests.app.test_procesar_datos.TestProcesarDatosImportacion)
-----
Traceback (most recent call last):
  File "/home/jack_/UNL/siaaf/app/becas/tests/app/test_procesar_datos.py", line 101, in
test_monto_vacio
    monto_valido = es_monto_valido(monto)
  File "/home/jack_/UNL/siaaf/app/becas/utils/procesar_datos.py", line 37, in
es_monto_valido
    monto = float(monto) or 0
ValueError: could not convert string to float: ''

=====
FAIL: test_celda_meses_vacia_a_lista
(app.becas.tests.app.test_procesar_datos.TestProcesarDatosImportacion)
-----
Traceback (most recent call last):
  File "/home/jack_/UNL/siaaf/app/becas/tests/app/test_procesar_datos.py", line 184, in
test_celda_meses_vacia_a_lista
    self.assertEqual(meses, [])
AssertionError: Lists differ: [''] != []

First list contains 1 additional elements.
First extra element 0:
''

- ['']
+ []

=====
FAIL: test_lista_contiene_elementos_vacios
(app.becas.tests.app.test_procesar_datos.TestProcesarDatosImportacion)
-----
Traceback (most recent call last):
  File "/home/jack_/UNL/siaaf/app/becas/tests/app/test_procesar_datos.py", line 77, in
test_lista_contiene_elementos_vacios
    self.assertFalse(existen_meses)
AssertionError: True is not false

=====
FAIL: test_pasaporte
(app.becas.tests.app.test_procesar_datos.TestProcesarDatosImportacion)
-----
Traceback (most recent call last):
  File "/home/jack_/UNL/siaaf/app/becas/tests/app/test_procesar_datos.py", line 50, in
test_pasaporte
    self.assertTrue(dni_validado)
AssertionError: False is not true

-----
```



1859

```
Ran 19 tests in 0.003s  
FAILED (failures=3, errors=2)  
(siaafenv) jack_@Legion:~/UNL/siaaf$
```

**Figura 16.** Resultado de la ejecución de Pruebas Unitarias a las funciones para procesar datos de importación con 14 aciertos, 3 fallos y 2 errores de 19 pruebas [resultado por consola]

Luego de las correcciones realizadas a las funciones que presentaron inconvenientes con casos de prueba no contemplados, se volvió a ejecutar las Pruebas Unitarias para validar que ya no se presenten los errores detectados, asegurando estos casos alternativos que pudieran llegar a presentarse.

```
(siaafenv) jack_@Legion:~/UNL/siaaf$ python manage.py test  
app.becas.tests.app.TestProcesarDatosImportacion  
System check identified no issues (0 silenced).  
----- Probando es_dni_valido -----  
Si recibe un número de cédula debería retornar True  
dni a validar: 1105740979  
respuesta: True  
----- Probando str_a_list -----  
Si la celda contiene texto con meses separados por comas debería retornar una lista con los meses  
celda: ene, feb, mar  
respuesta: ['ene', 'feb', 'mar']  
----- Probando str_a_list -----  
Si la celda esta vacia debería retornar una lista vacia  
celda:  
respuesta: []  
----- Probando es_dni_valido -----  
Si recibe un dni invalido debería retornar False  
dni a validar: 1234567  
respuesta: False  
----- Probando es_expediente_valido -----  
Si el expediente contiene 1 caracter debería retornar False  
expediente a validar: 1  
respuesta: False  
----- Probando es_expediente_valido -----  
Si el expediente contiene más de 23 caracteres debería retornar False  
expediente a validar: EXP12345678901234567890123  
respuesta: False  
----- Probando es_expediente_valido -----  
Si el expediente contiene más de 1 caracter debería retornar True  
expediente a validar: EXP123  
respuesta: True  
----- Probando hay_meses -----  
Si la lista contiene elementos vacios debería retornar False  
texto en la lista: ['']  
respuesta: False  
----- Probando hay_meses -----  
Si la lista contiene meses debería retornar True  
texto en la lista: ['oct, nov, dic']  
respuesta: True
```





1859

```
.----- Probando hay_meses -----  
Si la lista esta vacia debería retornar False  
texto en la lista: []  
respuesta: False  
.Si el monto contiene caracteres no numéricos debería retornar False  
monto a validar: 50a#  
respuesta: False  
.----- Probando es_monto_valido -----  
Si el monto es 0 debería retornar False  
monto a validar: 0  
respuesta: False  
.----- Probando es_monto_valido -----  
Si el monto está vacío debería retornar False  
monto a validar:  
respuesta: 0  
.----- Probando es_monto_valido -----  
Si el monto es un número entero debería retornar True  
monto a validar: 100  
respuesta: True  
.----- Probando es_dni_valido -----  
Si recibe un número de pasaporte debería retornar True  
dni a validar: A3705348  
respuesta: True  
.----- Probando es_dni_valido -----  
Si recibe un número de ruc debería retornar True  
dni a validar: 1105740979001  
respuesta: True  
.----- Probando deberia_validarse_expediente -----  
Si el tipo de registro es desconocido debería retornar True  
tipo de registro a importar: desconocido  
respuesta: True  
.----- Probando deberia_validarse_expediente -----  
Si el tipo de registro es devengado debería retornar True  
tipo de registro a importar: devengado  
respuesta: True  
.----- Probando deberia_validarse_expediente -----  
Si el tipo de registro es pagado debería retornar False  
tipo de registro a importar: pagado  
respuesta: False  
.-----  
Ran 19 tests in 0.002s  
  
OK
```

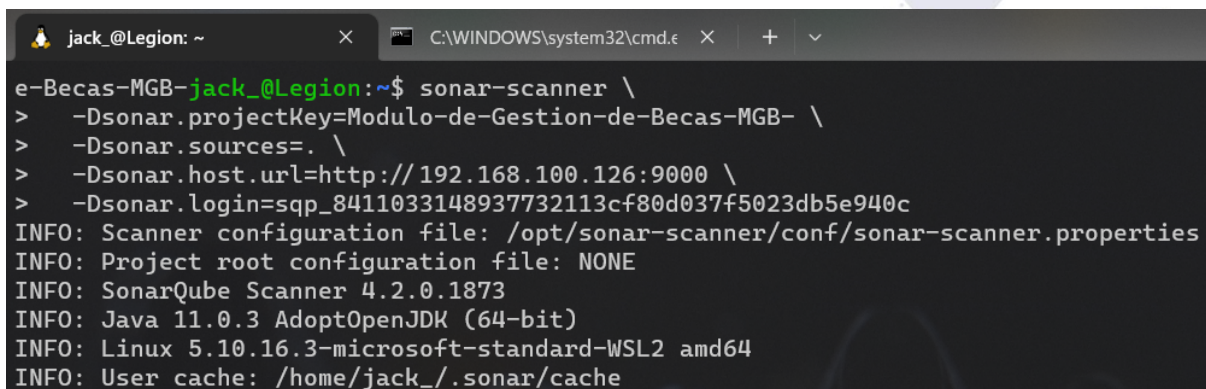
**Figura 17.** Resultado de la re-ejecución de Pruebas Unitarias a las funciones para procesar datos de importación con todas las pruebas validadas [respuesta por consola]

<b>Número de casos de prueba ejecutados:</b>	30
<b>Número de incidentes:</b>	5
<b>Número de casos de prueba re-ejecutados:</b>	19
<b>Número de incidentes resueltos:</b>	5

### 3.4 Pruebas de Código

Para realizar estas pruebas se hizo uso de la herramienta [SonarQube](#) que permite analizar el código de un proyecto de desarrollo. SonarQube se encarga de analizar el código del proyecto en **búsqueda de errores** complicados para evitar un comportamiento indefinido que afecten a los usuarios finales, **detectar vulnerabilidades** que comprometan a la aplicación, **asegurar que el código se mantenga limpio y mantenible**.

El desarrollo del proyecto de Becas se trabajó dentro de un entorno Windows Subsystem for Linux (WSL) ya que fue necesario ejecutar ciertos servicios que se encuentran únicamente disponibles en Linux. Para hacer uso de la herramienta de diagnóstico, se instaló y ejecutó [SonarScanner](#) dentro del subsistema linux para que detecte el proyecto [véase figura 18]. Mientras que SonarQube se instaló en Windows para acceder al servidor local de SonarQube [véase figura 19].

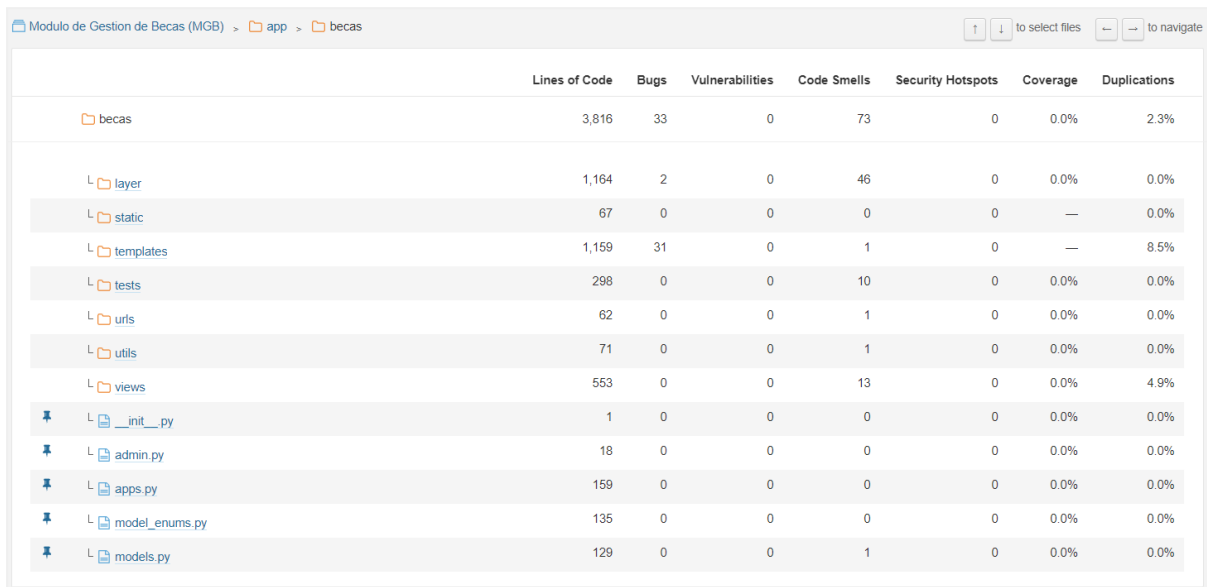


```
jack_@Legion: ~
C:\WINDOWS\system32\cmd.exe
e-Becas-MGB-jack_@Legion:~$ sonar-scanner \
> -Dsonar.projectKey=Modulo-de-Gestion-de-Becas-MGB- \
> -Dsonar.sources=. \
> -Dsonar.host.url=http://192.168.100.126:9000 \
> -Dsonar.login=sqp_8411033148937732113cf80d037f5023db5e940c
INFO: Scanner configuration file: /opt/sonar-scanner/conf/sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarQube Scanner 4.2.0.1873
INFO: Java 11.0.3 AdoptOpenJDK (64-bit)
INFO: Linux 5.10.16.3-microsoft-standard-WSL2 amd64
INFO: User cache: /home/jack_/.sonar/cache
```

**Figura 18.** Ejecución de SonarScanner en Ubuntu a través de WSL para el proyecto SIAAF [captura de pantalla]

Los resultados arrojados por la herramienta muestran que el MGB **contenía 33 bugs en total**, de los cuales 31 [véase figura 54] se refieren al uso no recomendado de etiquetas `<i>`, esto es debido a que estas etiquetas son usadas por [Bootstrap](#) para crear iconos de [FontAwesome](#) por lo que corresponde a bug menores. Por otro lado los 2 restantes se refieren a una asignación de variable innecesaria y a una duplicidad en la validación de sub-expresiones en un if [véase figura 57] que se corrigieron gracias a la detección [véase figura 58].

El MGB cuenta con **cero vulnerabilidades detectadas y cero puntos de acceso** que comprometan la seguridad [véase figura 19]. En cuanto a **code smells** que es un término del inglés para referirse a esas porciones de código que funcionan pero que impactan a primera vista; el MGB tiene 73 de ellos [véase Figura 20]. Por último en cuanto a **duplicidad de código** el MGB representa un 2.3% de código repetido de entre todos los módulos existentes en el SIAAF.



	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
becas	3,816	33	0	73	0	0.0%	2.3%
layer	1,164	2	0	46	0	0.0%	0.0%
static	67	0	0	0	0	—	0.0%
templates	1,159	31	0	1	0	—	8.5%
tests	298	0	0	10	0	0.0%	0.0%
urls	62	0	0	1	0	0.0%	0.0%
utils	71	0	0	1	0	0.0%	0.0%
views	553	0	0	13	0	0.0%	4.9%
__init__.py	1	0	0	0	0	0.0%	0.0%
admin.py	18	0	0	0	0	0.0%	0.0%
apps.py	159	0	0	0	0	0.0%	0.0%
model_enums.py	135	0	0	0	0	0.0%	0.0%
models.py	129	0	0	1	0	0.0%	0.0%

**Figura 19.** Vista del Módulo de Becas con los resultados del análisis de SonarScanner en el servidor local de SonarQube [captura de pantalla]

Para corregir los problemas detectados por la herramienta se revisó cuáles eran las recomendaciones sugeridas para proceder con la implementación. Una de las recomendaciones sugeridas por la herramienta fue **refactorizar el código** de la finalización de etapas para **reducir la complejidad cognitiva** [véase figura 20], con el fin de hacer el código más legible y comprensible para los desarrolladores.

Para ello **se trasladó toda la lógica de la vista a la capa de aplicación** [véase Figura 56] **y dominio** [véase Figura 55] según corresponda, creando métodos más pequeños y que se puedan entender con facilidad gracias a la descripción de cada uno de ellos. Con esto se crearon funciones específicas para cada acción haciendo que el código pase estar interconectado entre 3 archivos.

Se utilizó herencia de clases entre las capas de dominio y aplicación para **heredar todos los atributos y métodos** de la clase dominio (padre) a la clase aplicación (hijo) esto con la finalidad de acceder desde la capa aplicación a la aplicación de dominio y a su vez hacer un llamado desde la vista a la capa aplicación que se encargará de gestionar toda la lógica de la Finalización de Etapas [véase Figura 21].

En cuanto a la **duplicidad de código** el porcentaje más alto se encontraba en los templates [véase Figura 59 y Figura 60] por lo que para reducir el código repetido presente entre el template de la etapa de revisión y el template de la etapa de selección, y los demás templates de las etapas; **se creó un template común para todas las etapas** de la cuál pudieran extender los demás templates de cada etapa [véase Figura 61, Figura 62, Figura 63, Figura 64 y Figura 65].

Además para reducir las líneas de código **se crearon template\_tags y modificar las funciones de la capa de aplicación de las etapas** para delegar la construcción de objetos json al backend [véase Figura 67 y Figura 68], optimizando la carga de los templates en el navegador.

```
298 | @login_required
299 | @require_http_methods(["POST"])
300 | = ⚠ def finalizar_etapa(request: HttpRequest):
    |
    | Refactor this function to reduce its Cognitive Complexity from 16 to the 15 allowed. Why is this a
    | ⚠ Code Smell 🔴 Critical 🔵 Open 🔵 Not assigned 🔵 6min effort Comment
    |
301 | =     try:
302 | =         if request.method == 'POST':
303 | =             # inicializar variables
304 | =             data = json.load(request)['json_data']
305 | =             etapa_codigo = data['etapa_codigo']
306 | =             periodo_academico_id = data['periodo_academico']
307 | =             # obtener etapa actual y siguiente etapa
308 | =             etapa = Etapa.objects.get(
309 | =                 codigo=etapa_codigo)
310 | =             siguiente_etapa = Etapa.objects.get(
311 | =                 codigo=etapa.codigo_siguiente_etapa)
312 | =             (finalizada, mensaje, estado) = FinalizarEtapaDomService.comprobar_estado_etapa(
313 | =                 etapa, periodo_academico_id)
314 | =             if finalizada:
315 | =                 return get_response(mensaje=mensaje, status=estado)
316 | =             # obtener becas del presente periodo academico etapa
317 | =             becas_periodo_academico_etapa = BecaPeriodoAcademicoEtapa.objects.filter(
318 | =                 etapa_id=etapa.id, periodo_academico_id=int(periodo_academico_id))
319 | =             # crear registros en la siguiente etapa
320 | =             listado_becas_siguiente_etapa = []
321 | =             for beca_etapa in becas_periodo_academico_etapa:
322 | =                 (no_permitido, mensaje, estado) = FinalizarEtapaDomService.comprobar_estado_permitido(
323 | =                     beca_etapa.estado)
324 | =                 if no_permitido:
325 | =                     return get_response(mensaje=mensaje, status=estado)
326 | =                 esta_aprobada = FinalizarEtapaDomService.beca_esta_aprobada(
327 | =                     beca_etapa, etapa)
328 | =                 if esta_aprobada:
329 | =                     listado_becas_siguiente_etapa.append(
330 | =                         FinalizarEtapaDomService.crear_beca_siguiente_etapa(becca_etapa, siguiente_etapa))
331 | =
332 | =             with transaction.atomic():
333 | =                 # guardar registros en la siguiente etapa
334 | =                 lista_becas_periodo_academico_nueva_etapa = BecaPeriodoAcademicoEtapa.objects.bulk_create(
335 | =                     listado_becas_siguiente_etapa)
336 | =                 # guardar notificaciones si se finalizo la etapa de seleccion
337 | =                 if siguiente_etapa.codigo == Constantes.ETAPA_ADJUDICACION.value:
338 | =                     EtapasAppService.notificar_nuevos_becados(
339 | =                         lista_becas_periodo_academico_nueva_etapa)
340 | =
341 | =             (finalizada, mensaje, estado) = FinalizarEtapaDomService.finalizar_etapa(
342 | =                 etapa.id, periodo_academico_id)
343 | =             if finalizada:
344 | =                 return get_response(mensaje=mensaje, status=estado)
345 | =
346 | =             return JsonResponse('No se pudo finalizar la Etapa')
347 | =         except Exception as e:
348 | =             return get_response(mensaje=e.args[0], status=400)
349 | =     jacks...
```

**Figura 20.** Recomendación de refactorizar la función de Finalizar Etapas para reducir su complejidad cognitiva [captura de pantalla]

```
siiaaf > app > becas > views > etapas.py > finalizar_etapa

297 @login_required
298 @require_http_methods(["POST"])
299 def finalizar_etapa(request: HttpRequest):
300     try:
301         if request.method == 'POST':
302             # inicializar variables
303             data = json.load(request)['json_data']
304             etapa_codigo = data['etapa_codigo']
305             periodo_academico_id = data['periodo_academico']
306             # obtener etapa actual y siguiente etapa
307             etapa = Etapa.objects.get(
308                 codigo=etapa_codigo)
309             siguiente_etapa = Etapa.objects.get(
310                 codigo=etapa.codigo_siguiente_etapa)
311             (finalizada, mensaje, estado) = FinalizarEtapaDomService.
312                 comprobar_estado_etapa(
313                     etapa, periodo_academico_id)
314             if finalizada:
315                 return get_response(mensaje=mensaje, status=estado)
316             # obtener becas del presente periodo academico etapa
317             becas_periodo_academico_etapa = BecaPeriodoAcademicoEtapa.objects.filter(
318                 etapa_id=etapa.id, periodo_academico_id=int(periodo_academico_id))
319             # crear registros en la siguiente etapa
320             listado_becas_siguiente_etapa = []
321             for beca_etapa in becas_periodo_academico_etapa:
322                 (no_permitido, mensaje, estado) = FinalizarEtapaDomService.
323                     comprobar_estado_permitido(
324                         beca_etapa.estado)
325                 if no_permitido:
326                     return get_response(mensaje=mensaje, status=estado)
327                 esta_aprobada = FinalizarEtapaDomService.beca_esta_aprobada(
328                     beca_etapa, etapa)
329                 if esta_aprobada:
330                     listado_becas_siguiente_etapa.append(
331                         FinalizarEtapaDomService.crear_beca_siguiente_etapa(
332                             beca_etapa, siguiente_etapa))
333             with transaction.atomic():
334                 # guardar registros en la siguiente etapa
335                 lista_becas_periodo_academico_nueva_etapa =
336                     BecaPeriodoAcademicoEtapa.objects.bulk_create(
337                         listado_becas_siguiente_etapa)
338                 # guardar notificaciones si se finalizo la etapa de seleccion
339                 if siguiente_etapa.codigo == Constantes.ETAPA_ADJUDICACION.value:
340                     EtapasAppService.notificar_nuevos_becados(
341                         lista_becas_periodo_academico_nueva_etapa)
342             (finalizada, mensaje, estado) = FinalizarEtapaDomService.
343                 finalizar_etapa(
344                     etapa_id, periodo_academico_id)
345             if finalizada:
346                 return get_response(mensaje=mensaje, status=estado)
347         return JsonResponse("No se pudo finalizar la Etapa")
348     except Exception as e:
349         return get_response(mensaje=e.args[0], status=400)

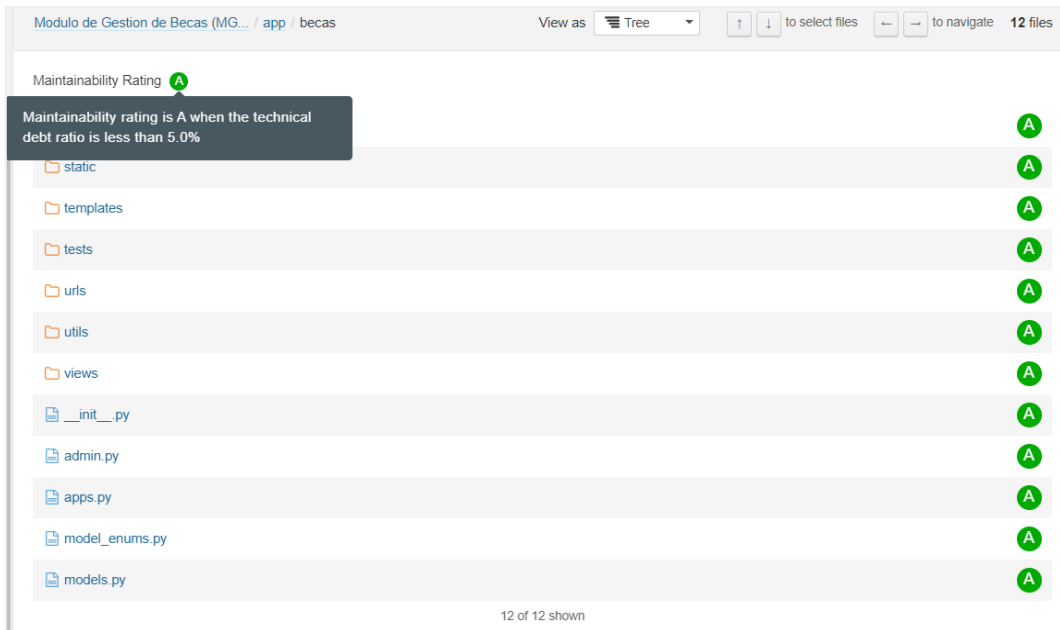
298 @login_required
299 @require_http_methods(["POST"])
300 def finalizar_etapa(request: HttpRequest):
301     try:
302         if request.method == 'POST':
303             data = json.load(request)['json_data']
304             etapa_codigo = data['etapa_codigo']
305             etapa = Etapa.objects.get(codigo=etapa_codigo)
306             periodo_academico_id = int(data['periodo_academico'])
307             finalizar_etapa_app_service = FinalizarEtapasAppService(
308                 etapa, periodo_academico_id)
309             finalizar_etapa_app_service.set_siguiente_etapa()
310             finalizar_etapa_app_service.comprobar_estado_etapa()
311             finalizar_etapa_app_service.set_becas_periodo_academico_etapa()
312             finalizar_etapa_app_service.crear_becas_siguiente_etapa()
313             with transaction.atomic():
314                 finalizar_etapa_app_service.guardar_becas_siguiente_etapa()
315                 etapa_finalizada_ok = finalizar_etapa_app_service.finalizar_etapa()
316                 if etapa_finalizada_ok:
317                     return get_response(
318                         mensaje=FinalizarEtapasAppService.mensaje_etapa_finalizada,
319                         status=status.HTTP_200_OK)
320                 else:
321                     return JsonResponse(FinalizarEtapasAppService.mensaje_etapa_no_finalizada)
322             except Exception as e:
323                 return get_response(mensaje=e.args[0], status=status.HTTP_400_BAD_REQUEST)

324 return JsonResponse(FinalizarEtapasAppService.mensaje_etapa_no_finalizada)
325 except Exception as e:
326 return get_response(mensaje=e.args[0], status=status.HTTP_400_BAD_REQUEST)
327
328
```

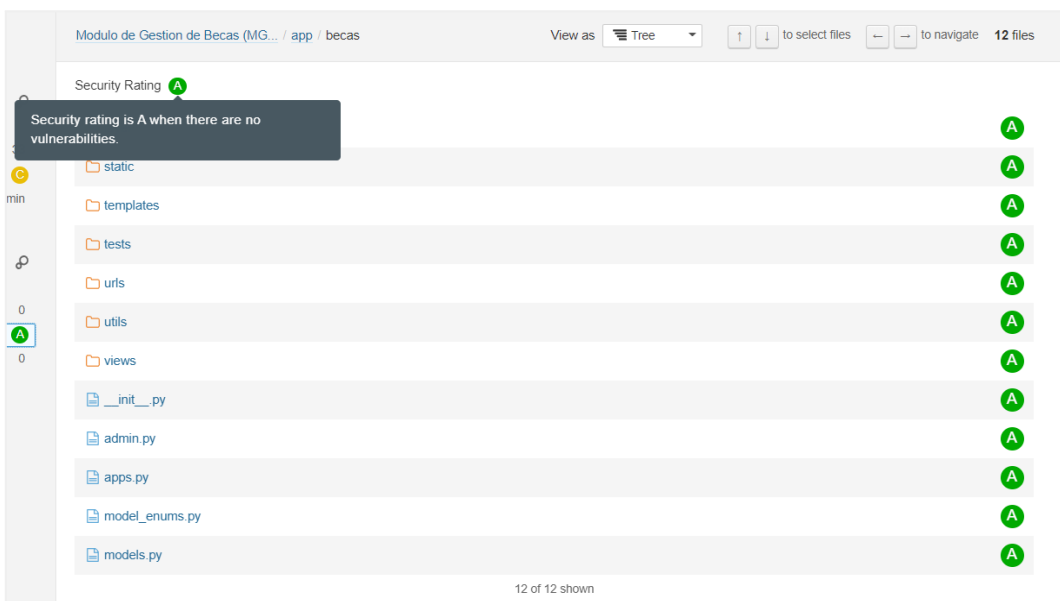
**Figura 21.** Comparativa de cambios antes y después de aplicada la refactorización a la función de Finalizar Etapas para reducir su complejidad cognitiva [captura de pantalla]

Por último cabe destacar que la herramienta SonarQube puntúa con una calificación A (mejor puntuación) en cuanto a mantenibilidad del código y a la seguridad evidenciando que no se han detectado vulnerabilidades graves que afecten a este apartado [véase Figura 22 y Figura 23].





**Figura 22.** Calificación A (mejor puntuación) en la mantenibilidad del código para el MGB [captura de pantalla]



**Figura 23.** Calificación A (mejor puntuación) en la seguridad del código para el MGB [captura de pantalla]

### 3.5 Pruebas de Aceptación del Usuario

Las Pruebas de Aceptación del Usuario se llevaron a cabo en 3 fases:

1. Fase de Socialización del MGB con los Clientes
2. Fase de Pilotaje para los Clientes Directos
3. Fase de Aceptación del MGB por parte de los Clientes



## Fase de Socialización del MGB con los Clientes

La Fase de Socialización del MGB se realizó el día viernes 05 de agosto de 2022, de acuerdo a la agenda prevista que se muestra en el acta de socialización [véase anexo]. Primeramente se aplicó una encuesta [véase [Encuesta de Satisfacción previo al uso del MGB aplicada a UBU](#), [Encuesta de Satisfacción previo al uso del MGB aplicada a Contabilidad Financiera](#), [Encuesta de Satisfacción previo al uso del MGB aplicada a Tesorería Financiera](#)] para obtener información acerca del tiempo que toma realizar algunos de los procesos a cada uno de los Clientes Directos sin el uso del módulo informático. Luego se procedió con la capacitación del personal indicando las funcionalidades y responsabilidades que tendría cada rol de usuario. Durante la capacitación se resolvieron las dudas que iban surgiendo con respecto al uso del módulo informático y posteriormente se abrió un foro de preguntas y respuestas para resolver dudas específicas, una vez cumplido con la capacitación se firmó el acta de asistencia a la capacitación [véase anexo [Acta de Reunión UNL-DTI-2022-002](#)] y se conversó sobre la fase de pilotaje.

## Fase de Pilotaje para los Clientes Directos

La Fase de Pilotaje consistió en el espacio de tiempo que los Clientes Directos tuvieron para probar y validar las funcionalidades desarrolladas del módulo. Esta fase se habilitó desde el día 09 de agosto de 2022 hasta el día 19 de agosto de 2022. En esta fase los participantes tuvieron la oportunidad de probar y validar las funcionalidades del MGB y comprobar que cada uno pueda realizar las acciones de las que son responsables. Durante esta fase se solicitó ayuda en el manejo del MGB por parte del usuario de Tesorería Financiera y el Administrador de Becas; para brindar soporte se organizaron reuniones de asistencia virtual guiada [véase [Asistencia virtual en el uso del MGB al usuario de Tesorería Financiera durante la fase de Pilotaje](#), [Asistencia virtual en el uso del MGB al usuario Administrador de Becas durante la fase de Pilotaje](#)] en la que el usuario fue realizando las acciones indicadas por el tesista para la gestión de becas de acuerdo a las funciones correspondientes..

## Fase de Aceptación del MGB por parte de los Clientes

La Fase de Aceptación se realizó el día 19 de agosto de 2022 mediante una reunión con los Clientes, y de acuerdo a la agenda prevista se aplicó 2 encuestas, la primera [véase [Encuesta de Satisfacción posterior al uso del MGB aplicada a UBU](#), [Encuesta de Satisfacción posterior al uso del MGB aplicada a Contabilidad Financiera](#), [Encuesta de Satisfacción posterior al uso del MGB aplicada a Tesorería Financiera](#)] acerca del tiempo que les toma realizar los mismos procesos que se consultaron durante la Socialización pero esta vez enfocada al uso del MGB, para comparar los resultados obtenidos antes y después del MGB. La segunda encuesta estuvo enfocada en determinar el cumplimiento de los requisitos de



1859 software tanto funcionales como no funcionales para conocer el grado de satisfacción del MGB [véase [Encuesta de Aceptación del MGB aplicada al usuario de la UBU](#), [Encuesta de Aceptación del MGB aplicada al usuario de Contabilidad Financiera](#), [Encuesta de Aceptación del MGB aplicada al usuario de Tesorería Financiera](#)]. El cumplimiento de los requisitos funcionales fue evaluado por parte de los Clientes Directos, mientras que los requisitos no funcionales fueron evaluados por los Clientes Indirectos ya que involucra conocimiento técnico para su correcta evaluación [véase [Encuesta de Aceptación del MGB aplicada al product owner en calidad de representante de la DTI](#), [Encuesta de Aceptación del MGB aplicada al QA en calidad de representante de la DTI](#)].

Para finalizar la fase todos los Clientes firmaron el acta de Aceptación del MGB con los acuerdos y compromisos a los que se llegaron [véase [Acta de Reunión UNL-DTI-2022-003](#)].

Los resultados recopilados durante esta fase son visibles en las tablas de resultados de evaluación basados en los requisitos funcionales y no funcionales. Las tablas de resultados muestran el número identificador de Requisito Funcional y No Funcional respectivamente para su comparación con el documento de la ERS. Se indica el nombre de requisito, las personas que participaron en la evaluación de dicho requerimiento, identificadas con un número único que es utilizado en la columna de evaluación correspondiente para indicar la apreciación proporcionada por cada uno de los participantes en la evaluación mediante las encuestas aplicadas a cada Cliente.

**Tabla 5.** Resultados de la evaluación del MGB en base a los Requisitos Funcionales capturados durante la fase de análisis.

Identificador	Nombre	Evaluadores	Evaluación		
			No cumple	En parte	Cumple
RF01	Sincronizar datos de postulación desde el MGT hacia el MGB	1. Licda. Marlene Valdez 2. Ing. Máximo Alvarez 3. Ing. Johana Cueva			1 2 3
RF02	Actualizar estados de postulaciones	1. Licda. Marlene Valdez 2. Ing. Máximo Alvarez 3. Ing. Johana Cueva			1 2 3





RF03	Descargar documentos de postulación	<ol style="list-style-type: none"><li>1. Licda. Marlene Valdez</li><li>2. Ing. Máximo Alvarez</li><li>3. Ing. Johana Cueva</li></ol>			<ol style="list-style-type: none"><li>1</li><li>2</li><li>3</li></ol>
RF04	Ver trámite en el MGT	<ol style="list-style-type: none"><li>1. Licda. Marlene Valdez</li><li>2. Ing. Máximo Alvarez</li><li>3. Ing. Johanna Cueva</li></ol>			<ol style="list-style-type: none"><li>1</li><li>2</li><li>3</li></ol>
RF05	Finalizar Etapas	<ol style="list-style-type: none"><li>1. Licda. Marlene Valdez</li><li>2. Ing. Máximo Alvarez</li><li>3. Ing. Johana Cueva</li></ol>			<ol style="list-style-type: none"><li>1</li><li>2</li><li>3</li></ol>
RF06	Actualizar estados de Becas	<ol style="list-style-type: none"><li>1. Licda. Marlene Valdez</li><li>2. Ing. Máximo Alvarez</li><li>3. Ing. Johana Cueva</li></ol>			<ol style="list-style-type: none"><li>1</li><li>2</li><li>3</li></ol>
RF07	Visualizar Postulaciones y Becas	<ol style="list-style-type: none"><li>1. Licda. Marlene Valdez</li><li>2. Licda. Mónica Mora</li><li>3. Ing. Sebastian Varela</li><li>4. Ing. Máximo Alvarez</li><li>5. Ing. Johana Cueva</li></ol>			<ol style="list-style-type: none"><li>1</li><li>2</li><li>3</li><li>4</li><li>5</li></ol>
RF08	Generar Reportes	<ol style="list-style-type: none"><li>1. Licda. Marlene Valdez</li><li>2. Licda. Mónica Mora</li><li>3. Ing. Sebastian Varela</li></ol>			<ol style="list-style-type: none"><li>1</li><li>2</li><li>3</li><li>4</li><li>5</li></ol>



		4. Ing. Máximo Alvarez 5. Ing. Johana Cueva			
RF09	Agregar mensaje frecuente	1. Licda. Marlene Valdez 2. Ing. Máximo Alvarez 3. Ing. Johana Cueva			1 2 3
RF10	Actualizar mensaje frecuente	1. Licda. Marlene Valdez 2. Ing. Máximo Alvarez 3. Ing. Johana Cueva			1 2 3
RF11	Eliminar mensaje frecuente	1. Licda. Marlene Valdez 2. Ing. Máximo Alvarez 3. Ing. Johana Cueva			1 2 3
RF12	Marcar como favorito mensaje frecuente	1. Licda. Marlene Valdez 2. Ing. Máximo Alvarez 3. Ing. Johana Cueva			1 2 3
RF13	Actualizar montos de pago para los tipos de beca	1. Licda. Marlene Valdez 2. Ing. Máximo Alvarez 3. Ing. Johana Cueva			1 2 3
RF14	Visualizar estadísticas generales de becados	1. Licda. Marlene Valdez 2. Licda. Mónica Mora 3. Ing. Sebastian Varela			1 2 3 4 5



		4. Ing. Máximo Alvarez 5. Ing. Johana Cueva			
RF15	Iniciar Sesión	1. Licda. Marlene Valdez 2. Licda. Mónica Mora 3. Ing. Sebastian Varela 4. Ing. Máximo Alvarez 5. Ing. Johana Cueva			1 2 3 4 5
RF16	Notificar resultados de postulación	1. Ing. Máximo Alvarez 2. Ing. Johana Cueva			1 2
RF17	Guardar registro de becado en el sistema financiero	1. Licda. Mónica Mora 2. Ing. Máximo Alvarez 3. Ing. Johana Cueva			1 2 3
RF18	Registrar Pago de Becas	1. Licda. Mónica Mora 2. Ing. Máximo Alvarez 3. Ing. Johana Cueva			1 2 3
<b>OBSERVACIONES:</b>					

**Figura 24.** Resultados en la evaluación del MGB en base a los Requisitos No Funcionales considerados durante la fase de análisis.

Identificador	Nombre	Evaluadores	Evaluación		
			No cumple	En parte	Cumple
RNF02	La información solicitada se puede filtrar, ordenar y se encuentra paginada	1. Ing. Máximo Alvarez 2. Ing. Johana Cueva			1 2



1859

RNF03	El MGB no requiere actualizar su interfaz para interactuar con la información	<ol style="list-style-type: none"> <li>1. Ing. Máximo Alvarez</li> <li>2. Ing. Johana Cueva</li> </ol>			1 2
RNF04	El MGB permite la conexión simultánea de varios usuarios con los distintos roles a la vez	<ol style="list-style-type: none"> <li>1. Ing. Máximo Alvarez</li> <li>2. Ing. Johana Cueva</li> </ol>			1 2
RNF05	El MGB permite el acceso únicamente a los grupos de usuarios que se definieron para interactuar con el módulo y a los superadministradores del SIAAF	<ol style="list-style-type: none"> <li>1. Ing. Máximo Alvarez</li> <li>2. Ing. Johana Cueva</li> </ol>			1 2
RNF06	El MGB hace una distinción entre los permisos que tiene cada grupo de usuario de acuerdo a las funcionalidades que les corresponden	<ol style="list-style-type: none"> <li>1. Ing. Máximo Alvarez</li> <li>2. Ing. Johana Cueva</li> </ol>			1 2
RNF07	El MGB hace uso de la autenticación del SIAAF	<ol style="list-style-type: none"> <li>1. Ing. Máximo Alvarez</li> <li>2. Ing. Johana Cueva</li> </ol>			1 2
RNF08	La seguridad del MGB está cubierta por la arquitectura provista por el SIAAF	<ol style="list-style-type: none"> <li>1. Ing. Máximo Alvarez</li> <li>2. Ing. Johana Cueva</li> </ol>			1 2
RNF09	Las tablas de base de datos del MGB serán respaldadas junto con la base de datos del SIAAF	<ol style="list-style-type: none"> <li>1. Ing. Máximo Alvarez</li> <li>2. Ing. Johana Cueva</li> </ol>			1 2
RNF10	El MGB una vez puesto en producción estará disponible en	<ol style="list-style-type: none"> <li>1. Ing. Máximo Alvarez</li> </ol>			1 2



	horarios laborables exceptuando mantenimientos planificados que serán notificados con anticipación	2. Ing. Johana Cueva			
	El MGB sigue los patrones de diseño y sintaxis que están establecidos por el SIAAF hasta la fecha para facilitar su continuo desarrollo	1. Ing. Máximo Alvarez 2. Ing. Johana Cueva			1 2
RNF11	Se realizó la socialización y capacitación al personal que hará uso del MGB	1. Ing. Máximo Alvarez 2. Ing. Johana Cueva			1 2
RNF12	El MGB puede comunicarse con otros módulos si se requiere.	1. Ing. Máximo Alvarez 2. Ing. Johana Cueva			1 2
RNF13	El MGB ofrece un API de comunicación con el MGT para su comunicación.	1. Ing. Máximo Alvarez 2. Ing. Johana Cueva			1 2
RNF14	El MGB está construido con el framework Django como se solicitó.	1. Ing. Máximo Alvarez 2. Ing. Johana Cueva			1 2
RNF15	El MGB es accesible a través de internet.	1. Ing. Máximo Alvarez 2. Ing. Johana Cueva			1 2
<b>OBSERVACIONES:</b>					

Al finalizar la encuesta de Aceptación se les consultó a los Clientes Directos aspectos que no iban tanto de la mano de evaluar los requisitos del módulo sino más bien de conocer la apreciación que tienen del módulo. Para ello se consultó sobre aspectos útiles para cada Cliente Directo, las mejoras a futuro que consideran útiles, su intención de uso, la utilidad que les representa a cada uno



1859 para su gestión, y la factibilidad de uso inmediato. A continuación se detalla toda la información recopilada de las encuestas aplicadas.

### Aspectos que les resultaron más útiles a los usuarios

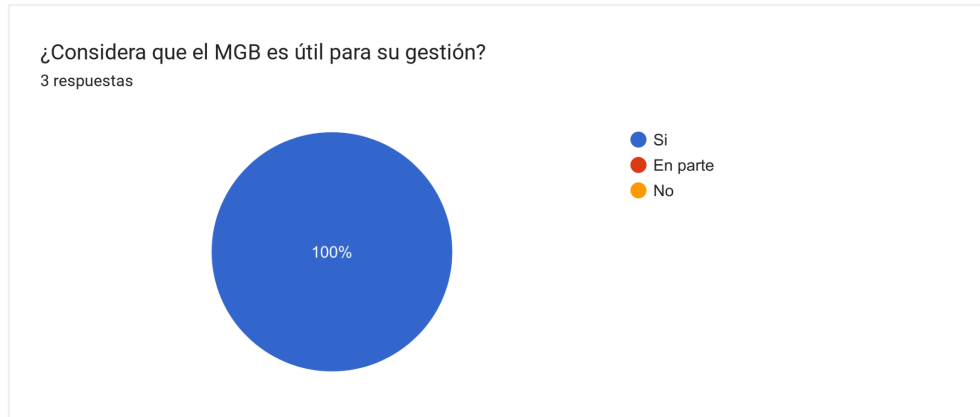
- Administrador de Becas: conocer con precisión el número de becarios por facultades y carreras
- Contabilidad Financiera: generar y cargar en excel la información de los registros de concesión.
- Tesorería Financiera: el registro de becas pagadas para permitir llevar un mejor control en el registro de pagos.
- Administrador del SIAAF: la exportación de un formato y permitir así la carga masiva de los devengados.

### Mejoras a futuro para el Módulo de Gestión de Becas según la opinión de los usuarios

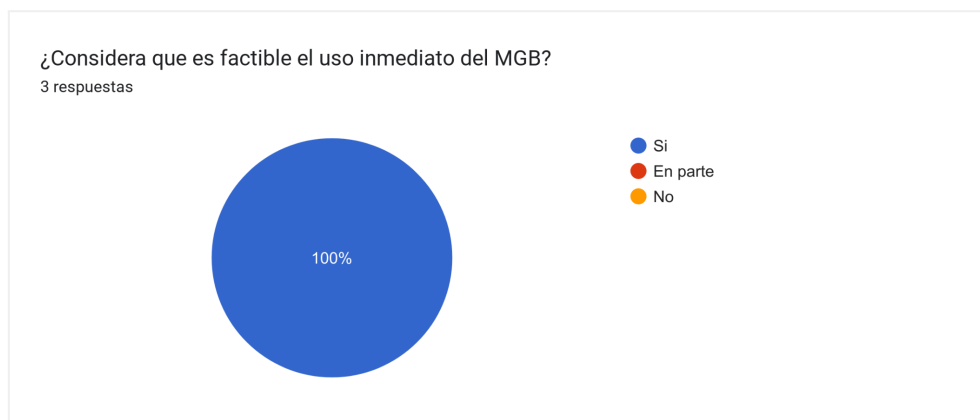
- Administrador de Becas: especificando mucho más los datos de los postulantes especialmente el número de cuenta bancaria, tipo de cuenta y entidad bancaria.
- Contabilidad Financiera: la vinculación con el eSIGEF si fuera posible dicha comunicación.
- Tesorería Financiera: obtener un reporte de los números de cuentas de los beneficiarios para el respectivo ingreso en el eSIGEF
- Administrador del SIAAF: actualizando el Módulo de Gestión de Trámites (MGT) en la fase que queda pendiente, una vez se le haya aprobado la beca a un postulante que fue migrado.

### Intención de uso con el Módulo de Gestión de Becas por cada rol de usuario

- Administrador de Becas: priorizar y dar una información oportuna y en menor tiempo respecto a Becarios.
- Contabilidad Financiera: estar seguros de no realizar 2 pagos sobre un mismo periodo.
- Tesorería Financiera: llevar un control adecuado de los pagos de las becas.
- Administrador del SIAAF: desde mi rol no hago uso directo del módulo



**Figura 25.** Gráfico que muestra el porcentaje de utilidad del MGB para los 3 Clientes Directos encuestados [gráfico]








**Figura 26.** Gráfico que muestra el porcentaje de factibilidad del uso inmediato MGB según los 3 Clientes Directos encuestados [gráfico]

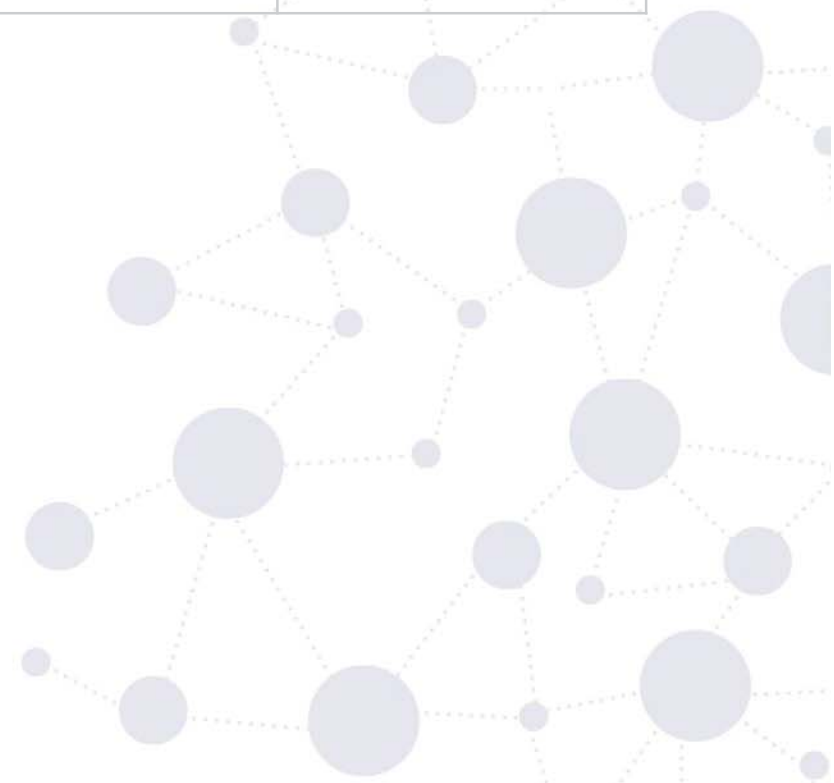
<b>Número de casos de prueba ejecutados:</b>	33
<b>Número de incidentes:</b>	12
<b>Número de casos de prueba re-ejecutados:</b>	33
<b>Número de incidentes resueltos:</b>	12





## FIRMAS DE RESPONSABILIDAD

Acción	Funcionario	Firmas
Elaborado por:	Jackson Andrews Guzmán Tituana <b>Tesista</b>	 Digitally signed by Jackson Guzmán Date: 2022.09.02 15:34:51 -05'00'
Revisado por:	Danny Emanuel Muñoz Flores <b>Especialista en Sistemas de Información</b>	 Firmado electrónicamente por: <b>DANNY EMANUEL MUNOZ FLORES</b>
	Máximo Andres Alvarez Pacheco <b>Analista de Sistemas de Información 1</b>	 Firmado electrónicamente por: <b>MAXIMO ANDRES ALVAREZ PACHECO</b>
	Johana Elizabeth Cueva Dominguez <b>Analista de Sistemas Informáticos 1</b>	 Firmado electrónicamente por: <b>JOHANA ELIZABETH CUEVA DOMINGUEZ</b>
Aprobado por:	Jhon Alexander Calderón Sanmartín <b>Director de Tecnologías de Información</b>	 Firmado electrónicamente por: <b>JHON ALEXANDER CALDERON SANMARTIN</b>





## ANEXOS

### Documento de Especificación de Requisitos de Software para el Módulo de Gestión de Becas

ERS\_IEEE\_830 para el MGB - firmado.pdf

### Pruebas de Casos de Uso

\_QA\_PROYECTO\_2020-007\_MODULO\_GESTIÓN\_BECAS

### Pruebas de Carga y Estabilidad

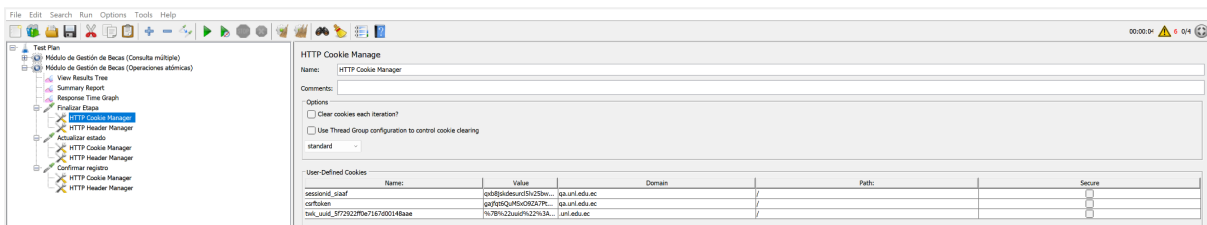


Figura 27. Configuración de Cookies para las peticiones del MGB [captura de pantalla]

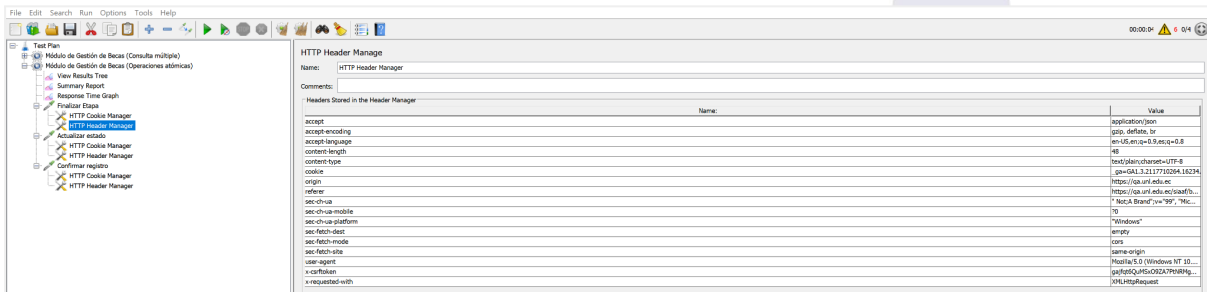


Figura 28. Configuración de las cabeceras para las peticiones del MGB [captura de pantalla]

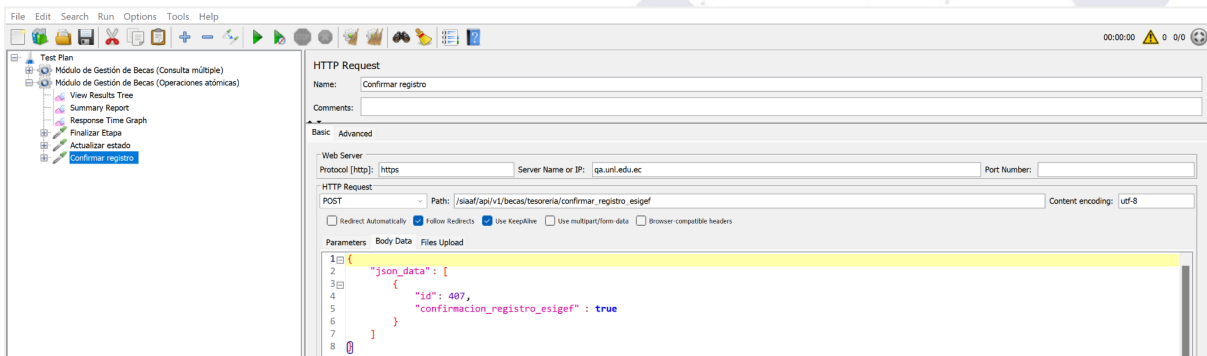


Figura 29. Datos del cuerpo de la petición para la confirmación del registro en el sistema financiero [captura de pantalla]



1859

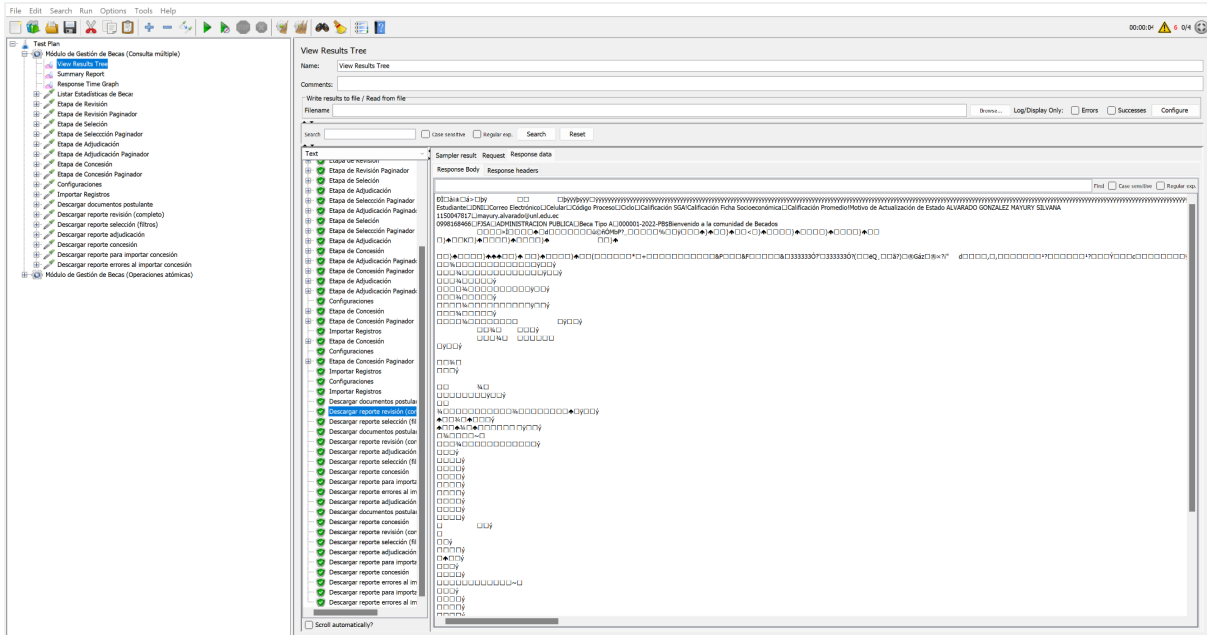


Figura 30. Respuesta de la petición para la descarga del reporte de revisión [captura de pantalla]

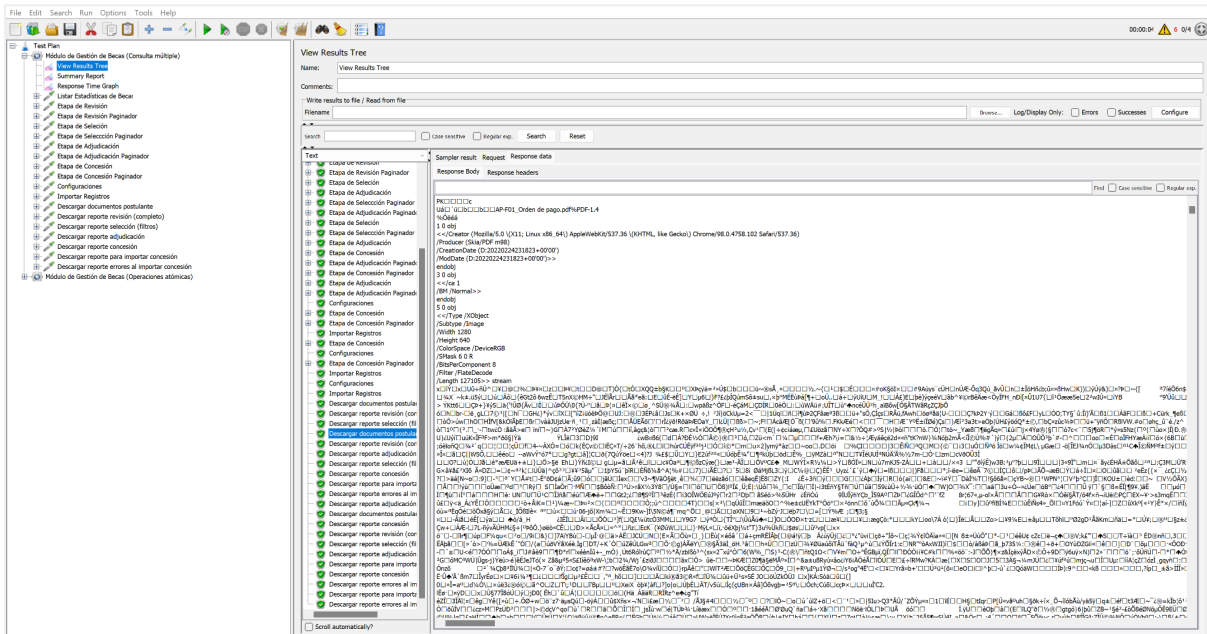


Figura 31. Respuesta de la petición para la descarga de la documentación del postulante [captura de pantalla]

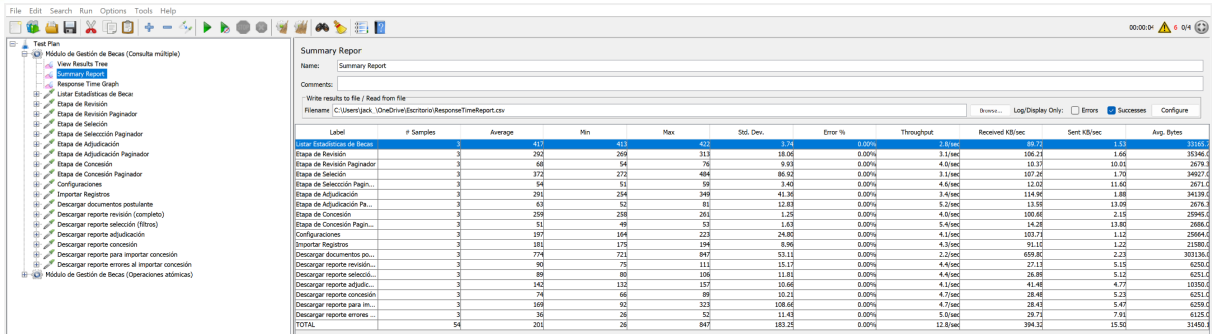


Figura 32. Reporte resumen de los tiempos de ejecución del grupo de hilos de consultas múltiples generado por Apache JMeter [captura de pantalla]

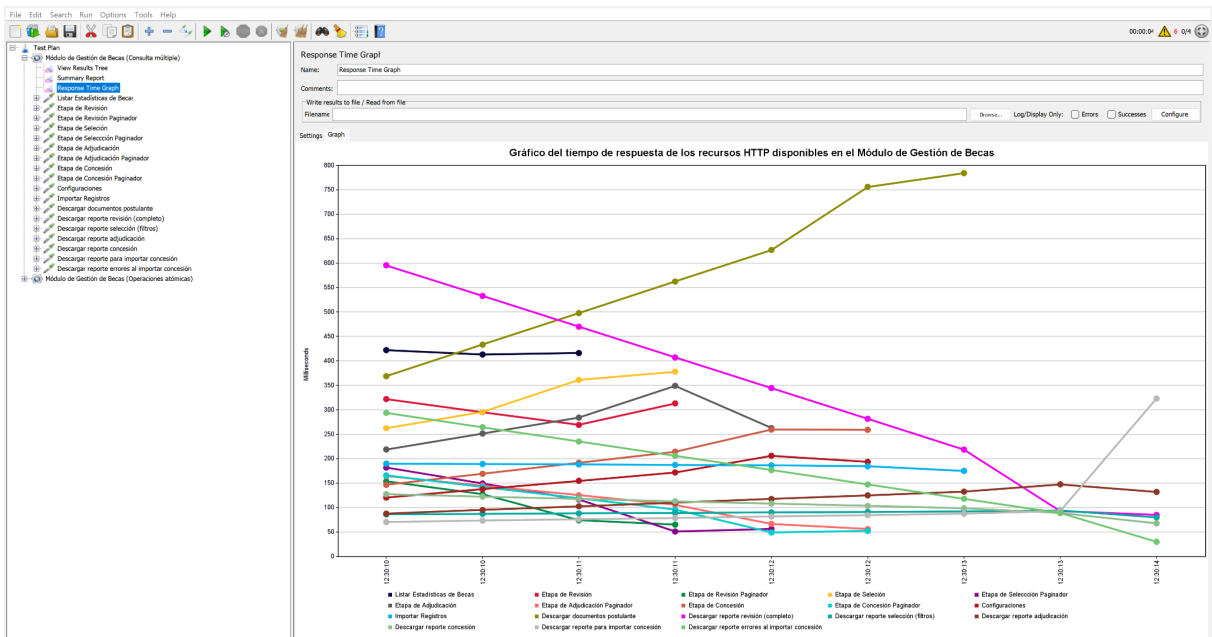


Figura 33. Gráfica del tiempo de respuesta de las peticiones del hilo de consultas múltiples ejecutadas cada 500ms [captura de pantalla]

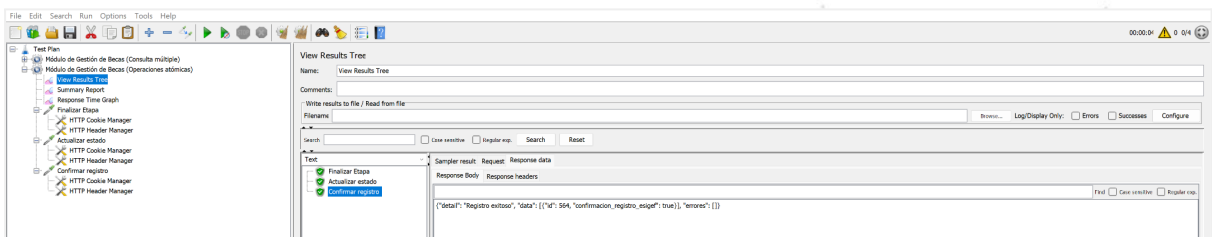
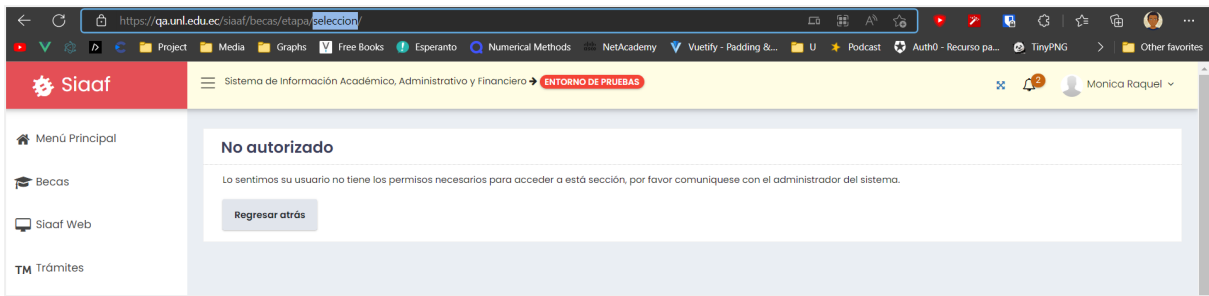


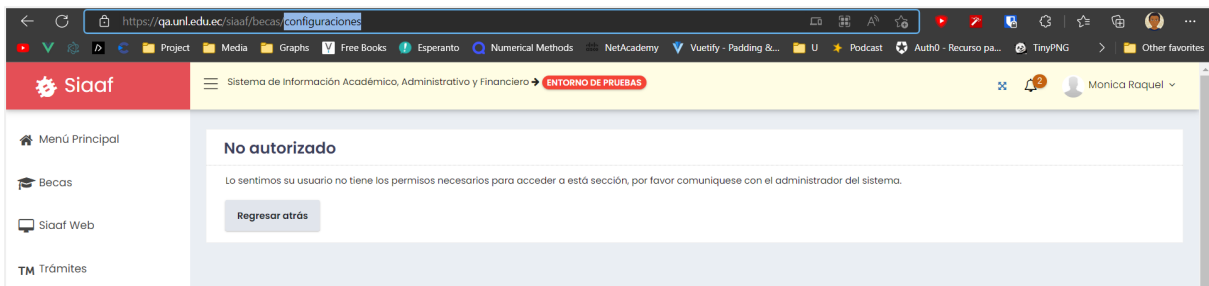
Figura 34. Ejecución de las Pruebas de carga y estabilidad de los de recursos atómicos con JMeter [captura de pantalla]



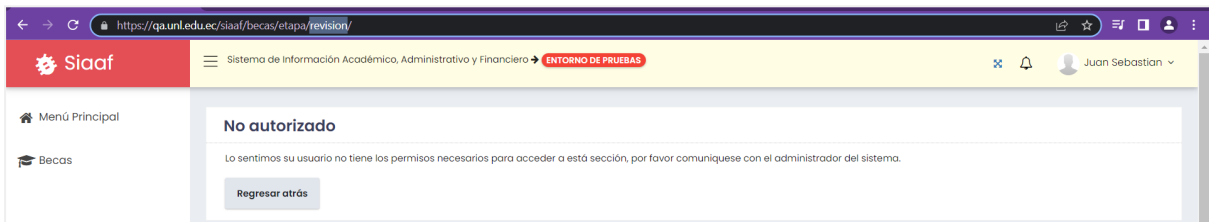
## Pruebas de Seguridad



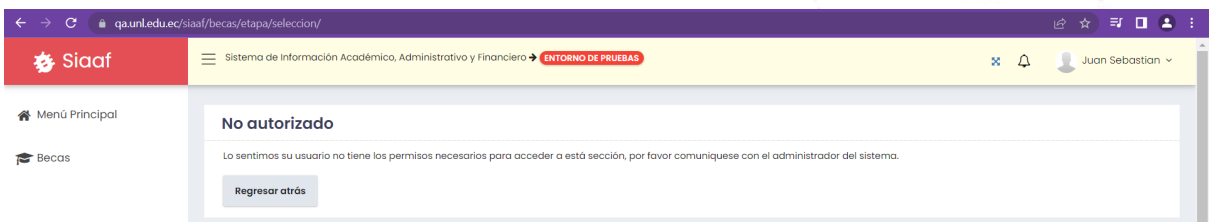
**Figura 37.** Intento de acceso a la Etapa de Selección con el rol de Tesorería [captura de pantalla]



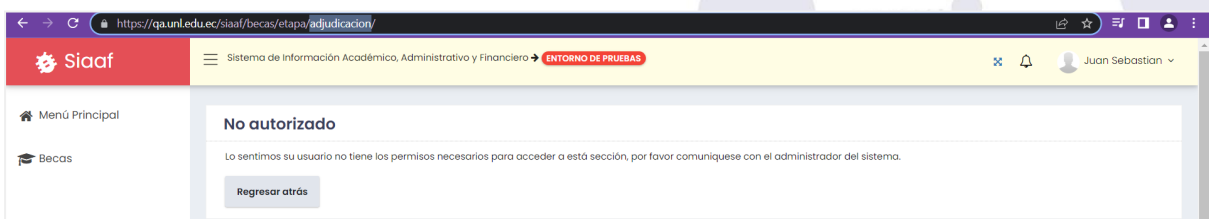
**Figura 38.** Intento de acceso a Configuraciones con el rol de Tesorería [captura de pantalla]



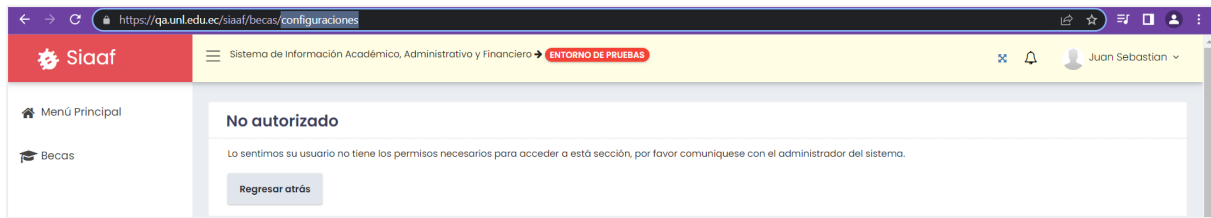
**Figura 39.** Intento de acceso a la Etapa de Revisión con el rol de Contabilidad Financiera [captura de pantalla]



**Figura 40.** Intento de acceso a la Etapa de Selección con el rol de Contabilidad Financiera [captura de pantalla]



**Figura 41.** Intento de acceso a la Etapa de Adjudicación con el rol de Contabilidad Financiera [captura de pantalla]



**Figura 42.** Intento de acceso a Configuraciones con el rol de Contabilidad Financiera [captura de pantalla]

## Pruebas Unitarias

```
test_finalizar_etapas_dom_service.py

class TestFinalizarEtapaDomService(SimpleTestCase):
    def setUp(self) -> None:
        self.beca = Beca(
            id=1,
            persona=Persona(id=1),
            tipo_beca=TipoBeca(id=1)
        )
        self.periodo_academico = PeriodoAcademico(id=1)
        self.etapa = Etapa(
            id=1,
            codigo=Constantes.ETAPA_REVISION.value
        )
        self.beca_periodo_academico_etapa = BecaPeriodoAcademicoEtapa(
            id=1,
            beca=self.beca,
            periodo_academico=self.periodo_academico,
            etapa=self.etapa,
            estado=BecaPeriodoAcademicoEtapaEstados.APROBADO
        )
```

**Figura 43.** Código de las Pruebas Unitarias configuración inicial para las pruebas de la clase FinalizarEtapaDomService [captura de pantalla]



```
test_finalizar_etapas_dom_service.py

def test_beca_esta_aprobada_postulante_aprobado(self):
    self.etapa.codigo = Constantes.ETAPA_SELECCION.value
    es_beca_aprobada = FinalizarEtapaDomService.beca_esta_aprobada(
        self.beca_periodo_academico_etapa, self.etapa)
    print('Si el postulante fue aprobado debería retornar True')
    print('es_beca_aprobada: ', es_beca_aprobada)
    self.assertTrue(es_beca_aprobada)

def test_beca_esta_aprobada_postulante_rechazado(self):
    self.beca_periodo_academico_etapa.estado = 3
    self.etapa.codigo = Constantes.ETAPA_SELECCION.value
    es_beca_aprobada = FinalizarEtapaDomService.beca_esta_aprobada(
        self.beca_periodo_academico_etapa, self.etapa)
    print('Si el postulante fue rechazado debería retornar False')
    print('es_beca_aprobada: ', es_beca_aprobada)
    self.assertFalse(es_beca_aprobada)
```

**Figura 44.** Código de las Pruebas Unitarias para el método beca\_esta\_aprobada [captura de pantalla]

```
test_finalizar_etapas_dom_service.py

def test_comprobar_estado_permitido_aprobado(self):
    estado_comprobar = BecaPeriodoAcademicoEtapaEstados.APROBADO
    dict_esperado = FinalizarEtapaDomService.comprobar_estado_permitido(
        estado_comprobar)
    print('Debería retornar un dict sin mensaje y false si el estado es APROBADO')
    print('el estado a comprobar es: ', estado_comprobar)
    print('respuesta: ', dict_esperado)

def test_comprobar_estado_permitido_rechazado(self):
    estado_comprobar = BecaPeriodoAcademicoEtapaEstados.RECHAZADO
    dict_esperado = FinalizarEtapaDomService.comprobar_estado_permitido(
        estado_comprobar)
    print('Debería retornar un dict sin mensaje y false si el estado es RECHAZADO')
    print('el estado a comprobar es: ', estado_comprobar)
    print('respuesta: ', dict_esperado)
```

**Figura 45.** Código de las Pruebas Unitarias al método comprobar\_estado\_permitido con los estados APROBADO y RECHAZADO [captura de pantalla]

```
test_finalizar_etapas_dom_service.py

def test_comprobar_estado_permitido_pendiente(self):
    estado_comprobar = BecaPeriodoAcademicoEtapaEstados.PENDIENTE
    dict_esperado = FinalizarEtapaDomService.comprobar_estado_permitido(
        estado_comprobar)
    print('Debería retornar un dict con un mensaje de advertencia si el estado es PENDIENTE')
    print('el estado a comprobar es: ', estado_comprobar)
    print('respuesta: ', dict_esperado)

def test_comprobar_estado_permitido_en_revision(self):
    estado_comprobar = BecaPeriodoAcademicoEtapaEstados.EN_REVISION
    dict_esperado = FinalizarEtapaDomService.comprobar_estado_permitido(
        estado_comprobar)
    print('Debería retornar un dict con un mensaje de advertencia si el estado es EN_REVISION')
    print('el estado a comprobar es: ', estado_comprobar)
    print('respuesta: ', dict_esperado)
```

**Figura 46.** Código de las Pruebas Unitarias al método comprobar\_estado\_permitido con los estados PENDIENTE y EN\_REVISION [captura de pantalla]

```
test_reportes_app_service.py

class TestReportesAppService(SimpleTestCase):
    def setUp(self):
        self.persona = Persona(id=1)
        self.tipo_beca = TipoBeca(id=1)
        self.beca = Beca(
            id=1,
            persona=self.persona,
            tipo_beca=self.tipo_beca
        )
        self.periodo_academico = PeriodoAcademico(id=1)
        self.registro_pago = RegistroPago(
            beca=self.beca,
            periodo_academico=self.periodo_academico,
            monto=None
        )

    def test_obtener_monto_None(self):
        monto_obtenido = ReportesAppService.obtener_monto(self.registro_pago)
        print('Si el monto no existe debería retornar un monto de 0')
        print('monto obtenido: ', monto_obtenido)
        self.assertEqual(monto_obtenido, 0)

    def test_obtener_monto_guardado(self):
        self.registro_pago.monto = 45
        monto_obtenido = ReportesAppService.obtener_monto(self.registro_pago)
        print('Si el monto existe debería retornar el monto guardado')
        print('monto obtenido: ', monto_obtenido)
        self.assertEqual(monto_obtenido, 45)
```

**Figura 47.** Código de las Pruebas Unitarias a la clase ReportesAppService [captura de pantalla]



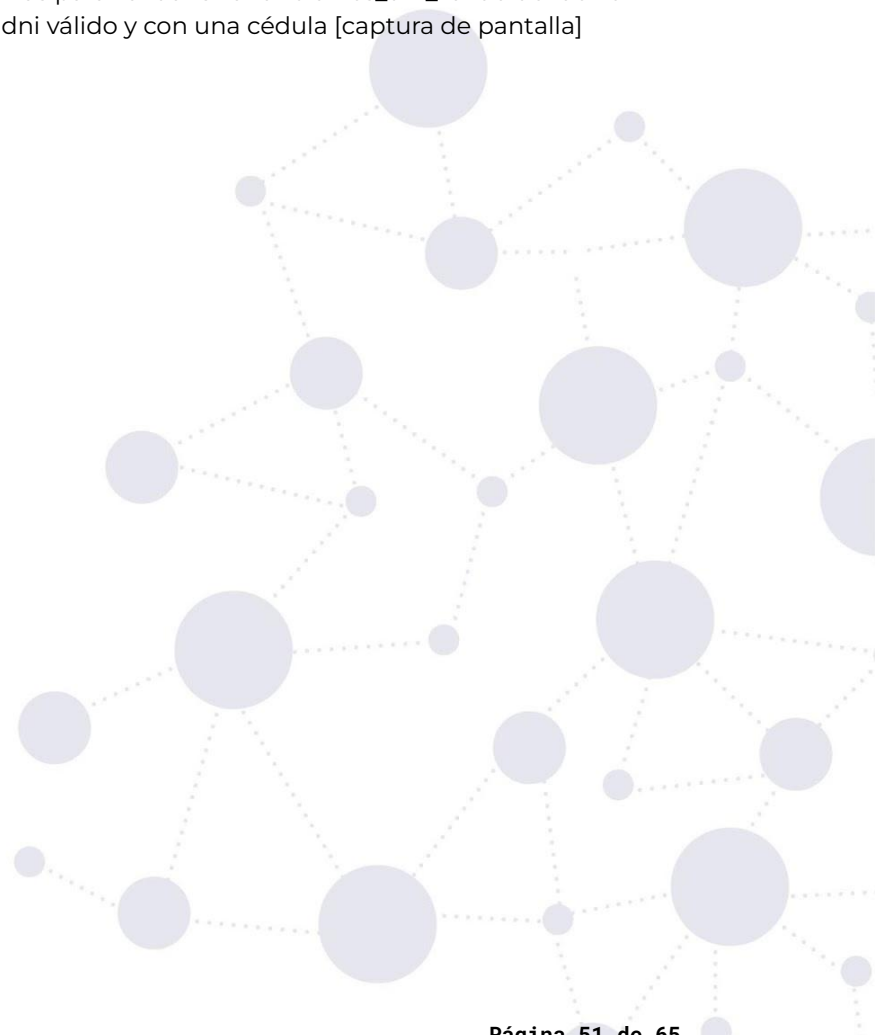
```
test_procesar_datos_importacion.py

class TestProcesarDatosImportacion(SimpleTestCase):
    def setUp(self):
        self.dni = '1234567'

    def test_dni_invalido(self):
        print('Si el dni es invalido debería retornar False')
        dni_validado = es_dni_valido(self.dni)
        print('dni a validar: ', self.dni)
        print('es un dni valido: ', dni_validado)
        self.assertFalse(dni_validado)

    def test_cedula(self):
        print('Si el dni es valido debería retornar True')
        self.dni = '1105740979'
        dni_validado = es_dni_valido(self.dni)
        print('dni a validar: ', self.dni)
        print('es un dni valido: ', dni_validado)
        self.assertTrue(dni_validado)
```

**Figura 48.** Código de las Pruebas Unitarias para validar si la función `es_dni_valido` devuelve correctamente la respuesta con un dni válido y con una cédula [captura de pantalla]

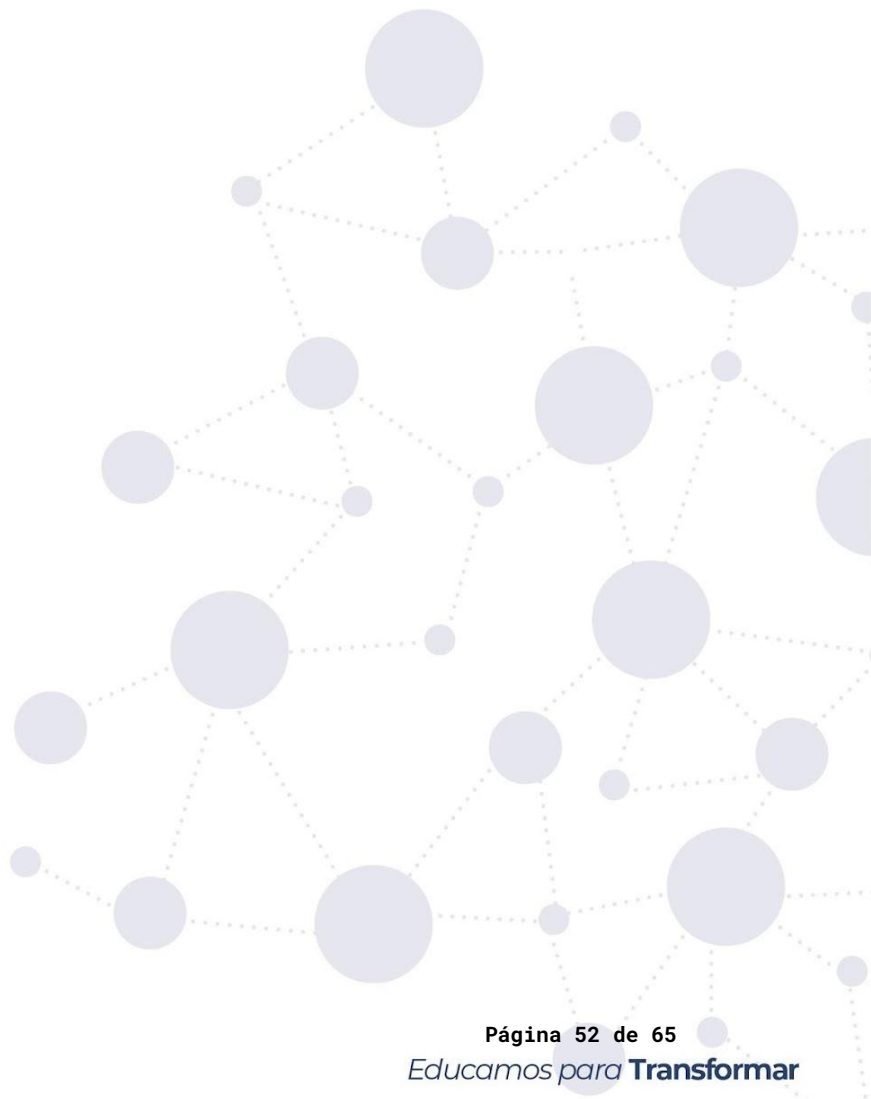


```
test_procesar_datos_importacion.py

def test_ruc(self):
    print('Si el dni es valido debería retornar True')
    self.dni = '1105740979001'
    dni_validado = es_dni_valido(self.dni)
    print('dni a validar: ', self.dni)
    print('es un dni valido: ', dni_validado)
    self.assertTrue(dni_validado)

def test_pasaporte(self):
    print('Si el dni es valido debería retornar True')
    self.dni = 'A3705348'
    dni_validado = es_dni_valido(self.dni)
    print('dni a validar: ', self.dni)
    print('es un dni valido: ', dni_validado)
    self.assertTrue(dni_validado)
```

**Figura 49.** Código de las Pruebas Unitarias para validar si la función `es_dni_valido` devuelve correctamente la respuesta con un número de ruc y con un número de pasaporte [captura de pantalla]



```
test_procesar_datos_importacion.py

def test_monto_valido(self):
    print(self.probandando_es_monto_valido)
    print('Si el monto es un número entero debería retornar True')
    monto = '100'
    monto_valido = es_monto_valido(monto)
    print('monto a validar: ', monto)
    print('respuesta: ', monto_valido)
    self.assertTrue(monto_valido)

def test_monto_invalido(self):
    print(self.probandando_es_monto_valido)
    print('Si el monto es 0 debería retornar False')
    monto = '0'
    monto_valido = es_monto_valido(monto)
    print('monto a validar: ', monto)
    print('respuesta: ', monto_valido)
    self.assertFalse(monto_valido)

def test_monto_vacio(self):
    print(self.probandando_es_monto_valido)
    print('Si el monto esta vacio debería retornar False')
    monto = ''
    monto_valido = es_monto_valido(monto)
    print('monto a validar: ', monto)
    print('respuesta: ', monto_valido)
    self.assertFalse(monto_valido)

def test_monto_caracteres_no_numericos(self):
    print('Si el monto contiene caracteres no numéricos debería retornar False')
    monto = '50a#'
    monto_valido = es_monto_valido(monto)
    print('monto a validar: ', monto)
    print('respuesta: ', monto_valido)
    self.assertFalse(monto_valido)
```

**Figura 50.** Código de las Pruebas Unitarias para validar si la función `es_monto_valido` devuelve correctamente la respuesta con todos los escenarios posibles [captura de pantalla]

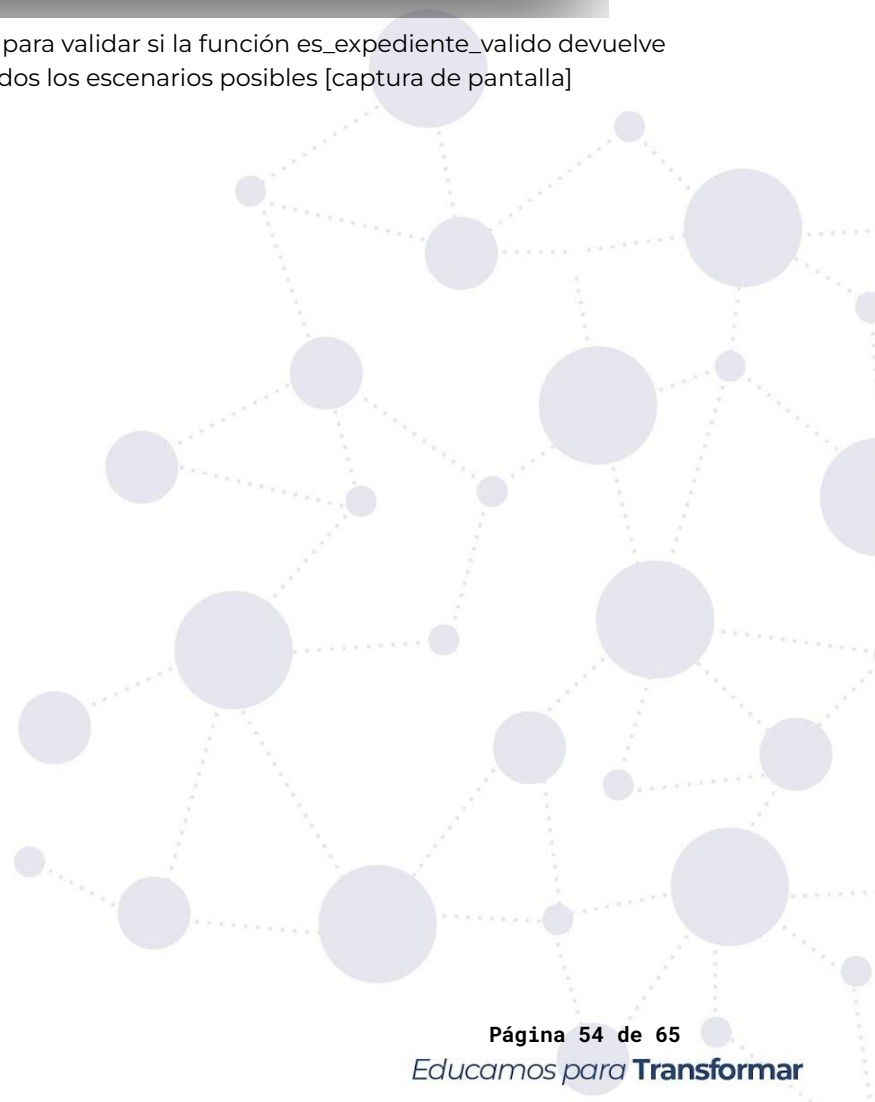
```
test_procesar_datos_importacion.py

def test_expediente_valido(self):
    print(self.probando_es_expediente_valido)
    print('Si el expediente contiene más de 1 caracter debería retornar True')
    expediente = 'EXP123'
    expediente_valido = es_expediente_valido(expediente)
    print('expediente a validar: ', expediente)
    print('respuesta: ', expediente_valido)
    self.assertTrue(expediente_valido)

def test_expediente_corto(self):
    print(self.probando_es_expediente_valido)
    print('Si el expediente contiene 1 caracter debería retornar False')
    expediente = '1'
    expediente_valido = es_expediente_valido(expediente)
    print('expediente a validar: ', expediente)
    print('respuesta: ', expediente_valido)
    self.assertFalse(expediente_valido)

def test_expediente_largo(self):
    print(self.probando_es_expediente_valido)
    print('Si el expediente contiene más de 23 caracteres debería retornar False')
    expediente = 'EXP12345678901234567890123'
    expediente_valido = es_expediente_valido(expediente)
    print('expediente a validar: ', expediente)
    print('respuesta: ', expediente_valido)
    self.assertFalse(expediente_valido)
```

**Figura 51.** Código de las Pruebas Unitarias para validar si la función `es_expediente_valido` devuelve correctamente la respuesta con todos los escenarios posibles [captura de pantalla]



```
test_procesar_datos_importacion.py

def test_tipo_registro_es_pagado(self):
    print(self.probando_deberia_validarse_expediente)
    print('Si el tipo de registro es pagado debería retornar False')
    tipo_registro = 'pagado'
    validar_expediente = deberia_validarse_expediente(tipo_registro)
    print('tipo de registro a importar: ', tipo_registro)
    print('respuesta: ', validar_expediente)
    self.assertFalse(validar_expediente)

def test_tipo_registro_es_devengado(self):
    print(self.probando_deberia_validarse_expediente)
    print('Si el tipo de registro es devengado debería retornar True')
    tipo_registro = 'devengado'
    validar_expediente = deberia_validarse_expediente(tipo_registro)
    print('tipo de registro a importar: ', tipo_registro)
    print('respuesta: ', validar_expediente)
    self.assertTrue(validar_expediente)

def test_tipo_registro_es_desconocido(self):
    print(self.probando_deberia_validarse_expediente)
    print('Si el tipo de registro es desconocido debería retornar True')
    tipo_registro = 'desconocido'
    validar_expediente = deberia_validarse_expediente(tipo_registro)
    print('tipo de registro a importar: ', tipo_registro)
    print('respuesta: ', validar_expediente)
    self.assertTrue(validar_expediente)
```

**Figura 52.** Código de las Pruebas Unitarias para validar si la función `deberia_validarse_expediente` devuelve correctamente la respuesta con todos los escenarios posibles [captura de pantalla]

```
test_procesar_datos_importacion.py

def test_celda_meses_a_lista(self):
    print(self.probandando_str_a_list)
    print('Si la celda contiene texto con meses separados por comas debería retornar una lista con los meses')
    celda = 'ene, feb, mar'
    meses = str_a_list(celda)
    print('celda: ', celda)
    print('respuesta: ', meses)
    self.assertEqual(meses, ['ene', 'feb', 'mar'])

def test_celda_meses_vacia_a_lista(self):
    print(self.probandando_str_a_list)
    print('Si la celda esta vacia debería retornar una lista vacia')
    celda = ''
    meses = str_a_list(celda)
    print('celda: ', celda)
    print('respuesta: ', meses)
    self.assertEqual(meses, [])
```

**Figura 53.** Código de las Pruebas Unitarias para validar si la función `str_a_list` devuelve correctamente la respuesta con todos los escenarios posibles [captura de pantalla]

## Pruebas de Código

```
1 = {% extends 'becas/index_becas.html' %}
2 {% load core_tags %}
3 {% block estilos %}
4 jacks... {% include 'fragmentos/estilos_comunes.html' %}
5 = {% endblock %}
6 {% block contenido %}
7 <div class="card card-box">
8 <div class="card-head">
9 <header>Importación de Registros</header>
10 jacks... <div class="tools">
11 {% if datos_no_procesados|length > 0 %}
12 {% # descargar reporte con errores de la importacion #}
13 {% include 'becas/partial/boton.html' with tipo="danger" titulo="Descargar reporte de errores" boton_id="btn-errores-importacion" icono="times"
14 etiqueta="Descargar reporte con errores" %}
15 {% # descargar reporte con errores de la importacion #}
16 {% endif %}
17 jacks... </div>
18 jacks... </div>
19 jacks... <div class="card-body">
20 jacks... <div class="row">
21 jacks... <div class="col-md-2">
22 - <form method="post" enctype="multipart/form-data" enctype="multipart/form-data" name="enviar-archivo">
23 <input type="hidden" name="csrf_token" value="{% csrf_token %}" />
24 <div class="form-group">
25 <input type="file" accept=".xls" name="archivo_devengado" class="custom-file-input" id="archivo" lang="es">
26 <small class="form-text text-muted">
27 <span class="text-danger">*</span>
28 Solo se aceptan archivos .xls
29 </small>
30 jacks... </div>
31 jacks... <div class="form-group">
32 <label for="tipo_archivo">Tipo de Archivo</label>
33 <select name="tipo_archivo" id="tipo_archivo" class="form-control">
34 <option value="devengado">Devengado</option>
35 <option value="pagado">Pago</option>
36 </select>
37 jacks... <small class="form-text text-muted">
38 <span class="text-danger">*</span>
39 Elija el tipo de importación
40 </small>
41 jacks... </div>
42 = <div class="d-flex justify-content-center form-group">
43 <button type="submit" class="btn btn-success" id="cargar_archivo">Cargar</button>
44 </div>
45 jacks... </form>
46 jacks... </div>
47 jacks... <div class="col-md-10 w-100 p-3">
48 <div class="form-group">
49 <small class="form-text text-muted">
50 <span class="text-danger">*</span>
51 Elija el tipo de importación
52 </small>
53 </div>
54 <div class="d-flex justify-content-center form-group">
55 <button type="submit" class="btn btn-success" id="cargar_archivo">Cargar</button>
56 </div>
57 </div>
58 <div class="alert alert-warning text-justify" role="alert">
59 <span class="fa fa-info-circle"></span>
60 </div>
```

Surround this <i> item tag by a <ul> or <ol> container one. Why is this an issue? 3 months ago · L59

🐛 Bug · 🟡 Minor · 🔓 Open · Not assigned · 10min effort · Comment · No tags

Figura 54. Bugs detectados en el código html referente al uso inapropiado de la etiqueta <i> [captura de pantalla]





```

Modulo de Gestion de Becas (MG... / app / becas / layer / application / filtros_etapas_app_service.py

27     qset = Q(etapa_codigo=etapa_codigo)
28     total = queryset.filter(qset).count()
29
30     if carrera and carrera != '':
31         qset = qset & (
32             Q(beca_carrera_id=carrera))
33     if estado and estado != '':
34         qset = qset & (
35             Q(estado=estado))
36     if facultad and facultad != '':
37         qset = qset & (
38             Q(beca_carrera_facultad_id=facultad))
39         qset = qset

40
41     if periodo_academico and periodo_academico != '':
42         qset = qset & (
43             Q(periodo_academico_id=periodo_academico))
44     if tipo_beca and tipo_beca != '':
45         qset = qset & (
46             Q(beca_tipo_beca_id=tipo_beca))
47
48     if search_value and search_value:
49         for s_value in search_value:

```

**Remove or correct this useless self-assignment.** Why is this an issue? 1 month ago L39 No tags

**Figura 57.** Bugs detectados referentes a la asignación innecesaria de una variable y la duplicidad en la validación de sub-expresiones de un if [captura de pantalla]

```

shell > app > becas > layer > application > filtros_etapas_app_service.py > FiltrosEtapasAppService > get_filtros_etapas

39     qset = qset
40     if periodo_academico and periodo_academico != '':
41         qset = qset & (
42             Q(periodo_academico_id=periodo_academico))
43     if tipo_beca and tipo_beca != '':
44         qset = qset & (
45             Q(beca_tipo_beca_id=tipo_beca))
46
47     if search_value and search_value:
48         for s_value in search_value:
49             qset = qset & (
50                 Q(
51                     beca_persona_primer_apellido_icontains=s_value) |
52                     Q(
53                         beca_persona_segundo_apellido_icontains=s_value) |
54                     Q(
55                         beca_persona_primer_nombre_icontains=s_value) |
56                     Q(
57                         beca_persona_segundo_nombre_icontains=s_value) |
58                     Q(
59                         beca_persona_numero_documento_icontains=s_value)
60             )
61     return queryset, qset, total

39     if periodo_academico and periodo_academico != '':
40         qset = qset & (
41             Q(periodo_academico_id=periodo_academico))
42     if tipo_beca and tipo_beca != '':
43         qset = qset & (
44             Q(beca_tipo_beca_id=tipo_beca))
45
46     if search_value and search_value != '':
47         for s_value in search_value:
48             qset = qset & (
49                 Q(
50                     beca_persona_primer_apellido_icontains=s_value) |
51                     Q(
52                         beca_persona_segundo_apellido_icontains=s_value) |
53                     Q(
54                         beca_persona_primer_nombre_icontains=s_value) |
55                     Q(
56                         beca_persona_segundo_nombre_icontains=s_value) |
57                     Q(
58                         beca_persona_numero_documento_icontains=s_value)
59             )
60     return queryset, qset, total

```

**Figura 58.** Corrección de bugs detectados referentes a la asignación innecesaria de una variable y la duplicidad en la sub-expresiones de un if [captura de pantalla]

Modulo de Gestion de Becas (MG... / app / becas) View as **Tree** to select files to navigate 12 files

Duplicated Lines (%) 2.3%

	Duplicated Lines (%)	Duplicated Lines
layer	0.0%	0
static	0.0%	0
templates	8.5%	102
tests	0.0%	0
urls	0.0%	0
utils	0.0%	0
views	4.9%	37
__init__.py	0.0%	0
admin.py	0.0%	0
apps.py	0.0%	0
model_enums.py	0.0%	0
models.py	0.0%	0

**Figura 59.** Porcentaje de duplicidad del código en el MGB [captura de pantalla]



Modulo de Gestion de Becas (MGB) > app > becas > templates > becas > etapas > etapa\_revision.html

Lines	Duplications	Bug	Vuln
91	56.0%	0	0

```

1  jacks...  {% extends 'becas/index_becas.html' %}
2  jacks...  {% load core_tags %}
3  jacks...  {% block estilos %}
4  jacks...  {% include 'fragmentos/estilos_comunes.html' %}
5  jacks...  {% endblock %}
6  jacks...  {% block contenido %}
7  jacks...  <div class="card card-box">
8  jacks...  <div class="card-head">
9  jacks...  <header>Listado de Postulaciones - Revisión de la Documentación</header>
10 jacks...  {% block botones para etapa de revision y seleccion %}
11 jacks...  {% include 'fragmentos/botones_postulantes.html' with opciones=estados_etapa %}
12 jacks...  </div>
13 jacks...  <div class="card-body ">
14 jacks...  {% filtros #}
15 jacks...  {% include 'fragmentos/filtros_etapas.html' with facultades=facultades programas_estudio=programas_estudio tipos_beca=
periodos_academicos=periodos_academicos %}
16 jacks...  {% filtros #}
17 jacks...  {% mensaje etapa finalizada #}
18 jacks...  {% include 'fragmentos/mensaje_etapa_finalizada.html' %}
19 jacks...  {% mensaje etapa finalizada #}
20 jacks...  {% botones de accion para la tabla #}
21 jacks...  {% include 'fragmentos/botones_tabla.html' %}
22 jacks...  {% botones de accion para la tabla #}
23 jacks...  {% tabla #}
24 jacks...  {% include 'fragmentos/tabla_etapas.html' with estudiante='Postulante' etapa=etapa %}
25 jacks...  {% tabla #}
26 jacks...  =
27 jacks...  {% footer etapas #}
28 jacks...  {% include 'fragmentos/footer_etapas.html' with etapa=etapa estados=estados_etapa %}
29 jacks...  </div>
30 jacks...  </div>
31 jacks...  {% endblock %}
32 jacks...
33 jacks...  {% block javascript %}
34 jacks...  {% include 'fragmentos/javascript_comun.html' %}
35 jacks...
36 jacks...  Duplicated By
37 jacks...  app/becas/templates/becas/etapas/etapa_seleccion.html
38 jacks...  Lines: 28 – 50
39 jacks...  let estados_etapa = [];
40 jacks...  {% for estado in estados_etapa %}
41 jacks...  estados_etapa.push({
42 jacks...  'accion': '{{estado.accion}}',
43 jacks...  'valor': '{{estado.valor}}',
44 jacks...  'requiere_confirmacion': ('{{estado.requiere_confirmacion}}' === 'True'),
45 jacks...  'requiere_mensaje': ('{{estado.requiere_mensaje}}' === 'True')
46 jacks...  });
47 jacks...  {% endfor %}
48 jacks...  console.log(estados_etapa)
49 jacks...  var mensajes_frecuentes = {
50 jacks...  'aprobar': {
51 jacks...  'opciones': '',
52 jacks...  'favorito': ''
53 jacks...  },
54 jacks...  'rechazar': {
55 jacks...
56 jacks...  Duplicated By
57 jacks...  app/becas/templates/becas/etapas/etapa_seleccion.html
58 jacks...  Lines: 52 – 79
59 jacks...  {% for mensaje in mensajes_frecuentes %}
60 jacks...  {% if mensaje.tipo == 0 %}
61 jacks...  mensajes_frecuentes['aprobar']['opciones'] += '<option value="{{mensaje.texto}}">{{mensaje.texto}}</option>';
62 jacks...  {% if mensaje.favorito %}
63 jacks...  mensajes_frecuentes['aprobar']['favorito'] = '{{mensaje.texto}}';
64 jacks...  {% endif %}
65 jacks...  {% endif %}
66 jacks...  {% if mensaje.tipo == 1 %}
67 jacks...  mensajes_frecuentes['rechazar']['opciones'] += '<option value="{{mensaje.texto}}">{{mensaje.texto}}</option>';
68 jacks...  {% if mensaje.favorito %}
69 jacks...  mensajes_frecuentes['rechazar']['favorito'] = '{{mensaje.texto}}';
70 jacks...  {% endif %}
71 jacks...  {% endif %}
72 jacks...  {% endfor %}
73 jacks...  var periodos_academicos_finalizados = []
74 jacks...  {% for etapa_finalizada in etapas_finalidas %}
75 jacks...  periodos_academicos_finalizados.push('{{etapa_finalizada.periodo_academico_id}}');
76 jacks...  {% endfor %}

```

Figura 60. Código duplicado entre los template de las etapas de revisión y selección en el MGB [captura de pantalla]

```
1- {% extends 'becas/index_becas.html' %}
2- {% load core_tags %}
3- {% load etapas_tags %}
4- {% block estilos %}
5-     {% include 'fragmentos/estilos_comunes.html' %}
6-     {% block estilos_etapa %}
7-     {% endblock estilos_etapa %}
8- {% endblock %}
9- {% block contenido %}
10-     <div class="card card-box">
11-         <div class="card-head">
12-             {% block card_head %}
13-                 {# botones comunes para etapa de revision y
14-                 seleccion #}
15-                 {% include 'fragmentos/botones_postulantes.
16-                 html' with opciones=estados_etapa %}
17-             {% endblock card_head %}
18-         </div>
19-         <div class="card-body ">
20-             {% block filtros %}
21-                 {# filtros comunes para las etapas de revision,
22-                 seleccion y adjudicacion #}
23-                 {% include 'fragmentos/filtros_etapas.html'
24-                 with facultades=facultades
25-                 programas_estudio=programas_estudio
26-                 tipos_becas=tipos_becas estados=estados_etapa
27-                 periodos_academicos=periodos_academicos %}
28-             {% endblock filtros %}
29-             {% block card_body %}
30-                 {# mensaje etapa finalizada #}
31-                 {% include 'fragmentos/mensaje_etapa_finalizada.
32-                 html' %}
33-                 {# mensaje etapa finalizada #}
34-                 {# botones de accion para la tabla #}
35-                 {% include 'fragmentos/botones_tabla.html' %}
36-                 {# botones de accion para la tabla #}
37-                 {# tabla #}
38-                 {% include 'fragmentos/tabla_etapas.html' with
39-                 estudiante='Postulante' etapa=etapa %}
40-                 {# tabla #}
41-                 {# footer etapas #}
42-                 {% include 'fragmentos/footer_etapas.html' with
43-                 etapa=etapa estados=estados_etapa %}
44-             {% endblock card_body %}
45-         </div>
46-     </div>
47- {% endblock contenido %}
```

Figura 61. Template etapas creado para agrupar código común de todas las etapas [captura de pantalla]

```
1- {% extends 'becas/index_becas.html' %}
2- {% load core_tags %}
3- {% block estilos %}
4-     {% include 'fragmentos/estilos_comunes.html' %}
5- {% endblock %}
6- {% block contenido %}
7-     <div class="card card-box">
8-         <div class="card-head">
9-             <header>Listado de Postulaciones - Revisión de la
10-             Documentación</header>
11-             {# botones comunes para etapa de revision y
12-             seleccion #}
13-             {% include 'fragmentos/botones_postulantes.html'
14-             with opciones=estados_etapa %}
15-         </div>
16-         <div class="card-body ">
17-             {# filtros #}
18-             {% include 'fragmentos/filtros_etapas.html' with
19-             facultades=facultades
20-             programas_estudio=programas_estudio
21-             tipos_beca=tipos_beca estados=estados_etapa
22-             periodos_academicos=periodos_academicos %}
23-             {# filtros #}
24-             {# mensaje etapa finalizada #}
25-             {% include 'fragmentos/mensaje_etapa_finalizada.
26-             html' %}
27-             {# mensaje etapa finalizada #}
28-             {# botones de accion para la tabla #}
29-             {% include 'fragmentos/botones_tabla.html' %}
30-             {# botones de accion para la tabla #}
31-             {# tabla #}
32-             {% include 'fragmentos/tabla_etapas.html' with
33-             estudiante='Postulante' etapa=etapa %}
34-             {# tabla #}
35-             {# footer etapas #}
36-             {% include 'fragmentos/footer_etapas.html' with
37-             etapa=etapa estados=estados_etapa %}
38-         </div>
39-     </div>
40- {% endblock %}
```

Figura 62. Refactorización del Template de la Etapa de Revisión [captura de pantalla]

```
home > jack > UNL > siaf > app > becas > templates > becas > etapas > dj etapa_seleccion.html  
1- {{ extends 'becas/index_becas.html' % }}  
2- {{ load core_tags % }}  
3- {{ block estilos % }}  
4- {{ include 'fragmentos/estilos_comunes.html' % }}  
5- {{ endblock % }}  
6- {{ block contenido % }}  
7- <div class="card card-box">  
8- <div class="card-head">  
9- <header>Listado de Postulaciones - Evaluación  
10- interna de la Comisión de Becas</header>  
11- <# botones comunes para etapa de revision y  
12- seleccion #>  
13- {{ include 'fragmentos/botones_postulantes.html'  
14- with opciones=estados_etapa % }}  
15- </div>  
16- <div class="card-body ">  
17- <# filtros #>  
18- <div class="form-row">  
19- <# filtros comunes para todas las etapas #>  
20- {{ include 'fragmentos/filtros_etapas.html'  
21- with facultades=facultades  
22- programas_estudio=programas_estudio  
23- tipos_becas=tipos_becas estados=estados_etapa  
24- periodos_academicos=periodos_academicos % }}  
25- </div>  
26- <# filtros #>  
27- <# mensaje etapa finalizada #>  
28- {{ include 'fragmentos/mensaje_etapa_finalizada.  
29- html' % }}  
30- <# mensaje etapa finalizada #>  
31- <# botones de accion para la tabla #>  
32- {{ include 'fragmentos/botones_tabla.html' % }}  
33- <# botones de accion para la tabla #>  
34- <# tabla #>  
35- {{ include 'fragmentos/tabla_etapas.html' with  
36- estudiante='Postulante' esEtapaRevision=False % }}  
37- <# tabla #>  
38- <# footer etapas #>  
39- {{ include 'fragmentos/footer_etapas.html' with  
40- etapa=etapa estados=estados_etapa % }}  
41- </div>  
42- </div>  
43- {{ endblock % }}  
44- {{ endblock % }}  
45- {{ extends './etapa.html' % }}  
46- {{ block card_head % }}  
47- <header>Listado de Postulaciones - Evaluación interna de la  
48- Comisión de Becas</header>  
49- <# botones comunes para etapa de revision y  
50- seleccion #>  
51- {{ include 'fragmentos/botones_postulantes.html' with  
52- opciones=estados_etapa % }}  
53- </div>  
54- <div class="card-body ">  
55- <# filtros #>  
56- <div class="form-row">  
57- <# filtros comunes para todas las etapas #>  
58- {{ include 'fragmentos/filtros_etapas.html'  
59- with facultades=facultades  
60- programas_estudio=programas_estudio  
61- tipos_becas=tipos_becas estados=estados_etapa  
62- periodos_academicos=periodos_academicos % }}  
63- </div>  
64- <# filtros #>  
65- <# mensaje etapa finalizada #>  
66- {{ include 'fragmentos/mensaje_etapa_finalizada.  
67- html' % }}  
68- <# mensaje etapa finalizada #>  
69- <# botones de accion para la tabla #>  
70- {{ include 'fragmentos/botones_tabla.html' % }}  
71- <# botones de accion para la tabla #>  
72- <# tabla #>  
73- {{ include 'fragmentos/tabla_etapas.html' with  
74- estudiante='Postulante' esEtapaRevision=False % }}  
75- <# tabla #>  
76- <# footer etapas #>  
77- {{ include 'fragmentos/footer_etapas.html' with  
78- etapa=etapa estados=estados_etapa % }}  
79- </div>  
80- </div>  
81- {{ endblock % }}  
82- {{ endblock % }}
```

Figura 63. Refactorización del Template de la Etapa de Selección [captura de pantalla]

```
home > jack > UNL > siaf > app > becas > templates > becas > etapas > dj etapa_adjudicacion.html  
1- {{ extends 'becas/index_becas.html' % }}  
2- {{ load core_tags % }}  
3- {{ block estilos % }}  
4- {{ include 'fragmentos/estilos_comunes.html' % }}  
5- {{ endblock % }}  
6- {{ block contenido % }}  
7- <div class="card card-box">  
8- <div class="card-head">  
9- <header>Listado de Becas</header>  
10- <div class="tools">  
11- <# if perms.becas.can_cambiar_estado_etapa %>  
12- {{ include 'becas/partials/boton_desplegable.html'  
13- with tipo="danger"  
14- boton_id='btn-actualizar-desplegable'  
15- titulo='Actualizar el estado de los  
16- Postulantes en la Etapa de Revisión'  
17- icono='circle-o-notch' etiqueta='Actualizar  
18- Estado' opciones=estados_etapa  
19- opcion_clase='actualizar_estados' % }}  
20- <# endif %>  
21- <# if perms.becas.can_confirmar_registro %>  
22- {{ include 'becas/partials/boton.html' with  
23- tipo="success" titulo='Confirmar registro  
24- en el ESIGEF'  
25- boton_id='confirmar_registro_esigef'  
26- icono='save' etiqueta='Confirmar Registro'  
27- % }}  
28- <# endif %>  
29- </div>  
30- </div>  
31- <div class="card-body ">  
32- <# filtros #>  
33- <div class="form-row">  
34- <# filtros comunes para todas las etapas #>  
35- {{ include 'fragmentos/filtros_etapas.html'  
36- with facultades=facultades carreras=carreras  
37- tipos_becas=tipos_becas estados=estados_etapa  
38- codigo_etapa=etapa.codigo  
39- periodos_academicos=periodos_academicos % }}  
40- </div>  
41- <# filtros #>  
42- <# botones de accion para la tabla #>  
43- {{ include 'fragmentos/botones_tabla.html' % }}  
44- <# botones de accion para la tabla #>  
45- <# tabla #>  
46- {{ include 'fragmentos/tabla_etapas.html' with  
47- estudiante='Becado' esEtapaRevision=False % }}  
48- <# tabla #>  
49- </div>  
50- </div>  
51- {{ endblock % }}  
52- {{ extends './etapa.html' % }}  
53- {{ block card_head % }}  
54- <header>Listado de Becas</header>  
55- <div class="tools">  
56- <# if perms.becas.can_cambiar_estado_etapa %>  
57- {{ include 'becas/partials/boton_desplegable.html'  
58- with tipo="danger"  
59- boton_id='btn-actualizar-desplegable'  
60- titulo='Actualizar el estado de los Postulantes en  
61- la Etapa de Revisión' icono='circle-o-notch'  
62- etiqueta='Actualizar Estado' opciones=estados_etapa  
63- opcion_clase='actualizar_estados' % }}  
64- <# endif %>  
65- <# if perms.becas.can_confirmar_registro %>  
66- {{ include 'becas/partials/boton.html' with  
67- tipo="success" titulo='Confirmar registro en el  
68- ESIGEF' boton_id='confirmar_registro_esigef'  
69- icono='save' etiqueta='Confirmar Registro' % }}  
70- <# endif %>  
71- </div>  
72- <div class="card-body ">  
73- <# filtros #>  
74- <div class="form-row">  
75- <# filtros comunes para todas las etapas #>  
76- {{ include 'fragmentos/filtros_etapas.html'  
77- with facultades=facultades carreras=carreras  
78- tipos_becas=tipos_becas estados=estados_etapa  
79- codigo_etapa=etapa.codigo  
80- periodos_academicos=periodos_academicos % }}  
81- </div>  
82- <# filtros #>  
83- <# botones de accion para la tabla #>  
84- {{ include 'fragmentos/botones_tabla.html' % }}  
85- <# botones de accion para la tabla #>  
86- <# tabla #>  
87- {{ include 'fragmentos/tabla_etapas.html' with  
88- estudiante='Becado' esEtapaRevision=False % }}  
89- <# tabla #>  
90- </div>  
91- </div>  
92- {{ endblock % }}  
93- {{ endblock % }}
```

Figura 64. Refactorización del Template de la Etapa de Adjudicación [captura de pantalla]





```
102 @staticmethod
103 def obtener_etapas_finalizadas(etapa: Union[QuerySet,
104                               Etapa], periodos_academicos: Union[QuerySet, List
105                               [PeriodoAcademico]]) -> Union[QuerySet, List
106                               [EtapaFinalizada]]:
107     """
108     Obtiene las etapas finalizadas de acuerdo a la etapa y
109     los periodos academicos
110     :param etapa: Union[QuerySet, Etapa]
111     :param periodos_academicos: Union[QuerySet, List
112     [PeriodoAcademico]]
113     :return: QuerySet
114     """
115     return EtapaFinalizada.objects.filter(
116         etapa=etapa,
117         periodo_academico__in=periodos_academicos)
118
```

```
102 @staticmethod
103 def obtener_etapas_finalizadas(etapa: Union[QuerySet,
104                               Etapa], periodos_academicos: Union[QuerySet, List
105                               [PeriodoAcademico]]) -> Union[QuerySet, List
106                               [EtapaFinalizada]]:
107     """
108     Obtiene las etapas finalizadas de acuerdo a la etapa y
109     los periodos academicos
110     :param etapa: Union[QuerySet, Etapa]
111     :param periodos_academicos: Union[QuerySet, List
112     [PeriodoAcademico]]
113     :return: QuerySet
114     """
115     lista_etapas_finalizadas = EtapaFinalizada.objects.
116     filter(
117         etapa=etapa,
118         periodo_academico__in=periodos_academicos).
119     values_list('periodo_academico_id', flat=True)
120     etapas_finalizadas_json = json.dumps(list
121     (lista_etapas_finalizadas))
122     return BecasAppService.get_determinante_json(
123     etapas_finalizadas_json)
124
```

Figura 67. Refactorización de la función obtener\_etapas\_finalizadas en la capa de aplicación de las Etapas [captura de pantalla]

```
139 @staticmethod
140 def obtener_mensajes_frecuentes(tipos: List[str]) -> Union
141 [QuerySet, List[MensajeFrecuente]]:
142     """
143     Obtiene los mensajes frecuentes de acuerdo a los tipos
144     :param tipos: List[str]
145     :return: QuerySet
146     """
147     mensajes_frecuentes = MensajeFrecuente.objects.filter(
148     tipo__in=tipos).all().order_by('orden')
149     return mensajes_frecuentes
150
```

```
142 @staticmethod
143 def obtener_mensajes_frecuentes(tipos: List[str]) -> Union
144 [QuerySet, List[MensajeFrecuente]]:
145     """
146     Obtiene los mensajes frecuentes de acuerdo a los tipos
147     :param tipos: List[str]
148     :return: QuerySet
149     """
150     mensajes_frecuentes = MensajeFrecuente.objects.filter(
151     tipo__in=tipos).all().order_by('orden').values
152     ('tipo', 'texto', 'favorito')
153     mensajes_frecuentes_json = json.dumps(list
154     (mensajes_frecuentes))
155     return BecasAppService.get_determinante_json(
156     mensajes_frecuentes_json)
157
```

Figura 68. Refactorización de la función obtener\_mensajes\_frecuentes en la capa de aplicación de las Etapas [captura de pantalla]

## Pruebas de Aceptación del Usuario

### Acta de Reunión UNL-DTI-2022-002

Acta de Reunión Nro. UNL-DTI-2022-002-firmado.pdf

### Encuesta de Satisfacción previo al uso del MGB aplicada a UBU

Encuesta de satisfacción en la implantación del Módulo de Gestión de Becas ...

### Encuesta de Satisfacción previo al uso del MGB aplicada a Contabilidad Financiera

Encuesta de satisfacción en la implantación del Módulo de Gestión de Becas ...



## Encuesta de Satisfacción previo al uso del MGB aplicada a Tesorería Financiera

Encuesta de satisfacción en la implantación del Módulo de Gestión de Becas ...

### Acta de Reunión UNL-DTI-2022-003

Acta de Reunión Nro. UNL-DTI-2022-003-firmado.pdf

## Asistencia virtual en el uso del MGB al usuario de Tesorería Financiera durante la fase de Pilotaje

[Asistencia vía zoom](#)

## Asistencia virtual en el uso del MGB al usuario Administrador de Becas durante la fase de Pilotaje

- [Asistencia vía zoom 1](#)
- [Asistencia vía zoom 2](#)

## Encuesta de Satisfacción posterior al uso del MGB aplicada a UBU

Encuesta de satisfacción en la implantación del Módulo de Gestión de Becas ...

## Encuesta de Satisfacción posterior al uso del MGB aplicada a Contabilidad Financiera

Encuesta de satisfacción en la implantación del Módulo de Gestión de Becas ...



## Encuesta de Satisfacción posterior al uso del MGB aplicada a Tesorería Financiera

Encuesta de satisfacción en la implantación del Módulo de Gestión de Becas ...

## Encuesta de Aceptación del MGB aplicada al usuario de la UBU

Encuesta para la evaluación del Módulo de Gestión de Becas - Pruebas de Ac...

## Encuesta de Aceptación del MGB aplicada al usuario de Contabilidad Financiera

Encuesta para la evaluación del Módulo de Gestión de Becas - Pruebas de Ac...

## Encuesta de Aceptación del MGB aplicada al usuario de Tesorería Financiera

Encuesta para la evaluación del Módulo de Gestión de Becas - Pruebas de Ac...

## Encuesta de Aceptación del MGB aplicada al product owner en calidad de representante de la DTI

Encuesta para la evaluación del Módulo de Gestión de Becas - Pruebas de Ac...

## Encuesta de Aceptación del MGB aplicada al QA en calidad de representante de la DTI

Encuesta para la evaluación del Módulo de Gestión de Becas - Pruebas de Ac...

## **Anexo 8. Certificación de Control de Calidad para el MGB**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Certificado de Control de Calidad<sup>10</sup>

---

<sup>10</sup> Versión original en: <https://drive.google.com/Control-Calidad-DTI-MGB.pdf>





### Informe Nro. UNL-DTI-2022-126

<b>Para:</b>	Ing. Jhon Calderón <b>DIRECTOR DE TECNOLOGÍAS DE INFORMACIÓN</b> email: <a href="mailto:direccion.uti@unl.edu.ec">direccion.uti@unl.edu.ec</a>
<b>De:</b>	Ing. Johana Cueva <b>ANALISTA DE SISTEMAS INFORMÁTICOS 1</b>
<b>Asunto:</b>	<b>Certificación de Control de Calidad:</b> <b>Proyecto 2020-007</b> - Implementación del Módulo de Gestión de Becas para Bienestar Universitario - SIAAF
<b>Referencia:</b>	<a href="#">27626</a> - SE-No.02-ROCS-No-05-08-02-2021 (Reglamento de Becas)
<b>Tickets -</b>	<a href="#">33588</a> , <a href="#">33579</a> - Certificado para realizar el módulo de Becas
<b>Oficios No</b>	<a href="#">37202</a> - Revisión del documento de ERS para el MGB <a href="#">39880</a> - Registro de pago de becas <a href="#">40108</a> - Conexión MGT con MGB
<b>Fecha:</b>	Loja, 02 de septiembre del 2022

## ANTECEDENTES

La implementación de los requerimientos del Módulo de Gestión de Becas para Bienestar Universitario - SIAAF, se lleva a cabo considerando los siguientes antecedentes:

- Mediante ticket de la Mesa de Servicios Nro.. 27626 con Of. Nro. 09-SG-OCS-UNL se hace referencia a la Resolución SE-No.02-ROCS-No-05-08-02-2021, adoptada en la Segunda Sesión Extraordinaria del Órgano Colegiado Superior de la Universidad Nacional de Loja (OCS), desarrollada el 8 de Febrero de 2021, mediante la cual resolvió en lo principal aprobar el **“REGLAMENTO DE LA SECCIÓN DE BECAS E INCENTIVOS DE BIENESTAR UNIVERSITARIO DE LA UNIVERSIDAD NACIONAL DE LOJA”**.
- Mediante ticket de la Mesa de Servicios Nro. 33588 y 33579 en el que consta, de fecha 20 de septiembre del 2021, suscrito por el Sr. Jackson Guzmán estudiante de la Carrera de Ingeniería en Sistemas, donde solicita:

“ (...) la emisión de un certificado que avale la solicitud por mi parte de la Unidad de Bienestar Universitario del módulo de becas.(...)”

- Mediante ticket de la Mesa de Servicios Nro. 37202 en el que consta, de fecha 23 de febrero del 2022, suscrito por el Sr. Jackson Guzmán estudiante de la Carrera de Ingeniería en Sistemas, donde solicita:



“(…) reunión de trabajo ... para la revisión del documento de ERS para el MGB.(…)”

→ Mediante ticket de la Mesa de Servicios Nro. 39880 en el que consta, de fecha 12 de julio del 2022, suscrito por el Sr. Jackson Guzmán estudiante de la Carrera de Ingeniería en Sistemas, donde solicita:

“(…) Revisión Registro de Pago de Becas.(…)”

→ Mediante ticket de la Mesa de Servicios Nro. 40108 en el que consta, de fecha 27 de julio del 2022, suscrito por el Sr. Jackson Guzmán estudiante de la Carrera de Ingeniería en Sistemas, donde solicita:

“(…) Revisión de la conexión entre el MGB y el MGT.(…)”

### CONTROL DE CALIDAD

A continuación se detalla los casos de uso con el total de casos de pruebas realizados para el Módulo de Gestión de Becas - SIAAF.

Proyecto 2021-007 - Implementación del Módulo de Gestión de Becas para Bienestar Universitario - SIAAF								
CU Casos de uso	DESCRIPCIÓN	CP Casos de prueba	CPNE CP NO ejecutados	IR Incidencias reportadas	EI Estado de incidencia	A Aprobado	%	
<b>CASOS DE USO FUNCIONALES</b>								
CU001	Sincronizar datos de postulación desde el MGT hacia el MGB	1	0	2	Resuelta	Si	100	
CU002	Actualizar estados de postulaciones	9	0	13	Resuelta	Si	100	
CU003	Descargar documentos de postulación	2	0	2	Resuelta	Si	100	
CU004	Ver trámite en el MGT	2	0	2	Resuelta	Si	100	
CU005	Finalizar Etapas	3	0	3	Resuelta	Si	100	
CU006	Actualizar estados de Becas	7	0	10	Resuelta	Si	100	
CU007	Visualizar Postulaciones y Becas	3	0	5	Resuelta	Si	100	
CU008	Generar Reportes	4	0	5	Resuelta	Si	100	
CU009	Agregar mensaje frecuente	1	0	2	Resuelta	Si	100	
CU010	Actualizar mensaje frecuente	1	0	2	Resuelta	Si	100	
CU011	Eliminar mensaje frecuente	1	0	2	Resuelta	Si	100	



CU012	Marcar como favorito mensaje frecuente	1	0	2	Resuelta	Si	100
CU013	Actualizar montos de pago para los tipos de becas	1	0	2	Resuelta	Si	100
CU014	Visualizar estadísticas generales de becados	1	0	1	Resuelta	Si	100
CU015	Iniciar Sesión	1	0	1	Resuelta	Si	100
CU016	Notificar resultados de postulación	2	0	3	Resuelta	Si	100
CU017	Guardar confirmación de registro de beca del sistema financiero	2	0	3	Resuelta	Si	100
CU018	Registrar Pago de Becas	8	0	12	Resuelta	Si	100
<b>CASOS DE USO NO FUNCIONALES</b>							
CU001	Las transacciones de lectura de información deben realizarse en menos de 2s.	1	0	1	Resuelta	Si	100
CU002	Las transacciones de lectura deben cargarse bajo demanda organizados en lotes de 10, 25 y 50 registros al mismo tiempo.	1	0	1	Resuelta	Si	100
CU003	En el 90% de los casos las transacciones de guardado y modificación deben tardar 2s o menos.	1	0	1	Resuelta	Si	100
CU004	El sistema será capaz de permitir la conexión simultánea de 2 o más usuarios al mismo tiempo.	1	0	1	Resuelta	Si	100
CU005	El módulo informático permitirá únicamente el acceso a los administradores del SIAAF, al Responsable de la Gestión de Becas, al Responsable del Registro de Pagos y al Contador Financiero	1	0	1	Resuelta	Si	100
CU006	El módulo informático hará la distinción entre los roles y privilegios que tiene cada usuario.	1	0	1	Resuelta	Si	100
CU007	La sesión del usuario utilizará el servicio de autenticación implementado para el SIAAF.	1	0	1	Resuelta	Si	100
CU008	La seguridad de la aplicación está cubierta por la arquitectura provista por el SIAAF.	1	0	1	Resuelta	Si	100
CU009	El módulo informático está pensado para interactuar activamente con el MGT, por lo	1	0	1	Resuelta	Si	100



	que se hará uso de apis para permitir la comunicación entre estos.							
CU010	El Módulo puede accederse desde cualquier dispositivo con conexión a internet para su uso.	1	0	1	Resuelta	Si	100	

## ROLES TÉCNICOS

**Administrador de Becas Dirección de Bienestar Universitario (AB).**- El presente rol ejecuta el o la Responsable de la Administración de Becas; cuenta con los siguientes permisos:

- Sincronizar datos de postulantes desde el Módulo de Gestión de Trámites al Módulo de Gestión de Becas.
- Revisión de documentación de postulantes.
- Aprobación de las etapas de Revisión y Selección
- Asignación de de calificación a la ficha socioeconómica
- Actualizar estados de revisión, aprobación o rechazo del postulante a la beca
- Reactiva, suspende y termina la beca del becario.
- Sincronizar por periodo académico becas activas
- Notificar resultados
- Descargar listado de becados del periodo actual
- Configuración de registros de pagos por tipo de beca
- Configuración de mensajes frecuentes para las etapas del Módulo de Gestión de Becas
- Generación de reportes

**Subdirección de Tesorería (ST).**- El presente rol ejecuta el Subdirector/a de Tesorería; cuenta con los siguientes permisos:

- Guardar registro de pago de los becados en el Módulo de Gestión de Becas
- Guardar confirmación de registro de nuevos becados.
- Importar listas para registro de devengados en el Módulo de Gestión de Becas
- Importar registros de pagos
- Generación de reportes

**Subdirección de Contabilidad (SC).**- El presente rol ejecuta el Subdirector/a de Contabilidad; cuenta con los siguientes permisos:

- Visualizar el detalle del registro de pago de becas

**Administrador de Permisos a Roles (DTI).**- El presente rol ejecuta el Administrador de permisos de DTI; cuenta con los siguientes permisos:



unl

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

- Administración de los datos de la gestión de becas
- Dar de alta o baja a los usuarios del Módulo de Gestión de Becas
- Monitoreo del Módulo de Gestión de Becas

## CAPACITACIÓN

Se ha llevado a cabo capacitaciones dirigidas al Administrador de Becas, Subdirectora de Tesorería y Subdirector de Contabilidad a continuación detalladas el día Viernes 05 de Agosto en el horario de 10H30 a 12H30, mediante reunión meet del correo [jackson.guzman@unl.edu.ec](mailto:jackson.guzman@unl.edu.ec)

## MANUALES

Se realizan los manuales con la funcionalidades en el Módulo de Gestión de Becas del Sistema de Información Académico, Administrativo y Financiero - SIAAF versión:

- [v.22.08\\_Manual\\_Usuarios\\_MGB.pdf](#)
- [v.22.08\\_Manual\\_de Configuración\\_MGB.pdf](#)

## DOCUMENTACIÓN TÉCNICA

A continuación se detalla la documentación técnica realizada para el desarrollo del Módulo de Gestión de Becas en cada una de sus fases.

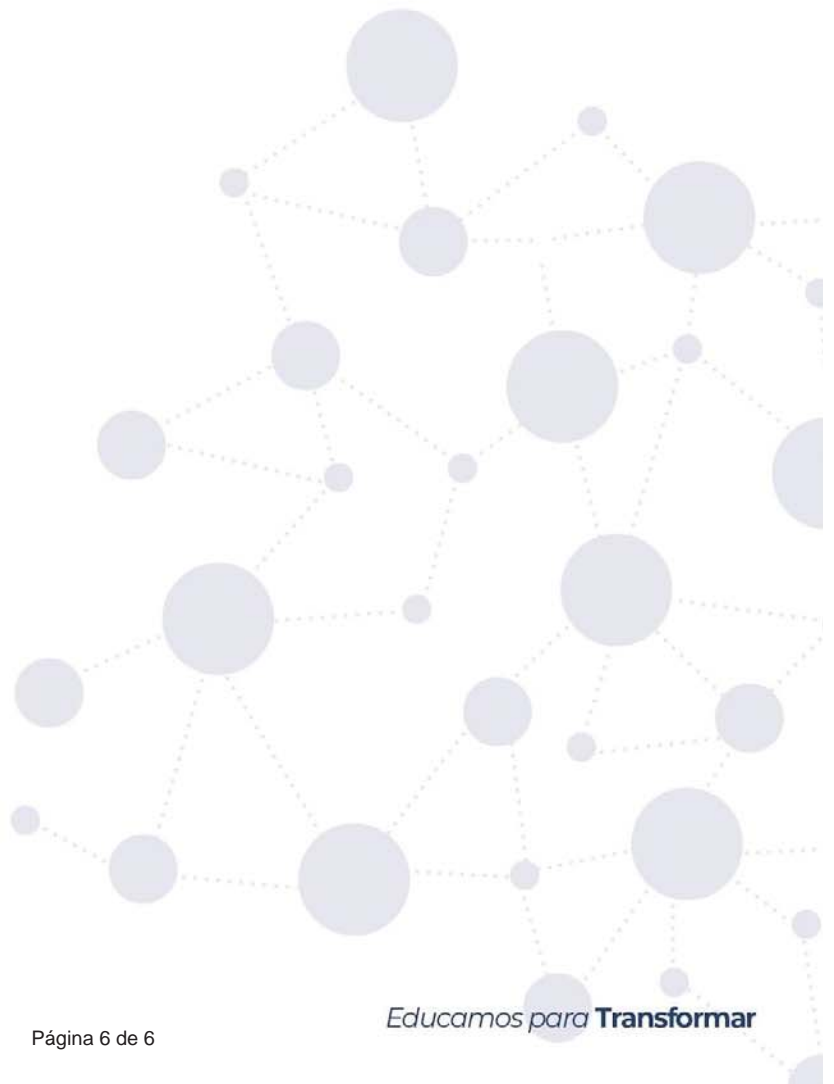
- Análisis: [Especificación de requisitos de software](#)
- Diseño y Desarrollo:
  - [Diagramas de clases](#)
  - [Diagramas de componentes](#)
  - [Diagrama de tecnologías](#)
  - [Diagrama del mapa de historia](#)
  - [Diagrama de casos de uso](#)
- Control de Calidad:
  - [Plan de pruebas](#)
  - [Ejecución del plan de pruebas](#)
- Puesta en Producción:
  - [Sprint tipo, permisos, parametrización, etapas](#)



## CONCLUSIÓN

Para la puesta en producción de la Implementación del Módulo de Gestión de Becas para Bienestar Universitario - SIAAF se validan **18** casos de uso funcionales y **10** casos de uso no funcionales. Con el [proceso de control de calidad](#) se logró analizar **60 casos de prueba**, en los que se encontraron **82 incidencias (40 mejoras, 16 recomendaciones y 26 errores)**. Todos los requerimientos fueron analizados y reportados al equipo de desarrollo. Luego de haber completado el desarrollo **de todos los requerimientos** se certifica que el servicio se encuentra listo para su uso.

Elaborado por:	Aprobado por:
 <p>Firmado electrónicamente por: <b>JOHANA ELIZABETH CUEVA DOMINGUEZ</b></p>	 <p>Firmado electrónicamente por: <b>DANNY EMANUEL MUNOZ FLORES</b></p>
<b>Analista de Sistemas Informáticos 1</b>	<b>Especialista de Sistemas de Información</b>



**Anexo 9. Comparativa entre las Bitácoras en curso y finalizada para las Pruebas de Control de Calidad aplicadas por DTI en el MGB**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Comparativa entre Bitácoras de QA en curso y finalizadas









## **Anexo 10. Aplicación de la técnica del User Story Mapping en el desarrollo ágil**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

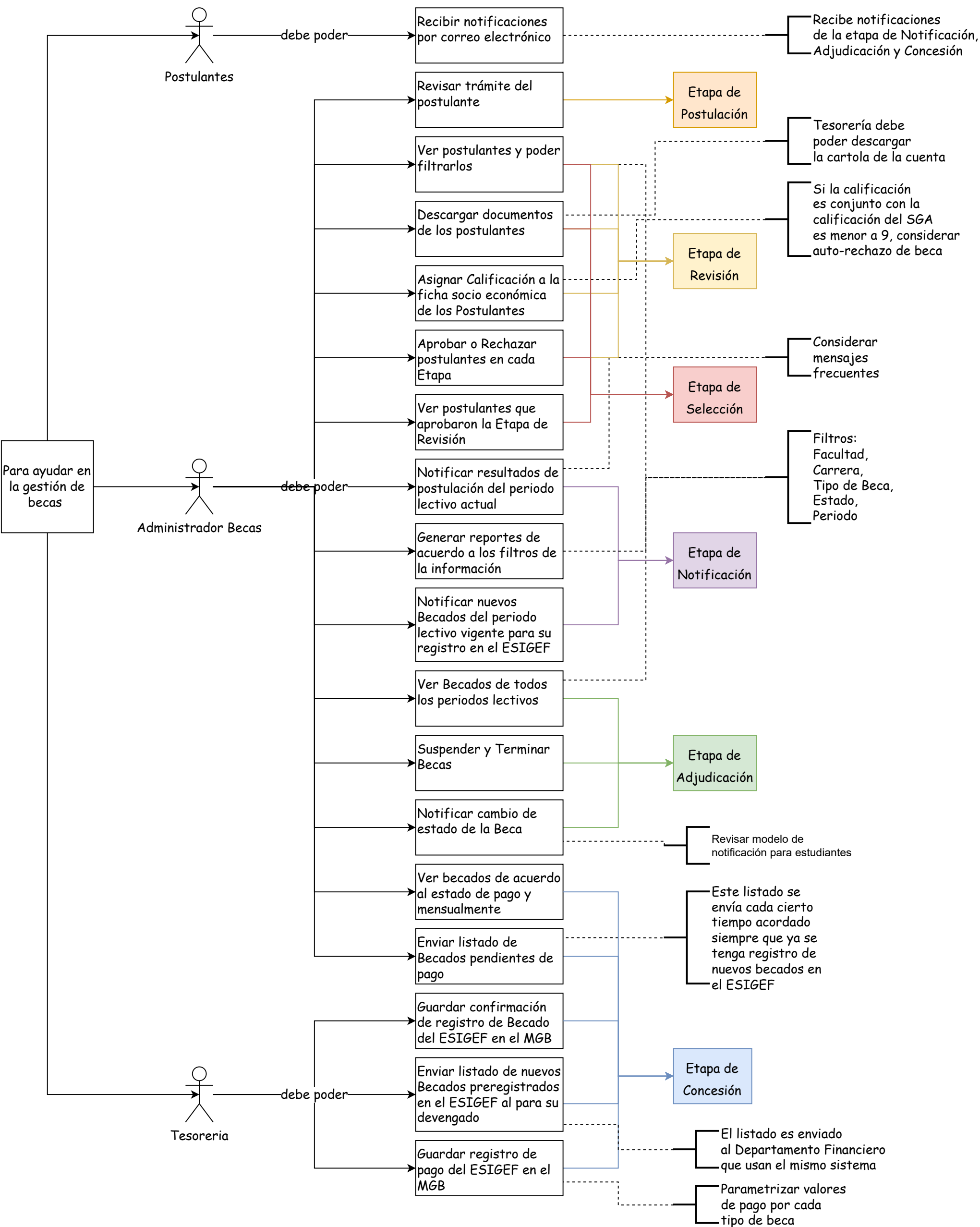
Aplicación de la técnica User Story Mapping<sup>11</sup>

---

<sup>11</sup> Acta de reunión para la validación de historias de usuario mediante la técnica User Story Mapping en: <https://drive.google.com/Acta-Validacion-Historias-Usuario.pdf>

# MAPA DE IMPACTO

**Acrónimos:**  
**MGB:** Módulo de Gestión de Becas  
**ESIGEF:** Sistema de Administración Financiera  
**SGA:** Sistema de Gestión Académica



# USER STORY MAPPING

## 1 Desglosar

## 2 Priorizar

## 3 Versionar

### Etapa de Postulación

Registrar postulantes desde el MGT

Crear enlace al trámite de cada postulante

Mostrar estadísticas generales de postulantes

### Etapa de Revisión

Listar postulantes de cada periodo lectivo

Permitir descarga de documentos de cada postulante

Asignar calificación a la ficha socio económica de cada postulante

Actualizar el estado del postulante en esta etapa

Registrar mensaje personalizado o frecuente cuando se rechaza a un postulante

Generar reportes de acuerdo a los filtros aplicados

Archivar Etapa

### Etapa de Selección

Listar postulantes que aprobaron la Etapa de Revisión de cada periodo lectivo

Permitir descarga de documentos de cada postulante

Actualizar el estado del postulante en esta etapa

Registrar mensaje personalizado o frecuente cuando se rechaza a un postulante

Generar reportes de acuerdo a los filtros aplicados

Archivar Etapa

### Etapa de Notificación

Notificar resultados de postulación una vez archivada la Etapa de Selección

Registrar mensajes frecuentes

Actualizar mensajes frecuentes

Eliminar mensajes frecuentes

Marcar como favorito un mensaje frecuente

Enviar listado de nuevos Becados del periodo lectivo vigente para su registro en el ESIGEF

### Etapa de Adjudicación

Listar becados del periodo lectivo vigente y de los anteriores

Actualizar el estado del becado en esta etapa

Notificar cambio de estado de la Beca

Generar reportes de acuerdo a los filtros aplicados

Archivar Etapa

Enlazar becados activos del periodo lectivo anterior con el periodo activo actual una vez archivada esta etapa

Mostrar estadísticas generales de becados

### Etapa de Concesión

Listar becados de acuerdo al estado de pago y el desglose de pago por cada mes

Guardar confirmación de registro de Becado del ESIGEF en el MGB

Enviar listado de nuevos Becados pre registrados en el ESIGEF junto a Becados activos del periodo lectivo en curso al

Guardar registro de pago del ESIGEF en el MGB

Guardar registro de pago del ESIGEF en el MGB

Actualizar monto de pago por cada tipo de beca

Generar reportes de acuerdo a los filtros aplicados

Archivar Etapa

# USER STORY MAPPING

1 Desglosar

2 Priorizar

3 Versionar

Etapa de Postulación

Etapa de Revisión

Etapa de Selección

Etapa de Notificación

Etapa de Adjudicación

Etapa de Concesión

TIENE QUE ESTAR

Registrar postulantes desde el MGT

Listar postulantes y becados de cada periodo lectivo por etapa

Permitir descarga de documentos de cada postulante

Asignar calificación a la ficha socio económica de cada postulante

Enviar listado de nuevos Becados del periodo lectivo vigente para su registro en el ESIGEF

Enviar listado de nuevos Becados pre registrados en el ESIGEF junto a Becados activos del periodo lectivo en curso al departamento financiero para su devengado

Actualizar el estado de postulantes y becados en cada etapa

Generar reportes de acuerdo a los filtros aplicados

Notificar resultados de postulación una vez archivada la Etapa de Selección

Notificar cambio de estado de la Beca

Listar becados de acuerdo al estado de pago y el desgloce de pago por cada mes

Guardar confirmación de registro de Becado del ESIGEF en el MGB

Guardar registro de pago del ESIGEF en el MGB

PODRÍA ESTAR

Crear enlace al trámite de cada postulante

Registrar mensaje personalizado o frecuente cuando se rechaza a un postulante

Archivar Etapa

Registrar mensajes frecuentes

Actualizar mensajes frecuentes

Marcar como favorito un mensaje frecuente

Mostrar estadísticas generales de becados

Mostrar estadísticas generales de postulantes

Eliminar mensajes frecuentes

DESCARTADO DE MOMENTO

Importar archivo con la confirmación del registro de Becados al ESIGEF en el MGB

Importar registro de pago de los Becados a través del ESIGEF en el MGB

Actualizar registro de pago por cada tipo de beca

# USER STORY MAPPING

1 Desglosar

2 Priorizar

3 Versionar

Etapa de Postulación

Etapa de Revisión

Etapa de Selección

Etapa de Notificación

Etapa de Adjudicación

Etapa de Concesión

## ·······VERSION 1······

Registrar postulantes desde el MGT

Listar postulantes de cada periodo lectivo

Permitir descarga de documentos de cada postulante

Asignar calificación a la ficha socio económica de cada postulante

Actualizar el estado de postulantes y becados en cada etapa

Crear enlace al trámite de cada postulante

Listar nuevos becados del periodo lectivo en curso para su registro en el ESIGEF

## ·······VERSION 2······

Mostrar estadísticas generales de postulantes

Notificar cambio de estado de la Beca

Listar becados de acuerdo al estado de pago y el desglose de pago por cada mes

Guardar confirmación de registro de Becado del ESIGEF en el MGB

Notificar resultados de postulación una vez archivada la Etapa de Selección

Enviar listado de nuevos Becados del periodo lectivo vigente para su registro en el ESIGEF

Enviar listado de nuevos Becados pre registrados en el ESIGEF junto a Becados activos del periodo lectivo en curso al departamento financiero para su devengado

## ·······VERSION 3······

Archivar Etapas

Importar archivo con la confirmación del registro de Becados al ESIGEF en el MGB

Registrar mensaje personalizado o frecuente cuando se rechaza a un postulante

Generar reportes de acuerdo a los filtros aplicados

Registrar mensajes frecuentes

Eliminar mensajes frecuentes

Actualizar mensajes frecuentes

Marcar como favorito un mensaje frecuente

Guardar registro de pago del ESIGEF en el MGB

Importar registro de pago de los Becados a través del ESIGEF en el MGB

Actualizar registro de pago por cada tipo de beca

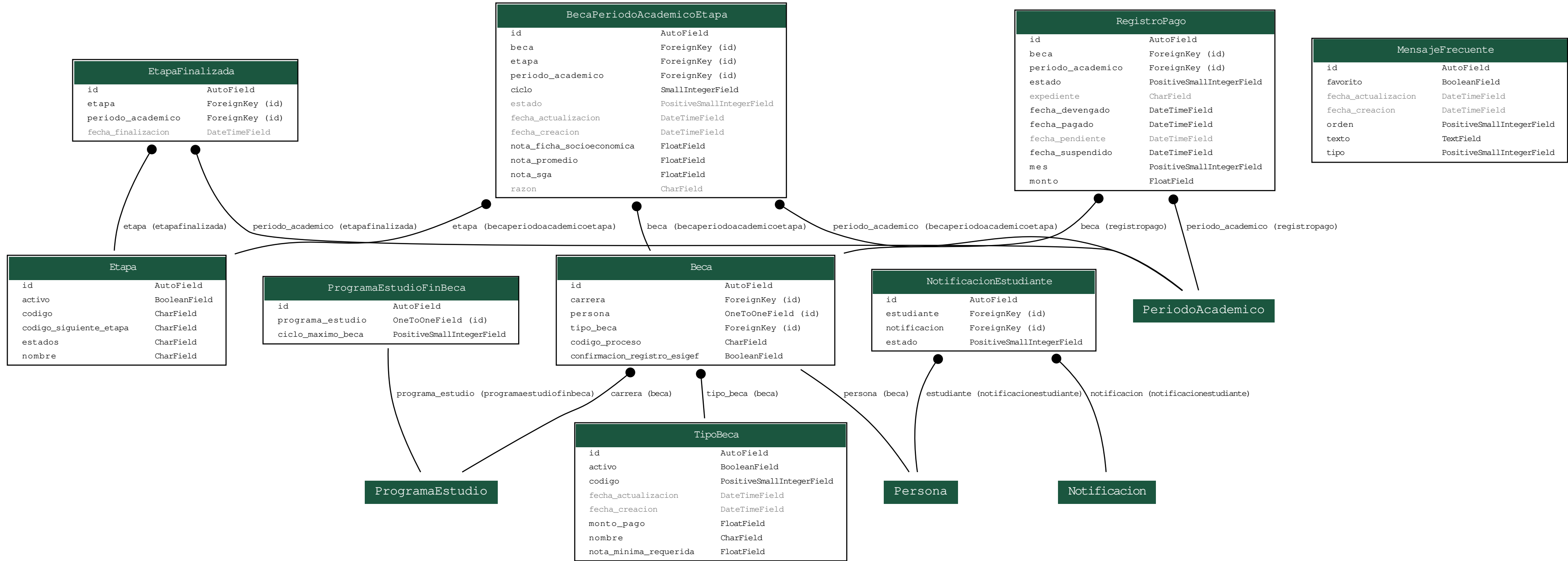
**Anexo 11. Modelo de Base de Datos generado usando el ORM de Django a partir del  
Diagrama de Clases para el MGB**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Modelo de Base de Datos





## **Anexo 12. Análisis de Datos obtenidos a partir de las encuestas aplicadas a cada uno de los usuarios del MGB**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

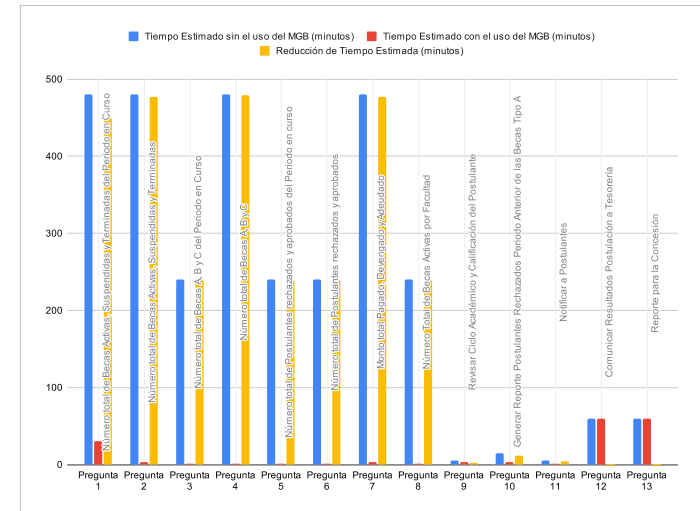
Análisis de Datos obtenidos de las encuestas de Aceptación<sup>12</sup>

---

<sup>12</sup> Véase encuestas y actas de reunión firmadas para las pruebas de aceptación en: <https://drive.google.com/Pruebas-Aceptacion>

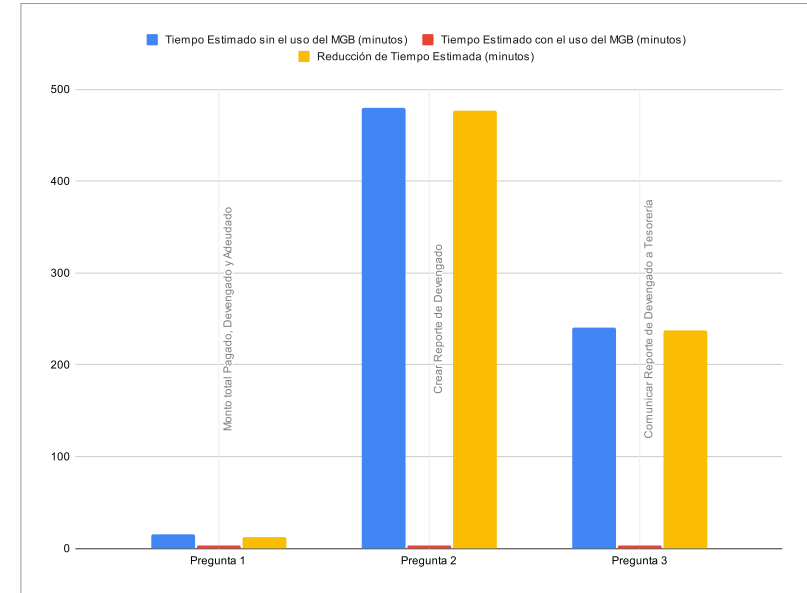
Pregunta aplicadas antes del uso del MGB	Pregunta aplicadas después del uso del MGB	Sin el uso del MGB (tiempo estimado)	Con el uso del MGB (tiempo estimado)	Evaluated	Identificador	Indicador Evaluado	Tiempo Estimado sin el uso del MGB (minutos)	Tiempo Estimado con el uso del MGB (minutos)	Reducción de Tiempo Estimada (minutos)
¿Qué tiempo considera que le tomaría dar con precisión el número total de becas activas, suspendidas y terminadas del periodo en curso?	Con el uso del módulo ¿qué tiempo cree que le tome dar el número exacto de becas activas, suspendidas y terminadas del periodo en curso?	Más de un día laboral	Entre 30 minutos y 1 hora	marlene.valdez@unl.edu.ec	Pregunta 1	Número total de Becas Activas, Suspendidas y Terminadas del Periodo en Curso	480	30	450
¿Qué tiempo considera que le tomaría dar un aproximado del número total de becas activas, suspendidas y terminadas de todos los periodos académicos?	Con el uso del módulo ¿qué tiempo considera que le tomaría dar con precisión el número total de becas activas, suspendidas y terminadas de todos los periodos académicos?	Más de un día laboral	Entre 3 a 5 minutos	marlene.valdez@unl.edu.ec	Pregunta 2	Número total de Becas Activas, Suspendidas y Terminadas	480	3	477
¿Qué tiempo considera que le tomaría dar un aproximado del número total de becas de acuerdo a su tipo (A, B y C) para el periodo en curso?	Con el uso del módulo desarrollado ¿qué tiempo considera que le tomaría dar un aproximado del número total de becas de acuerdo a su tipo (A, B y C) para el periodo en curso?	Entre 4 a 8 horas	Entre 1 a 3 minutos	marlene.valdez@unl.edu.ec	Pregunta 3	Número total de Becas A, B y C del Periodo en Curso	240	1	239
¿Qué tiempo considera que le tomaría dar un aproximado del número total de becas de todos los periodos académicos de acuerdo a su tipo (A, B y C)?	Con el uso del módulo desarrollado ¿qué tiempo considera que le tomaría dar un aproximado del número total de becas de todos los periodos académicos de acuerdo a su tipo (A, B y C)?	Más de un día laboral	Entre 1 a 3 minutos	marlene.valdez@unl.edu.ec	Pregunta 4	Número total de Becas A, B y C	480	1	479
¿Qué tiempo considera que le tomaría dar un aproximado del número total de postulantes rechazados y postulantes aprobados para el periodo en curso?	Con el uso del módulo desarrollado ¿qué tiempo considera que le tomaría dar un aproximado del número total de postulantes rechazados y postulantes aprobados para el periodo en curso?	Entre 4 a 8 horas	Entre 1 a 3 minutos	marlene.valdez@unl.edu.ec	Pregunta 5	Número total de Postulantes rechazados y aprobados del Periodo en curso	240	1	239
¿Qué tiempo considera que le tomaría dar un aproximado del número total de postulantes rechazados y aprobados de todos los periodos?	Con el uso del módulo desarrollado ¿qué tiempo considera que le tomaría dar un aproximado del número total de postulantes rechazados y postulantes aprobados de todos los periodos?	Entre 4 a 8 horas	Entre 1 a 3 minutos	marlene.valdez@unl.edu.ec	Pregunta 6	Número total de Postulantes rechazados y aprobados	240	1	239
¿Qué tiempo considera que le tomaría dar un aproximado del monto total pagado, devengado y adeudado de todos los periodos?	Con el uso del módulo desarrollado ¿qué tiempo considera que le tomaría dar un aproximado del monto total pagado, devengado y adeudado de todos los periodos?	Más de un día laboral	Entre 3 a 5 minutos	marlene.valdez@unl.edu.ec	Pregunta 7	Monto total Pagado, Devengado y Adeudado	480	3	477
¿Qué tiempo considera que le tomaría dar un aproximado del número total de becas activas por facultad de todos los periodos?	Con el uso del módulo desarrollado ¿qué tiempo considera que le tomaría dar un aproximado del número total de becas activas por facultad de todos los periodos?	Entre 4 a 8 horas	Entre 1 a 3 minutos	marlene.valdez@unl.edu.ec	Pregunta 8	Número Total de Becas Activas por Facultad	240	1	239
¿Qué tiempo considera que le toma revisar los datos del postulante como: el ciclo académico y la calificación del postulante en el SGA?	Con el uso del módulo desarrollado ¿qué tiempo considera que le toma revisar los datos del postulante como: el ciclo académico y la calificación del postulante en el MGB?	Entre 5 y 15 minutos	Entre 3 a 5 minutos	marlene.valdez@unl.edu.ec	Pregunta 9	Revisar Ciclo Académico y Calificación del Postulante	5	3	2
¿Qué tiempo considera que le toma generar un reporte de los postulantes que fueron rechazados en un periodo académico pasado del tipo de beca A?	Con el uso del módulo desarrollado ¿qué tiempo considera que le toma generar un reporte de los postulantes que fueron rechazados en un periodo académico pasado del tipo de beca A?	Entre 15 y 30 minutos	Entre 3 a 5 minutos	marlene.valdez@unl.edu.ec	Pregunta 10	Generar Reporte Postulantes Rechazados Periodo Anterior de las Becas Tipo A	15	3	12
¿Qué tiempo considera que le toma notificar a los postulantes que fueron aprobados en el periodo académico en curso?	Con el uso del módulo desarrollado ¿qué tiempo considera que le toma notificar a los postulantes que fueron aprobados en el periodo académico en curso?	Entre 5 y 15 minutos	Entre 1 a 3 minutos	marlene.valdez@unl.edu.ec	Pregunta 11	Notificar a Postulantes	5	1	4
¿Qué tiempo cree que le toma comunicar los resultados de la postulación a Tesorería?	Con el uso del módulo desarrollado ¿qué tiempo cree que le toma comunicar los resultados de la postulación a Tesorería?	Entre 1 y 2 horas	Entre 1 y 2 horas	marlene.valdez@unl.edu.ec	Pregunta 12	Comunicar Resultados Postulación a Tesorería	60	60	0
¿Qué tiempo cree que le toma crear el reporte para la concesión de becas activas?	Con el uso del módulo desarrollado ¿qué tiempo cree que le toma crear el reporte para la concesión de becas activas?	Entre 1 y 2 horas	Entre 1 y 2 horas	marlene.valdez@unl.edu.ec	Pregunta 13	Reporte para la Concesión	60	60	0

Escala de Tiempo Considerada	
Rangos	Valor mínimo
Menos de 1 minuto	0.5min
Entre 1 a 3 minutos	1min
Entre 3 a 5 minutos	3min
Entre 5 y 15 minutos	5min
Entre 15 y 30 minutos	15min
Entre 30 minutos y 1 hora	30min
Entre 1 y 2 horas	60min
Entre 2 a 4 horas	120min
Entre 4 a 8 horas	240min
Más de un día laboral	480min



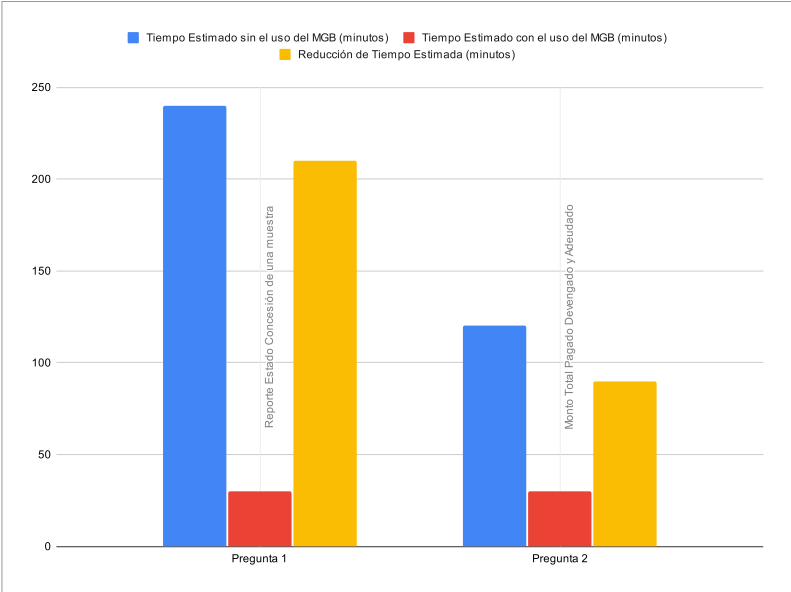
Pregunta aplicadas antes del uso del MGB	Pregunta aplicadas después del uso del MGB	Sin el uso del MGB (tiempo estimado)	Con el uso del MGB (tiempo estimado)	Evaluado	Identificador	Indicador Evaluado	Tiempo Estimado sin el uso del MGB (minutos)	Tiempo Estimado con el uso del MGB (minutos)	Reducción de Tiempo Estimada (minutos)
¿Qué tiempo cree que le tome informar con precisión el monto total de pago, devengado y adeudado?	Con el uso del módulo desarrollado ¿Qué tiempo considera que le tomaría informar con precisión el monto total pagado, devengado y adeudado?	De 15 a 30 minutos	De 3 a 5 minutos	jvarela@unl.edu.ec	Pregunta 1	Monto total Pagado, Devengado y Adeudado	15	3	12
¿Qué tiempo cree que le toma crear el reporte para el devengado de becas para que Tesorería se encargue del registro?	Con el uso del módulo desarrollado ¿Qué tiempo cree que le tomaría crear el reporte para el devengado de becas para que Tesorería se encargue del registro?	Más de 1 día laboral	De 3 a 5 minutos	jvarela@unl.edu.ec	Pregunta 2	Crear Reporte de Devengado	480	3	477
¿Qué tiempo cree que le toma comunicar a Tesorería el reporte de devengado o pago?	Con el uso del módulo desarrollado ¿Qué tiempo cree que le toma comunicar a Tesorería el reporte de devengado o pago?	De 4 a 8 horas	De 3 a 5 minutos	jvarela@unl.edu.ec	Pregunta 3	Comunicar Reporte de Devengado a Tesorería	240	3	237

Escala de Tiempo Considerada	
Rangos	Valor mínimo
Menos de 1 minuto	0,5min
Entre 1 a 3 minutos	1min
Entre 3 a 5 minutos	3min
Entre 5 y 15 minutos	5min
Entre 15 y 30 minutos	15min
Entre 30 minutos y 1 hora	30min
Entre 1 y 2 horas	60min
Entre 2 a 4 horas	120min
Entre 4 a 8 horas	240min
Más de un día laboral	480min



Pregunta antes	Pregunta después	Sin el uso del MGB (tiempo estimado)	Con el uso del MGB (tiempo estimado)	Evaluado	Identificador	Indicador Evaluado	Tiempo Estimado sin el uso del MGB (minutos)	Tiempo Estimado con el uso del MGB (minutos)	Reducción de Tiempo Estimada (minutos)
Para una muestra de 100 becas ¿Qué tiempo considera que le tome armar el reporte con el estado de la concesión de dichas becas y comunicarlo a la Unidad de Bienestar Universitario?	Con el uso del módulo desarrollado para una muestra de 100 becas. ¿qué tiempo considera que le tome armar el reporte con el estado de la concesión de dichas becas y comunicarlo a la Unidad de Bienestar Universitario?	Entre 4 a 8 horas	Entre 30 minutos y 1 h	monica.r.mora@unl.edu.ec	Pregunta 1	Reporte Estado Concesión de una muestra	240	30	210
Con el uso del módulo desarrollado ¿qué tiempo considera que le tomaría informar el monto total pagado, devengado y adeudado de todos los periodos?	Con el uso del módulo desarrollado ¿qué tiempo considera que le tomaría informar el monto total pagado, devengado y adeudado de todos los periodos?	Entre 2 a 4 horas	Entre 30 minutos y 1 h	monica.r.mora@unl.edu.ec	Pregunta 2	Monto Total Pagado Devengado y Adeudado	120	30	90

Escala de Tiempo Considerada	
Rangos	Valor mínimo
Menos de 1 minuto	0,5min
Entre 1 a 3 minutos	1min
Entre 3 a 5 minutos	3min
Entre 5 y 15 minutos	5min
Entre 15 y 30 minutos	15min
Entre 30 minutos y 1 hora	30min
Entre 1 y 2 horas	60min
Entre 2 a 4 horas	120min
Entre 4 a 8 horas	240min
Más de un día laboral	480min



## **Anexo 13. Acta de reunión para la validación de requisitos de software con DTI**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Acta de reunión para la validación de los Requisitos de Software con DTI<sup>13</sup>

---

<sup>13</sup> Versión original en: <https://drive.google.com/Acta-Validacion-DTI.pdf>

	<b>ACTA DE REUNIÓN</b>	Cód.: 001
		Organización:
		UTI & UBU - UNL

ACTA	1	TEMA	VALIDACIÓN DE REQUISITOS DEL MÓDULO PARA EL SEGUIMIENTO Y CONTROL DE BECAS DE LA UNIDAD DE BIENESTAR UNIVERSITARIO	D	M	A
				08	12	2021

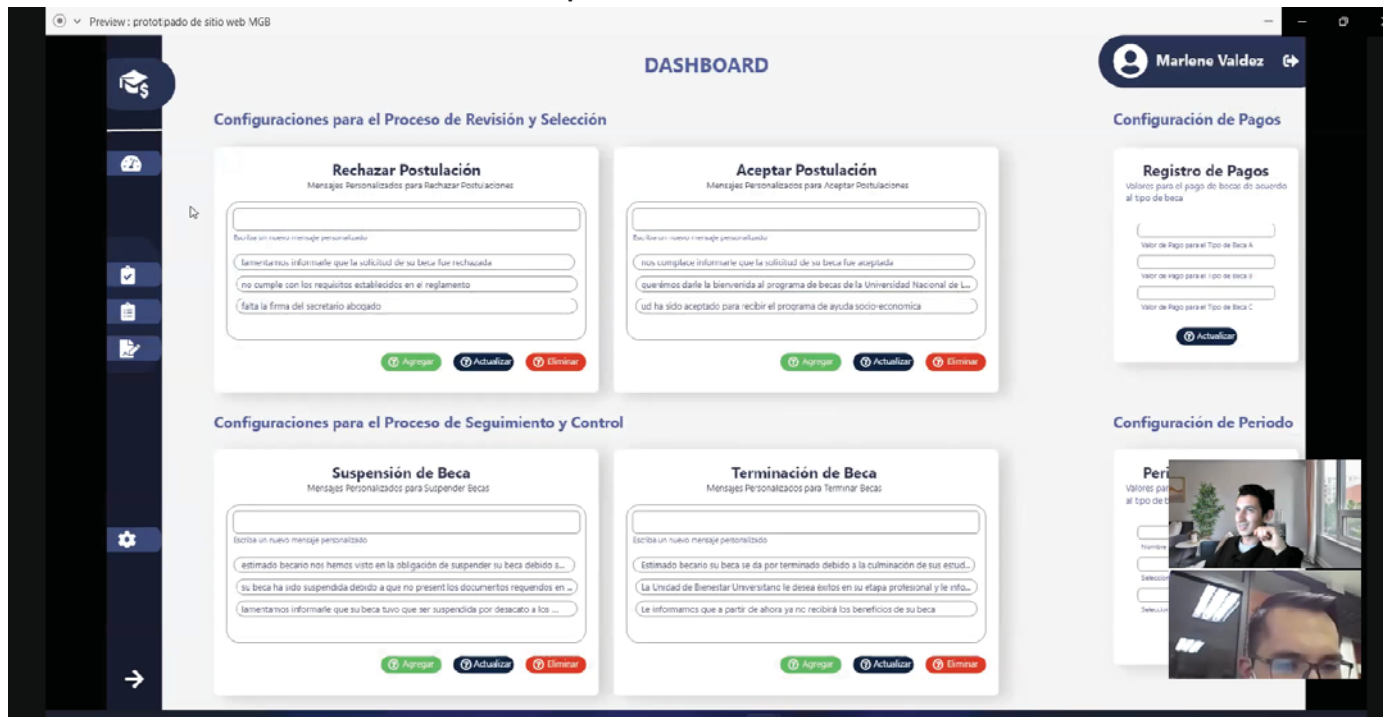
<b>LUGAR:</b>	Universidad Nacional de Loja - Reunión Virtual Via Zoom	
<b>HORA:</b>	17:00	
<b>OBJETIVO</b>	Validar los requisitos de software de la mano de los interesados en el desarrollo.	
<b>PARTICIPANTES</b>		
<ol style="list-style-type: none"> <li>1. Ing. Máximo Alvarez.</li> <li>2. Lcda. Marlene Vaaldez.</li> <li>3. Tesista Jackson Guzmán.</li> </ol>		
<b>ORDEN DEL DÍA</b>		
<ol style="list-style-type: none"> <li>1. Socializar el levantamiento de requisitos.</li> <li>2. Validar el funcionamiento del software mediante la revisión del prototipo de interfaces de usuario y el levantamiento de requisitos mediante el estándar IEEE.</li> <li>3. Definir cambios a considerar para su implementación.</li> </ol>		
<b>DESARROLLO DE LA REUNIÓN</b>		
Presentación del flujo actual de procesos desde el Módulo de Gestión de Trámites con el proceso de postulación de requisitos. Presentación del prototipo diseñado para el Módulo de Gestión de Becas Análisis del prototipo propuesto presentado.		
<b>CIERRE</b>		
Recopilación de acuerdo y compromisos por parte del tesista.		
<b>ACUERDOS / COMPROMISOS</b>		
TAREAS	RESPONSABLE	FECHA
Crear vista de inicio de sesión basado en el SIAAF	Jackson Guzmán	13/12/2021
Detallar en el dashboard para indicar que se trata del periodo activo	Jackson Guzmán	10/12/2021
Crear una gráfica que muestre el total de becados por año académico	Jackson Guzmán	10/12/2021
No enviar el campo modalidad en el API ya que todos deben ser presencial	Máximo Alvarez	
Para el api de consulta de postulantes enviar el inicio y fin del año académico para que devuelva todos los periodos de postulación existentes.	Jackson Guzmán	
Consultar acerca de la matriz de calificación de requisitos con la Lcda. Marlene	Jackson Guzmán	13/12/2021
Asignar estado activo y finalizado a los años académicos	Jackson Guzmán	12/12/2021
Analizar incluir reporte en porcentajes de becas disponibles por carrera	Jackson Guzmán	15/12/2021

 Universidad Nacional de Loja	<b>ACTA DE REUNIÓN</b>	Cód.: 001
		Organización:
		UTI & UBU - UNL

Enviar el ciclo actual del postulante y la calificación del último ciclo aprobado, en el api para su consumo en la etapa de selección y concesión	Máximo Alvarez	
Hacer la propuesta al Departamento Financiero de importar reporte de pagos para actualizar los pagos en el MGB	Jackson Guzmán	
Analizar no mostrar los campos de de facultad carrera y régimen para la vista de pagos, pero si en reportes	Jackson Guzmán	

### EVIDENCIA



#### Captura de Reunión Zoom



PRÓXIMA REUNIÓN	FECHA	HORA	LUGAR

*Se dio por terminada esta actividad, habiéndose desarrollado los temas contenidos en esta acta y comprometiéndose las partes a cumplir con los compromisos adquiridos en la misma.*

### FIRMA DE CONSTANCIA

POR PARTE DE TESISTA	POR BENEFICIARIOS
	 <p>Firmado electrónicamente por: <b>MAXIMO ANDRES ALVAREZ PACHECO</b></p>

**Anexo 14. Acta de Reunión para la Validación de Requisitos con el representante de la UBU en calidad de cliente**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Acta de reunión para la validación de requisitos de software con UBU<sup>14</sup>

---

<sup>14</sup> Versión original en: <https://drive.google.com/Acta-Validacion-UBU.pdf>



	<b>ACTA DE REUNIÓN</b>	Cód.: 001
		Organización:
		UTI & UBU - UNL

ACTA	1	TEMA	VALIDACIÓN DE REQUISITOS DEL MÓDULO PARA EL SEGUIMIENTO Y CONTROL DE BECAS DE LA UNIDAD DE BIENESTAR UNIVERSITARIO	D	M	A
				13	12	2021

<b>LUGAR:</b>	Universidad Nacional de Loja - Reunión Virtual Via Zoom	
<b>HORA:</b>	16:00	
<b>OBJETIVO</b>	Validar los requisitos de software de la mano de los interesados en el desarrollo.	
<b>PARTICIPANTES</b>		
<ol style="list-style-type: none"> <li>Lcda. Marlene Vaaldez.</li> <li>Tesista Jackson Guzmán.</li> </ol>		
<b>ORDEN DEL DÍA</b>		
<ol style="list-style-type: none"> <li>Socializar el levantamiento de requisitos.</li> <li>Validar el funcionamiento del software mediante la revisión del prototipo de interfaces de usuario y el levantamiento de requisitos mediante el estándar IEEE.</li> <li>Definir cambios a considerar para su implementación.</li> </ol>		
<b>DESARROLLO DE LA REUNIÓN</b>		
Presentación del flujo actual de procesos desde el Módulo de Gestión de Trámites con el proceso de postulación de requisitos. Presentación del prototipo diseñado para el Módulo de Gestión de Becas Análisis del prototipo propuesto presentado.		
<b>CIERRE</b>		
Recopilación de acuerdo y compromisos por parte del tesista.		
<b>ACUERDOS / COMPROMISOS</b>		
TAREAS	RESPONSABLE	FECHA
Crear vista de inicio de sesión basado en el SIAAF	Jackson Guzmán	17/12/2021
Eliminar el campo de régimen académico ya que no es necesario en ninguna de las etapas	Jackson Guzmán	17/12/2021
Crear una gráfica que muestre el total de becados por año académico	Jackson Guzmán	17/12/2021
Todas las notificación debe hacerse al finalizar la etapa de selección	Jackson Guzmán	13/12/2021
Solicitar reunión con Lcda Nancy León encargada del registro de pagos del dto. financiero	Jackson Guzmán	14/12/2021
La matriz de calificación de requisitos se la hace de manera interna y no debe contemplarse en el sistema ya que existen muchos parámetros y depende de cada caso del postulante	Jackson Guzmán	13/12/2021
Analizar si es necesario agregar un checklist por cada documento para indicar si esta bien o mal.	Jackson Guzmán	17/12/2021
Analizar incluir reporte en porcentajes de becas disponibles por carrera de manera informativa ya que mientras existan cupos disponibles esto no es una limitante	Jackson Guzmán	17/12/2021

 Universidad Nacional de Loja	<b>ACTA DE REUNIÓN</b>	Cód.: 001
		Organización:
		UTI & UBU - UNL

Enviar el ciclo actual del postulante y la calificación del último ciclo aprobado, en el api para su consumo en la etapa de revisión, selección y adjudicación	Máximo Alvarez	
Hacer la propuesta al Departamento Financiero de importar reporte de pagos para actualizar los pagos en el MGB	Jackson Guzmán	
Mostrar los campos de facultad carrera para la vista de pagos	Jackson Guzmán	13/12/2021
Los certificados de aprobación de ciclo inmediato anterior y certificado de matrícula de ciclo inmediato superior ya no son requeridos, pero analizar el consumo de esta información ya que desde este año académico se está revisando en el SGA.	Jackson Guzmán	22/12/2021
Una beca es terminada cuando tiene nota menor a 8.5, los estudiantes de enfermería si el ciclo es superior a 6to ciclo y medicina si el ciclo es superior a 9no o por parte del becado.	Jackson Guzmán	24/12/2021

### EVIDENCIA

#### Captura de Reunión Zoom



Preview : prototipado de sitio web MGB

**ETAPA DE SELECCIÓN DE POSTULANTES**

Marlene Valdez

[ACTUALIZAR ESTADO](#)
[DESCARGAR DOCUMENTOS](#)
[VER TRÁMITE](#)

Postulante	Cédula	Celular	Correo	Facultad	Carrera	Régimen	Tipo Beca	Estado	Nota EVA/UBU
JHULEIDY DELIA ALLUMA GUAMAN	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	FJSA	DERECHO	2019	Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	FJSA	DERECHO	2019	Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	FJSA	DERECHO	2019	Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	FJSA	DERECHO	2019	Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	FJSA	DERECHO	2019	Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	FJSA	DERECHO	2019	Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	FJSA	DERECHO	2019	Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	FJSA	DERECHO	2019	Tipo A	Sin Revisar	8.9
JHULEIDY DELIA ALLUMA GUAMAN	1950025617	0969689022	jhuleidy.alluma@unl.edu.ec	FJSA	DERECHO	2019	Tipo A	Sin Revisar	8.9

Mostrando 10 resultados de 250

Estados: Sin Revisar, Revisando, Aprobado, Rechazado

[DESCARGAR REPORTE](#)

PRÓXIMA REUNIÓN	FECHA	HORA	LUGAR

*Se dio por terminada esta actividad, habiéndose desarrollado los temas contenidos en esta acta y comprometiéndose las partes a cumplir con los compromisos adquiridos en la misma.*

 <b>unl</b> <small>1859</small>	Universidad Nacional de Loja	<b>ACTA DE REUNIÓN</b>	Cód.: 001
			Organización:
			UTI & UBU - UNL

**FIRMA DE CONSTANCIA**

**POR PARTE DE TESISTA**



**POR BENEFICIARIOS**



Firmado electrónicamente por:  
**CARMEN MARLENE  
 VALDEZ PARDO**

**Anexo 15. Acta de reunión con el equipo de desarrollo de la DTI para la validación del Diagrama de Clases a usarse en el MGB**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Acta de reunión para la validación del diagrama de clases con DTI<sup>15</sup>

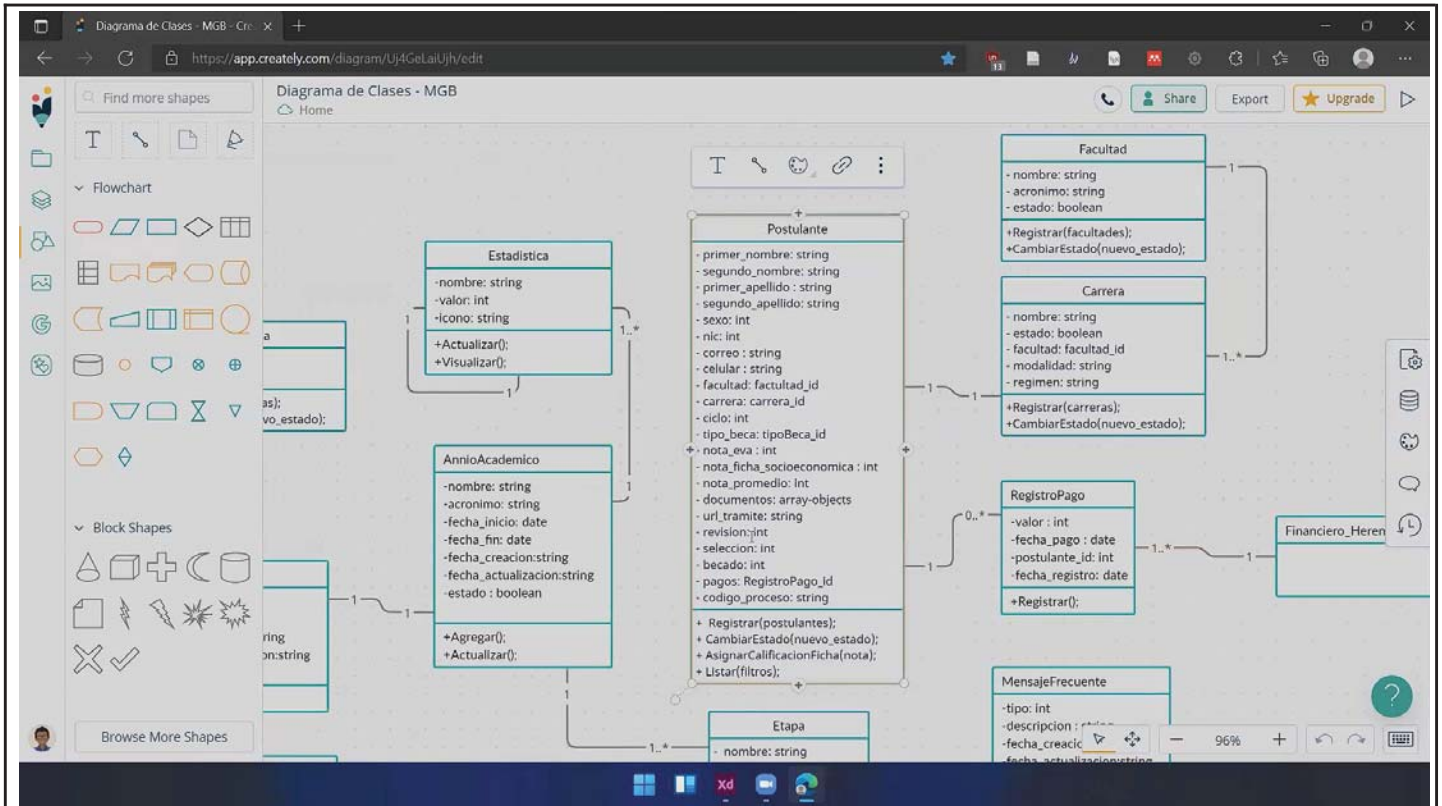
---

<sup>15</sup> Versión original en: <https://drive.google.com/Acta-Validacion-DC-MGB-DTI.pdf>

	<b>ACTA DE REUNIÓN</b>	Cód.: 002
		Organización:
		UTI - UNL

ACTA	1	TEMA	VALIDACIÓN DE BASE DE DATOS DEL MÓDULO PARA EL SEGUIMIENTO Y CONTROL DE BECAS DE LA UNIDAD DE BIENESTAR UNIVERSITARIO	D	M	A
				17	12	2021

<b>LUGAR:</b>	Universidad Nacional de Loja - Reunión Virtual Via Zoom	
<b>HORA:</b>	09:30	
<b>OBJETIVO</b>	Validar la Base de Datos para el Módulo de Gestión de Becas con el SCRUM TEAM de UTI para empezar con el desarrollo del módulo informático.	
<b>PARTICIPANTES</b>		
<ol style="list-style-type: none"> <li>1. Ing. José Martínez.</li> <li>2. Ing. Lissette Lopez</li> <li>3. Ing. Máximo Álvarez.</li> <li>4. Tesista Jackson Guzmán.</li> </ol>		
<b>ORDEN DEL DÍA</b>		
<ol style="list-style-type: none"> <li>1. Contextualizar el proyecto y socializar el objetivo que se pretende alcanzar con el mismo.</li> <li>2. Socializar la base de datos propuesta para el módulo de gestión de becas.</li> <li>3. Analizar el flujo de datos y las relaciones existentes.</li> <li>4. Tomar las consideraciones previstas durante el análisis del flujo de datos.</li> </ol>		
<b>DESARROLLO DE LA REUNIÓN</b>		
<p>Socializar el objetivo que se pretende alcanzar con el proyecto.</p> <p>Análisis del flujo de datos y las relaciones existentes.</p> <p>Considerar los cambios que se sugieren durante el análisis.</p>		
<b>CIERRE</b>		
Recopilación de acuerdo y compromisos por parte del tesista.		
<b>ACUERDOS / COMPROMISOS</b>		
<b>TAREAS</b>	<b>RESPONSABLE</b>	<b>FECHA</b>
Tomar en consideración todos los cambios sugeridos durante el análisis de la Base de Datos.	Jackson Guzmán	17/12/2021
Pasar a limpio el diagrama de Clases de la Base de Datos propuesta por el tesista.	Jackson Guzmán	20/12/2021
<b>EVIDENCIA</b>		
Captura de Reunión Zoom		



PRÓXIMA REUNIÓN	FECHA	HORA	LUGAR

Se dio por terminada esta actividad, habiéndose desarrollado los temas contenidos en esta acta y comprometiéndose las partes a cumplir con los compromisos adquiridos en la misma.

FIRMA DE CONSTANCIA

POR PARTE DE TESISISTA

**Jackson Guzmán**  
 Digitally signed by Jackson Guzmán  
 Date: 2022.02.08 00:02:30 -05'00'

Tesista Jackson Guzmán

POR BENEFICIARIOS



Firmado electrónicamente por:  
**JOSE JAVIER MARTINEZ OCHOA**

Ing. José Martínez.

POR BENEFICIARIOS



Firmado electrónicamente por:  
**LISSETTE GEOCONDA LOPEZ FAICAN**

Ing. Lissette Lopez



Firmado electrónicamente por:  
**MAXIMO ANDRES ALVAREZ PACHECO**

Ing. Máximo Álvarez.

**Anexo 16. Certificado proporcionado por la DTI que expresa la necesidad de automatizar los procesos de becas**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Certificado emitido por DTI que expresa la necesidad de automatizar los procesos de becas<sup>16</sup>

---

<sup>16</sup> Versión original en: <https://drive.google.com/Certificado-Requerimiento-Proyecto-MGB.pdf>



unl

Universidad  
Nacional  
de Loja

Dirección de  
Tecnologías de Información

## DIRECTOR DE TECNOLOGÍAS DE INFORMACIÓN

### CERTIFICA:

Que el Órgano Colegiado Superior y Dirección de Bienestar Universitario indicaron la necesidad de la automatización de los procesos de becas; acorde al requerimiento en los tickets **24587** y **32102** del 2020-10-06 y 2021-07-21 respectivamente, de la Dra. Cecilia Rocío Ruíz Toledo; Ticket **27626** del 2021-02-11, en donde se da a conocer la resolución **“SE-No.02-ROCS-No-05-08-02-2021”** del 8 de Febrero de 2021 en la que se aprueba el **“REGLAMENTO DE LA SECCIÓN DE BECAS E INCENTIVOS DE BIENESTAR UNIVERSITARIO DE LA UNIVERSIDAD NACIONAL DE LOJA”**.

Es cuanto puedo certificar, en honor a la verdad, facultando al interesado, hacer uso del presente documento.

Loja, 20 de septiembre de 2021



Firmado electrónicamente por:  
JHON ALEXANDER  
CALDERON  
SANMARTIN

Jhon Alexander Calderón Sanmartín  
**DIRECTOR DE TECNOLOGÍAS DE INFORMACIÓN**



**Anexo 17. Certificado que avala la realización de entrevistas para la captura de requisitos de software**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Certificado que avala las entrevistas realizadas para la captura de requisitos de software<sup>17</sup>

---

<sup>17</sup> Versión original en: <https://drive.google.com/Certificado-Entrevistas-UBU.pdf>



**unl**

Universidad  
Nacional  
de Loja

Bienestar  
Universitario

Licenciada Marlene Valdez Pardo. – **TRABAJADORA SOCIAL- RESPONSABLE DE LA SECCION DE BECAS E INCENTIVOS DE LA UNIDAD DE BIENESTAR UNIVERSITARIO- UNIVERSIDAD NACIONAL DE LOJA.**

**I N F O R M O:**

Que los días 04 de agosto de 2021 a las 17h00, y 30 de agosto de 2021 a las 16h00, se llevó a cabo una reunión de trabajo con el señor Jackson Andrews Guzmán Tituaña, estudiante de la carrera de Ingeniería en Sistemas, con la finalidad de dar a conocer sobre los procesos que interviene en la gestión de becas.

Lo informo. -

Loja, 20 de septiembre de 2021

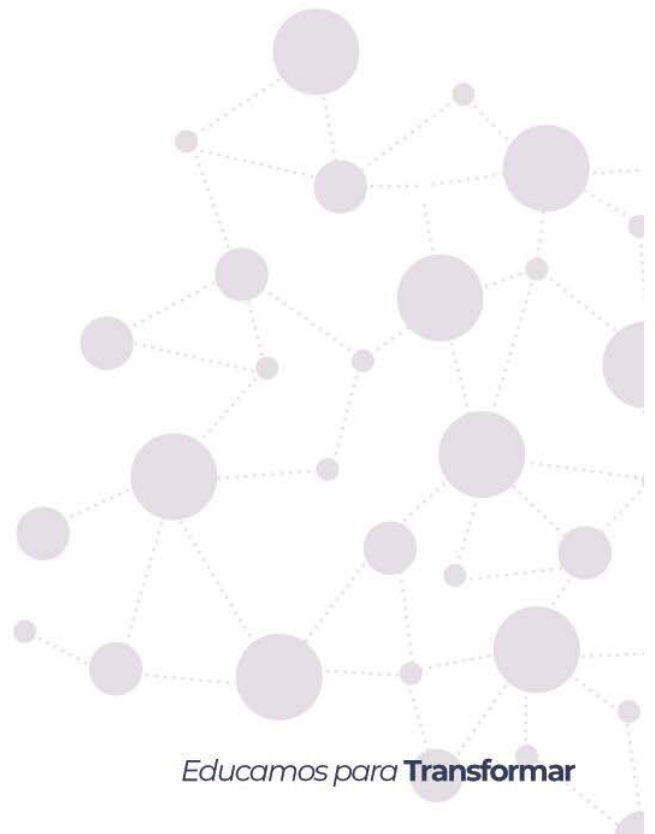


Firmado electrónicamente por:  
**CARMEN MARLENE  
VALDEZ PARDO**

Lic. Marlene Valdez Pardo.

**TRABAJADORA SOCIAL-RESPONSABLE DE LA SECCION DE BECAS E INCENTIVOS  
DE LA UNIDAD DE BIENESTAR UNIVERSITARIO-UNL.**

cc.- archivo.



## **Anexo 18. Certificado proporcionado por la DTI para el uso de Scrum en el proyecto**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Certificado emitido por DTI para el uso de SCRUM en el proyecto<sup>18</sup>

---

<sup>18</sup> Versión original en: <https://drive.google.com/Certificado-UTI-SCRUM.pdf>



## DIRECTOR DE TECNOLOGÍAS DE INFORMACIÓN

### CERTIFICA:

Que la cartera de proyectos de desarrollo de software de la Dirección de Tecnologías de Información se viene ejecutando con en el marco de trabajo scrum, en las fases de inicio, planificación y estimación, implementación, revisión y retrospectiva y lanzamiento; así mismo, de ser necesario se puede combinar con una metodología ágil dependiendo de las necesidades de cada proyecto.

Es cuanto puedo certificar, en honor a la verdad, facultando al interesado, hacer uso del presente documento.

Loja, 17 de septiembre de 2021

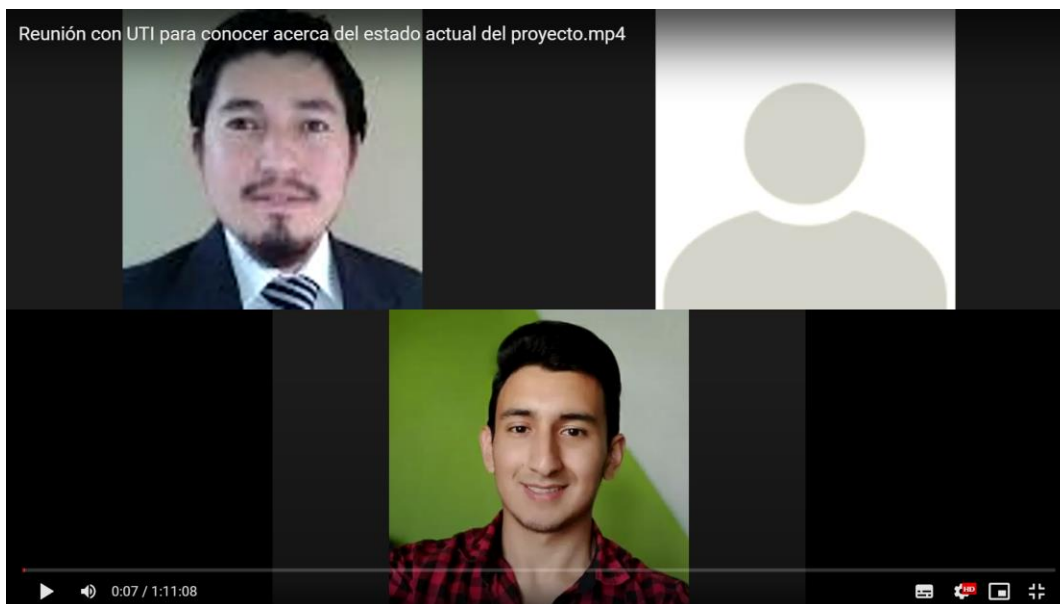


Firmado electrónicamente por:  
**JHON ALEXANDER  
CALDERON  
SANMARTIN**

Jhon Alexander Calderón Sanmartín  
**DIRECTOR DE TECNOLOGÍAS DE INFORMACIÓN**

## Anexo 19. Entrevista con miembros de la DTI para conocer el estado actual del proyecto

Entrevista virtual con el ing. Danny Muñoz y el ing. Máximo Álvarez para conocer el estado actual del proyecto; reunión grabada accesible mediante el siguiente enlace:  
<https://drive.google.com/entrevistaUTI>



*Figura 75. Captura de pantalla de la reunión de acercamiento con DTI.*

## **Anexo 20. Certificado de traducción de resumen**

Proyecto:

**Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario**

---

Certificado emitido por el Lcdo. Juan Diego Iñiguez Carrión que avala la revisión y traducción del  
resumen al idioma inglés



Loja, 23 de febrero de 2023

### **Certificación de traducción**

Yo Juan Diego Iñiguez Carrión, licenciado en Ciencias de la Educación mención inglés registro Nro.1031-2021-2295809 perteneciente a la academia Charlotte School corporación Jetmind de la ciudad de Cuenca certifico que:

El resumen de la tesis titulada: “Módulo para el seguimiento y control de becas de la Unidad de Bienestar Universitario” de la autoría de: Jackson Andrews Guzmán Tituana con cédula de ciudadanía número 1105740979, estudiante de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, es fiel traducción al idioma inglés elaborada por mi persona.

Lo certifico en honor a la verdad pudiendo el interesado hacer uso de este documento como estime conveniente.

Lcdo. Juan Diego Iñiguez Carrión  
Cdl: 1104652258

