



UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

CARRERA DE INGENIERÍA EN SISTEMAS

TÍTULO:

**“SISTEMA DE APOYO AL TRÁMITE DE
HOMOLOGACIÓN PARA EL A.E.I.R.N.N.R DE LA
U.N.L, UTILIZANDO EL FRAMEWORK SYMFONY”**

“TESIS DE GRADO PREVIA A LA OBTENCIÓN
DEL TÍTULO DE INGENIERA EN SISTEMAS”.

AUTORAS:

Janneth Alexandra Quizhpe Requelme

Silvana Jimena Vélez Suquilanda

DIRECTOR:

Ing. Edison Leonardo Coronel Romero

Loja – Ecuador

2013

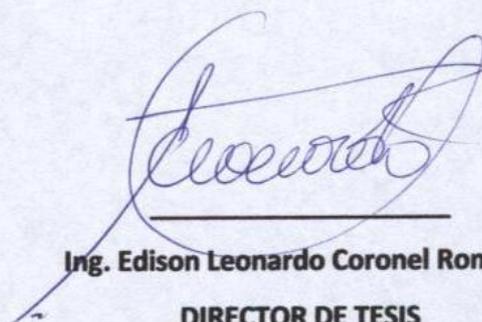
CERTIFICACIÓN DEL DIRECTOR

Ing. Edison Leonardo Coronel Romero
DIRECTOR DE TESIS

Certifico:

Que he revisado minuciosamente el trabajo de tesis denominado "SISTEMA DE APOYO AL TRÁMITE DE HOMOLOGACIÓN PARA EL A.E.I.R.N.N.R DE LA U.N.L UTILIZANDO EL FRAMEWORK SYMFONY", desarrollado por las egresadas Janneth Alexandra Quizhpe Requelme y Silvana Jimena Vélez Suquilanda, por lo cual autorizo su presentación y sustentación

Loja, febrero del 2013



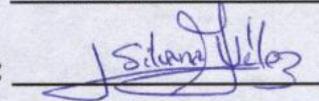
Ing. Edison Leonardo Coronel Romero
DIRECTOR DE TESIS

AUTORÍA

Nosotros, Silvana Jimena Vélez Suquilanda y Janneth Alexandra Quizhpe declaramos ser autoras del presente trabajo de tesis y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja, la publicación de nuestra tesis en el Repositorio Institucional-Biblioteca Virtual.

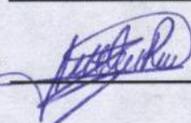
Autor: Silvana Jimena Vélez Suquilanda

Firma:  _____

Cédula: 1103735237

Fecha: 31 de mayo del 2013

Autor: Janneth Alexandra Quizhpe Requelme

Firma:  _____

Cédula: 4103780928

Fecha: 31 de mayo del 2013

AGRADECIMIENTO

Al finalizar el presente trabajo es inevitable expresar nuestra enorme gratitud y nuestros sinceros agradecimientos a todas las personas que de una u otra manera estuvieron apoyándonos incondicionalmente para poder llegar a la culminación del presente proyecto, a Dios, a nuestros padres y hermanos; y de manera especial y sincera a un gran amigo que con su apoyo y su capacidad supo guiar nuestras ideas, ha sido un aporte invaluable, no solamente en el desarrollo de la presente tesis, sino también en nuestra formación como investigadoras. No podemos dejar de lado nuestros agradecimientos a la Universidad Nacional de Loja, Carrera de Ingeniería en Sistemas, por permitir nuestra formación en sus aulas.

DEDICATORIA

*Este trabajo va dedicado a mis padres **Bolívar y Melania** por ser el pilar fundamental de mi vida, y porque gracias a ellos y a Dios soy todo lo que soy, tanto personal como profesionalmente, a mis hermanas **Yadira y Marjory** y **mi sobrina Melany**, porque junto con mis padres me han sabido brindar siempre su apoyo incondicional, consejos, comprensión y ayuda en cada momento, en cada instante de mi vida para cumplir todos mis objetivos propuestos.*

Y a todas las personas que de una u otra forma estuvieron apoyándome emocionalmente para poder culminar este trabajo. A todos Gracias por el apoyo brindado...

JANNETH A. QUIZHPE REQUELME.

Es un poco difícil realizar una dedicatoria, pues es mucha gente la que ha ayudado para que pueda llegar a este punto de mi trayectoria; sin embargo esta Tesis va dedicada especialmente a mi hija Jhulyana Estefanía, por quien cada día de mi vida tiene sentido, la testigo silenciosa de mis luchas cotidianas, a ella dedico, mi esperanza, mi alegría, mi vida y la culminación de este trabajo y lo que representa. Gracias Jhulyanita por regalarme cada minuto que me correspondía compartir contigo y poderlos dedicar a realizar el presente trabajo, gracias por tan enorme sacrificio.

A mi padre que me guía desde el cielo, a mi madre y mis hermanos el pilar fundamental que me sostiene, el apoyo incondicional y el consejo sabio y oportuno. A quienes adoro desde lo más profundo de mi corazón por ser artífices en la culminación de mis estudios

A Dios porque nos ha puesto en el camino ángeles llamados amigos. Que dedicaron su tan valioso tiempo en contribuir con su granito de arena. Gracias. Janneth y Benjamín, por su paciencia y su día día.

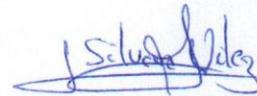
SILVANA J. VELEZ SUQUILANDA

CESIÓN DE DERECHOS

Janneth Alexandra Quizhpe Requelme y Silvana Jimena Vélez Suquilanda, autoras intelectuales del presente trabajo de tesis de investigación, autorizan a la Universidad Nacional de Loja, al Área de Energía, las Industrias y los Recursos Naturales no Renovables y por ende a la carrera de Ingeniería en Sistemas hacer uso del mismo en lo que estime conveniente.



Janneth Alexandra Quizhpe Requelme



Silvana Jimena Vélez Suquilanda

a. Título

**“SISTEMA DE APOYO AL TRÁMITE DE
HOMOLOGACIÓN PARA EL A.E.I.R.N.N.R DE
LA U.N.L, UTILIZANDO EL FRAMEWORK
SYMPHONY”**

b. Resumen

El presente trabajo comprende el desarrollo de una herramienta informática de apoyo al trámite de homologación para el A.E.I.R.N.N.R de la U.N.L, utilizando el framework *Symfony*. El proyecto inicia con una descripción de los diversos conceptos relacionados al desarrollo de aplicaciones web. Se describe como punto central el *framework Symfony 2*. La segunda parte comprende la ejecución del proyecto, en la cual se detallan el modelo de requerimientos, modelo conceptual, modelo de clases, modelo de casos de uso, esquemas de pantallas y diagramas de secuencia obtenidos en las diversas etapas de diseño de la aplicación, como resultado del uso de la metodología *ICONIX*. En base a este diseño y al esquema propuesto por *Symfony2* se realizó la codificación de la aplicación con los componentes necesarios para la administración de los documentos requeridos en el trámite de homologación, como es el componente de consulta online que permite a los estudiantes conocer el estado de la petición de homologación realizada.

Una vez terminado el proyecto podemos concluir que la estructura modular del Framework *Symfony2* permite rehusar funcionalidad, aunque tiene una clara desventaja por la diversidad de formatos que se usan para las configuraciones, lo cual eleva la curva de aprendizaje. Sin embargo el Framework ofrece gran facilidad en el uso de plantillas twig, que permiten representar una vista y separarla del código PHP; y en la configuración del motor de base de datos, ya que utiliza el ORM Doctrine el cual es totalmente independiente del motor de base de datos que se utilice. Y se lo se lo recomienda puesto que evita la escritura de sentencias DDL para la creación del modelo físico de datos, lo que ahorra tiempo de mantenimiento. Así también, en lo posible se debe utilizar un repositorio de código ya que facilita el trabajo desde cualquier lugar físico.

Dado que el enfoque de la presente investigación se centró en el estudio y aplicación práctica del framework *Symfony2*, el algoritmo de homologación desarrollado es susceptible de mejora, ante ello es posible el planteo de un nuevo tema de tesis que mejore el presente sistema a través de la construcción e integración de un algoritmo inteligente.

Summary

The present work comprises the development of a computer tool of support to the homologation step for to E.I.R.N.N.R of the U.N.L, using the framework Symfony. The project initializes with a description of the diverse concepts related to the development of web applications. There is described as central point the framework Symfony2. The second part comprises the execution of the project, in which there are detailed the model of requests, conceptual model, model of classes, model of cases of use, schemes of screens and diagrams of sequence obtained in the diverse stages of design of the application, like turned out from the use of the methodology ICONIX. Based on this design and the scheme proposed for Symfony2 the coding of the application was realized with the components necessary for the administration of the documents needed in the step of homologation, as it is the component of online consultation that there allows to the students to know the state of the request of realized homologation.

As soon as the project was finished we can conclude that the modular structure of Framework Symfony2 allows to refuse functionality, although it takes a clear disadvantage as the diversity of formats that are used for the configurations, which raises the learning curve. Nevertheless the Framework offers big facility in the use of templates twig, that they allow to represent a sight and to separate it from the code PHP; and in the configuration of the engine of database, since it uses the ORM There Indoctrinates which is completely independent from the engine of database that is used. And one recommends it since he avoids the writing of judgments DDL for the creation of the physical model of data, what saves maintenance time. This way also, in the possible thing it is necessary to use a code repository since it facilitates the work from any physical place.

Since the approach of the present investigation centred on the study and practical application of the framework Symfony2, the developed homologation algorithm is capable of progress, before it there is possible to raise of a new topic of thesis that improves the present system across the construction and integration of an intelligent algorithm.

Índice de contenidos

Certificación del director	ii
Autoría	iii
Agradecimiento	iv
Dedicatoria	v
Cesión de derechos	vi
a. Título.....	vii
b. Resumen	viii
Summary	ix
Índice de contenidos	x
Índice de figuras	xiii
Índice de tablas	xv
c. Introducción	1
d. Revisión de literatura.....	3
1. APLICACIONES WEB	3
1.1 Arquitectura de una aplicación web	3
1.2 Protocolo HTTP	3
1.3 Ventaja/desventaja HTTP	3
1.4 AJAX.....	4
1.5 Diferencia entre aplicaciones web clásicas y AJAX.....	4
1.6 HTML	7
1.7 ECMAScript	7
2. FRAMEWORK	8
2.1 Definición de framework	8
2.2 Symfony2.....	8
2.3 Seguridad en symfony2	9
3. ARQUITECTURA MVC	12
3.1 Descripción de la arquitectura MVC	12
3.2 MVC en symfony2.....	13
3.2.1 VISTA.....	13
3.2.2 MODELO	15
3.2.3. CONTROLADOR	17
4. MAPEO OBJETO RELACIONAL.....	22
4.1 Que es ORM	22
4.2 Casos de ORM	23
4.3 Ventajas y desventajas	24
4.4 Producto ORM para symfony2.0.....	25
4.5 Doctrine 2.....	25
4.5.1 Persistencia de los datos	26
4.5.2 Consulta de objetos	27
5. SUBVERSION	29
5.1 Repositorio SVN google.....	29
e. Materiales y Métodos	30

1. Métodos utilizados	30
1.1 Método deductivo	30
1.2 Método inductivo	31
1.3 Método experimental.....	31
2. Instrumentos para recolección de información	31
2.1 Entrevista	31
2.2 Observaciones	32
3. Metodología de desarrollo	32
3.1 Descripción de ICONIX	32
f. Resultados	34
Análisis de requerimientos.....	34
ESPECIFICACIONES DE LOS REQUISITOS DEL SOFTWARE	35
1. Introducción	35
1.1 Propósito	35
1.2 Alcance	35
1.3 Definiciones, siglas y abreviaturas	36
1.4 Referencias	36
1.5 Apreciación global.....	36
2. Descripción global	37
2.1 Perspectiva del producto.....	37
2.2 Funciones del producto.....	44
2.3. Características del usuario.....	45
2.4. Restricciones.....	45
2.5 Atención y dependencias	46
2.6 Prorratear los requisitos	46
3. Requisitos específicos	46
3.1 Requisitos de las interfaces externas	46
3.1.1 Interfaces del usuario:	46
3.1.2 Interfaces hardware	47
3.1.3 Interfaces del software.....	47
3.1.4 Interfaces de comunicación	47
3.2 Requisitos funcionales	48
3.3 Requisitos de desarrollo.....	53
3.4 Requisitos del banco de datos lógicos.....	53
3.5 Restricciones de diseño	53
3.6 Atributos del sistema de software	53
3.6.1 Fiabilidad.....	53
3.6.2 Disponibilidad.....	53
3.6.3 Seguridad.....	54
3.6.4 Portabilidad	54
3.7 Organizar los requisitos específicos.	54
3.7.1 Modo del sistema	54
3.7.2 Clases de usuario	54
3.7.3 objetos.....	54
3.7.4 Rasgo	54

3.7.5 Estímulo	55
3.7.6 Contestación	55
Resumen de requerimientos del sistema de homologación	56
Requerimientos funcionales	56
1. Administrador.....	56
2. Secretaria	56
3. Docente	57
4. Estudiante.....	58
Resumen de requerimientos no funcionales	59
GLOSARIO DE TÉRMINOS	60
DIAGRAMA DE CASOS DE USO	63
DIAGRAMA CONCEPTUAL	66
ANÁLISIS Y DISEÑO PRELIMINAR	67
OTROS DIAGRAMAS	141
g. Discusión.....	142
1. Desarrollo de la propuesta alternativa	142
2. Valoración técnica económica ambiental	147
h. Conclusiones	150
i. Recomendaciones	152
j. Bibliografía.....	153
K. Anexos	155

Índice de Figuras

Figura 1. Modelo Clásico de Aplicaciones Web[3]	6
Figura 2. Modelo Ajax de Aplicaciones Web [3].....	6
Figura 3. Arquitectura MVC (Modelo Vista Controlador)	12
Figura 4. Arquitectura MVC en Symfony2.....	13
Figura 5. Funcionamiento de un ORM	22
Figura 6. Muestra la persistencia de objetos con Doctrine.....	26
Figura 7. Prototipo de pantalla ingresar solicitud de homologación	38
Figura 8. Prototipo de pantalla consultar petición de homologación	389
Figura 9. Prototipo de pantalla solicitudes de homologación y asignar homologación.	39
Figura 10. Prototipo de pantalla manejar áreas y editar área.	40
Figura 11. Prototipo de pantalla manejar universidades y editar universidad.	40
Figura 12. Prototipo de pantalla manejar pensum y editar pensum	41
Figura 13. Prototipo de pantalla manejar homologación y editar homologación.....	42
Figura 14. Prototipo de pantalla resultado de homologación	43
Figura 15. Diagrama de caso de uso del peticionario	63
Figura 16. Diagrama de caso de uso del administrador de homologaciones.....	64
Figura 17. Diagrama de caso de uso del docente.....	64
Figura 18. Diagrama de caso de uso del administrador.....	65
Figura 19. Pantalla ingresar petición de homologación.....	67
Figura 20. Diagrama de secuencia ingresar petición de homologación.....	69
Figura 21. Pantalla edición de petición de homologación.	70
Figura 22. Diagrama de secuencia editar petición de homologación.....	72
Figura 23. Pantalla consultar petición de homologación para el secretario o docente	73
Figura 24. Pantalla consultar petición de homologación para el estudiante	73
Figura 25. Diagrama de secuencia consultar petición de homologación	75
Figura 26. Pantalla peticiones de homologación para el secretario	76
Figura 27. Pantalla editar petición de homologación.	77
Figura 28. Diagrama de secuencia administrar peticiones de homologación	80
Figura 29. Pantalla peticiones de homologación para docente.....	81
Figura 30. Pantalla homologar petición de homologación.....	82
Figura 31. Diagrama de secuencia homologaciones	85
Figura 32. Pantalla resultado previo de la petición homologación	86
Figura 33. Pantalla resultado de la petición de homologación	87
Figura 34. Diagrama de secuencia resultado de la petición de homologación	90
Figura 35. Pantalla administrar usuarios	92
Figura 36. Pantalla editar usuario	92
Figura 37. Diagrama de secuencia administrar usuarios	95
Figura 38. Pantalla eliminar usuario	96
Figura 39. Diagrama de secuencia eliminar usuario	98
Figura 40. Pantalla administrar universidades	99
Figura 41. Pantalla editar área	99
Figura 42. Diagrama de secuencia administrar universidades.....	102
Figura 43. Pantalla eliminar universidades.....	103

Figura 44. Diagrama de secuencia eliminar universidad.....	105
Figura 45. Pantalla administrar áreas	106
Figura 46. Pantalla editar área	106
Figura 47. Diagrama de secuencia administrar áreas.....	109
Figura 48. Pantalla eliminar área	110
Figura 49. Diagrama de secuencia eliminar área.....	112
Figura 50. Pantalla Administrar Mallas Curriculares	113
Figura 51. Diagrama de secuencia administrar mallas curriculares.....	116
Figura 52. Pantalla editar malla curricular	117
Figura 53. Diagrama de secuencia editar malla curricular	120
Figura 54. Pantalla eliminar malla curricular	121
Figura 55. Diagrama de secuencia eliminar malla curricular.....	123
Figura 56. Pantalla estructura pensum	124
Figura 57. Diagrama de secuencia estructura pensum.....	126
Figura 58. Pantalla editar unidad o materia	127
Figura 59. Diagrama de secuencia editar unidad o materia.....	130
Figura 60. Pantalla eliminar unidad.....	131
Figura 61. Diagrama de secuencia eliminar unidad	133
Figura 62. Pantalla administrar carreras	134
Figura 63. Pantalla editar carrera.....	134
Figura 64. Diagrama de secuencia administrar carreras.....	137
Figura 65. Pantalla eliminar carrera	138
Figura 66. Diagrama de secuencia eliminar carrera.....	140
Figura 67. Esquema de la interfaz de usuario.....	141

Índice de Tablas

Tabla I. Security.yml	10
Tabla II. Plantillas twig.....	14
Tabla III. Entity	15
Tabla IV. Controller	18
Tabla V. Mapeo	26
Tabla VI. Dql (createquery)	28
Tabla VII. Análisis de requerimientos - Definición de siglas y abreviaturas	36
Tabla VIII. Análisis de requerimientos - Referencias.....	36
Tabla IX. Análisis de requerimientos - Características del usuario	45
Tabla X. Análisis de requerimientos - Requisitos funcionales-administrador.....	48
Tabla XI. Análisis de requerimientos - Requisitos funcionales- secretaria(o)	49
Tabla XII. Análisis de requerimientos - Requisitos funcionales- docente.....	50
Tabla XIII. Análisis de requerimientos - Requerimientos funcionales- estudiante.....	52
Tabla XIV. Análisis de requerimientos - Resumen de requerimientos funcionales- Administrador	56
Tabla XV. Análisis de requerimientos - Resumen de requerimientos funcionales- Secretaria	56
Tabla XVI. Análisis de requerimientos - Resumen de requerimientos funcionales- Docente	57
Tabla XVII. Análisis de requerimientos - Resumen de requerimientos funcionales- Estudiante	58
Tabla XVIII. Análisis de requerimientos-Resumen de requerimientos no funcionales	59
Tabla XIX. Glosario de términos.....	60
Tabla XX. Caso de uso: Ingresar petición de homologación.....	68
Tabla XXI. Fragmento: Editar petición de homologación	70
Tabla XXII. Fragmento: Consultar petición de homologación	74
Tabla XXIII. Caso de uso: Administrar peticiones de homologación.....	77
Tabla XXIV. Caso de uso: Administrar homologaciones.....	83
Tabla XXV. Fragmento: Resultado de homologación	88
Tabla XXVI. Fragmento: Ejecutar homologación	91
Tabla XXVII. Caso de uso: Administrar usuarios	93
Tabla XXVIII. Fragmento: Eliminar usuario	96
Tabla XXIX. Caso de uso: Administrar universidades.....	100
Tabla XXX. Fragmento: Eliminar universidad.....	103
Tabla XXXI. Caso de uso: Administrar áreas	107
Tabla XXXII. Fragmento: Eliminar área.....	110
Tabla XXXIII. Caso de uso: Administrar mallas curriculares	114
Tabla XXXIV. Fragmento: Editar malla curricular.....	117
Tabla XXXV. Fragmento: Eliminar malla curricular	121
Tabla XXXVI. Fragmento: Estructura pensum	125
Tabla XXXVII. Fragmento: Editar Unidad o Materia.....	127
Tabla XXXVIII. Fragmento: Eliminar unidad.....	131

Tabla XXXIX. Caso de uso: Administrar carreras	135
Tabla XL. Fragmento: Eliminar carrera	138
Tabla XLI. Discusión - Valoración técnica - Recursos humanos.....	147
Tabla XLII. Discusión - Valoración técnica - Recursos económicos	147
Tabla XLIII. Discusión - Valoración técnica - Recursos materiales	148
Tabla XLIV. Discusión - Valoración técnica - Recursos técnicos y tecnológicos	148
Tabla XLV. Discusión - Valoración técnica - Resumen del presupuesto	149

c. Introducción

En la actualidad, la necesidad de aplicaciones web en las diferentes organizaciones ha llegado a ser un denominador común para el funcionamiento y la existencia de las mismas. En este contexto múltiples tecnologías han evolucionado con la promesa de facilitar la construcción de este tipo de sistemas de información, dada su facilidad de administración y la posibilidad de brindar servicios a los usuarios de una organización a través de la red.

En este contexto, surge la necesidad real de construcción de una aplicación que permita realizar el proceso de homologación de carreras el AEIRNNR de la Universidad Nacional de Loja, la cual resultó ser el punto de partida del presente trabajo.

Entre los múltiples entornos de desarrollo uno de los más populares es el basado en el lenguaje PHP, el cual permite construir fácilmente formularios a través del uso del script embebido en archivos HTML. El principal inconveniente de esta estrategia, resulta ser la mezcla de elementos de presentación y de control. Ante esta perspectiva, Symfony2 emerge como un framework cuyo objetivo es separar y aislar los diferentes aspectos relacionados a una aplicación.

Puesto que el desarrollo de una aplicación WEB en frameworks actuales, implica un conjunto de conceptos diferentes como llamadas asíncronas, mapeo objeto relacional, plantillas, controladores, arquitecturas y otros más, dichos temas son analizados con detalle, ello permite ubicar adecuadamente el marco teórico de lo aplicado en el presente trabajo

El desarrollo en grupo involucra también un reto que es el de mantener un repositorio de código conjunto a través del cual compartir el código pero más importante aún, es el realizar la integración continua del mismo. Para ello se configuró un proyecto Subversion en el repositorio público de Google Code.

Coincidió durante el desarrollo del trabajo, que el framework Symfony se encontraba en una etapa de transición de la versión 1.4 a la 2.0. Ambas versiones fueron

evaluadas definir aquella con la cual trabajar. Y, pese a la carencia inicial de documentación y ejemplos de uso, se decidió utilizar la nueva versión debido a sus múltiples ventajas de organización del proyecto e integración con otros frameworks como Doctrine ORM.

Posteriormente, se muestran todos los modelos generados a partir de la aplicación de las diversas etapas de la metodología ICONIX, se detallan los modelos de requerimientos, casos de uso, conceptual y de clases, esquemas de transición de pantallas, análisis de robustez y diagramas de secuencia.

Se muestran también ejemplos reales de como se ha llegado desde el diseño a la estructuración de código para uno de los casos de uso y resultado de ello se obtiene la aplicación incluida en el CD adjunto o disponible para descarga en <https://code.google.com/p/homologacion/>.

Finalmente se discuten los resultados del proyecto y se presenta un grupo de conclusiones y recomendaciones resultado de la experiencia lograda en el desarrollo de la aplicación.

d. Revisión de literatura

1. APLICACIONES WEB

1.1 Arquitectura de una aplicación web

En una aplicación web se distinguen tres niveles: interfaz de usuario, lógica de negocio y datos. La arquitectura de una aplicación web define como se organizan estos niveles

- El nivel de interfaz, compuesto por las páginas HTML¹ que el usuario solicita a un servidor web y que visualiza en un cliente web(normalmente un navegador web)
- El nivel de lógica, compuesto por los módulos que implementan la lógica de la aplicación y que se ejecutan en un servidor de aplicaciones.
- El nivel de datos está compuesto por los datos, normalmente gestionados por un sistema de gestión de bases de datos (servidor de datos), que maneja la aplicación web. [1]

1.2 Protocolo HTTP

El protocolo de transferencia de hipertexto (HyperText Transfer Protocol) es un lenguaje de texto que permite a dos máquinas comunicarse entre sí.

1.3 Ventaja/desventaja HTTP

El protocolo HTTP está basado en mensajes. Texto plano.

Ventajas:

- Legible.
- Fácil de depurar.

Desventajas:

- El mensaje es más largo.

¹ **HTML**, acrónimo del inglés HyperText Markup Language (Lenguaje de Marcado de Hipertexto)

1.4 AJAX

Siglas de Asynchronous JavaScript and XML, es un a término que describe la transferencia de información utilizando un conjunto de tecnologías existentes juntas, como son: HTML o XHTML², hojas de estilo (Cascading Style Sheets o css), Javascript, el DOM (Document Object Model), XML³, XSLT⁴, y el objeto XMLHttpRequest.

“Cuando se combinan estas tecnologías en el modelo Ajax, las aplicaciones funcionan mucho más rápido, ya que las interfaces de usuario se pueden actualizar por partes sin tener que actualizar toda la página completa”. [2]

1.5 Diferencia entre aplicaciones web clásicas y AJAX

La diferencia de funcionamiento entre las aplicaciones web “clásicas” en las que la las acciones del usuario en la interfaz disparan un requerimiento HTTP al servidor web y el servidor, después de efectuar un proceso (recopila información, procesa números, hablando con varios sistemas propietarios), le devuelve una página HTML al cliente; y las aplicaciones basadas en AJAX.

Las aplicaciones Web tradicionales basan su esquema de interacción en el funcionamiento del protocolo HTTP; el protocolo HTTP está basado en el paradigma petición-respuesta, donde participan dos actores, el cliente y el servidor.

El cliente envía peticiones de recursos al servidor, descartando la página desplegada actualmente. El servidor recibe la petición y la procesa, respondiendo seguidamente con una página HTML nueva, la cual es recibida y desplegada por el navegador. Este

² **XHTML**, acrónimo del inglés eXtensible Hyper Text Markup Language (lenguaje extensible de marcado de hipertexto)

³ **XML**, acrónimo del inglés eXtensible Markup Language (lenguaje de marcado ampliable o extensible)

⁴ **XSLT**, acrónimo de inglés Extensible Stylesheet Language Transformation

proceso genera una interrupción en la interacción entre el usuario y la aplicación en cada petición.

A diferencia de las aplicaciones Web clásicas, las aplicaciones AJAX pueden enviar peticiones al servidor sin interrumpir la interacción, manteniendo la página actual en el navegador de tal manera que le permita al usuario seguir interactuando con la aplicación. Esto es posible gracias al uso de peticiones en segundo plano (peticiones asíncronas).

En el esquema clásico de las aplicaciones Web, las páginas de respuesta enviadas por el servidor probablemente contienen pocos cambios respecto a la página anterior, lo que puede producir una sobrecarga innecesaria entre cada petición. Con el uso de peticiones asíncronas, las aplicaciones AJAX pueden solicitar al servidor Web únicamente la información que represente un cambio en la página, sin solicitar datos que ya se encuentren en el cliente (como imágenes, encabezados, menús, etc.). Esta característica permite disminuir considerablemente el impacto sobre la interacción del usuario con la aplicación que comúnmente se presenta en las aplicaciones Web tradicionales al momento de enviar peticiones al servidor.

Cuando del servidor se obtiene como respuesta datos que puedan representar un cambio en la aplicación, el cliente puede modificar la vista dinámicamente con la información que recibe, sin que necesite solicitar otra página. Esta característica, hace que AJAX permita un uso eficiente del ancho de banda, ya que se transmite del servidor al cliente solo la información que necesita y no las páginas completas con datos que ya están en cliente como por ejemplo: encabezados, pie de página, imágenes y gráficos, etc.

Las aplicaciones AJAX incorporan un componente adicional del lado del cliente denominado motor AJAX. El motor AJAX es un componente constituido principalmente por código JavaScript el cual se encarga de todo el procesamiento del lado del cliente y sirve como intermediario entre la interfaz de la aplicación y el servidor. (Ver figura 2)

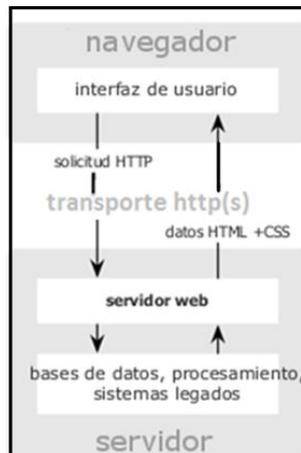


Figura 1. Modelo Clásico de Aplicaciones Web [3]

En el modelo clásico de Aplicaciones, las peticiones al servidor se originan directamente desde la interfaz como consecuencia de las acciones del usuario. (Ver Figura 1.)

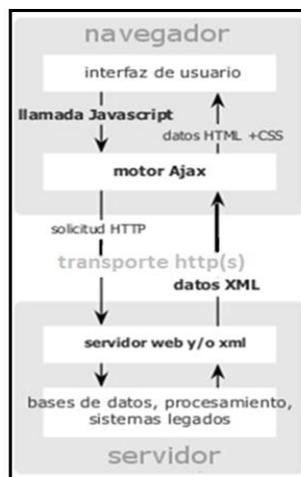


Figura 2. Modelo Ajax de Aplicaciones Web [3]

En cambio, en el modelo AJAX las acciones del usuario sobre la interfaz son interceptadas por el motor AJAX el cual tiene la posibilidad de darles respuesta directamente o de generar peticiones al servidor (normalmente) en segundo plano. (Ver Figura 2.)

En el modelo clásico de aplicaciones Web el servidor responde directamente con datos en formato HTML y CSS⁵ los cuales son desplegados directamente en el cliente. En el modelo AJAX el servidor responde comúnmente con datos en formato XML o texto plano, los cuales son obtenidos por el motor AJAX y son utilizados por éste para realizar los cambios dinámicos sobre la interfaz.

1.6 HTML

HTML, siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

1.7 ECMAScript

Javascript, se refiere al estándar ECMAScript, el cual es una especificación de lenguaje de programación publicada por Ecma International. Actualmente está aceptado como el estándar ISO⁶ 16262.

De este modo, un motor ECMAScript es un programa que ejecuta código escrito en sus versiones estándares como, por ejemplo, Javascript. Así pues, hay que recalcar que Javascript es una implementación de ECMAScript. Utilizada principalmente en el lado cliente, como parte de un entorno/navegador web. Lo que permite mejoras en la interfaz de usuario y páginas web dinámicas.

⁵ **CSS**, acrónimo del inglés Cascading Style Sheets (es decir, hojas de estilo en cascada)

⁶ **ISO**, acrónimo del inglés International Standards Organization (Organización Internacional de Estandarización)

2. FRAMEWORK

2.1 Definición de framework

Los Frameworks son desarrollados con el objetivo de brindar a los programadores y diseñadores una mejor organización y estructura a los proyectos. Se utiliza la Programación Orientada a Objetos (POO), permitiendo la reutilización de código.

Lo cual ayuda en el desarrollo de software, ya que proporciona una estructura definida la cual permite crear aplicaciones con mayor rapidez, también ayuda a la hora de realizar el mantenimiento del sitio gracias a la organización durante el desarrollo de la aplicación.

2.2 Symfony2

Symfony es un framework que permite optimizar el desarrollo de las aplicaciones web, se basa en la arquitectura MVC (Modelo, Vista, Controlador). Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web.

Desarrollado en PHP 5.3, es compatible con Windows, Unix o Linux. Compatible con las diferentes bases de datos: MySQL, PostgreSQL, Oracle o Microsoft SQL Server. También admite e integra herramientas para usar Propel o Doctrine como ORMs⁷ aunque cada vez más enfocado a Doctrine 2.0.

Entre sus principales características se tiene:

- Se instala y se configura en la mayoría de plataformas.
- Es independiente del SGBD⁸. Lo que aporta una enorme flexibilidad al desarrollador, ya que puede elegir en cada proyecto cual es la mejor opción, pudiendo elegir algunos gestores como MySQL, PostgreSQL, Oracle o Microsoft SQL Server.

⁷ **ORM**, acrónimo del inglés Object Relational Mapping y pertenece a la categoría Bases de Datos

⁸ **SGBD**, sistemas de gestión de base de datos, (en inglés: DataBase Management System, abreviado DBMS)

- Sigue el MVC
- Basado en la premisa convenir y no configurar. (CoC - Convención sobre Configuración). Se acuerda una configuración pre-establecida que permite al programador ponerse a trabajar sin tener que preocuparse por ello.
- Usa PHP con OOP (Programación Orientada a Objetos) y requiere PHP 5 aunque es altamente recomendable que sea la versión 5.3 o superior.
- Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- Fácil de extender permitiendo la integración con librerías de terceros.

2.3 Seguridad en symfony2

- La seguridad se configura usando uno o varios ficheros en Yaml⁹ y permite asignar roles a clases ver Tabla I. Así se simplifica mucho el trabajo. Todos los usuarios pueden registrarse como “estudiante” pueden iniciar sesión como “usuarios registrados”. Sin embargo solo el administrador, docente, y secretaria podrán acceder como tales, están puestos en el fichero de seguridad que es inaccesible y no existe en la BBDD.
- Symfony además cuenta con sistemas para evitar ataques XSS¹⁰ (cross-site scripting), y también contra CSRF¹¹ (cross site request forgery) mantiene campos ocultos en los formularios.

En la siguiente tabla se encuentra un ejemplo del código de configuración de la clase security.yml. Usado en la aplicación.

⁹ **YAML**, es un lenguaje de marcado "ligero" que permite especificar estructuras (tales como arrays) con menos caracteres que XML.

¹⁰ **XSS**, acrónimo del inglés XSS “Cross Site Scripting” que se podría traducir al castellano como “inyección de código entre websites”.

¹¹ **CSRF**, acrónimo del inglés Cross-site request forgery o falsificación de petición en sitios cruzados.

TABLA I SECURITY.YML

homologa/ app/ config/ security.yml	
1	security:
2	providers:
3	fos_userbundle:
4	id: fos_user.user_manager
5	
6	usuarios:
7	entity: { class: Uni\HomologacionBundle\Entity\Usuario, property: email }
8	
9	encoders:
10	FOS\UserBundle\Model\UserInterface: sha512
11	Uni\HomologacionBundle\Entity\Usuario: { algorithm: sha512, iterations: 10 }
12	
13	firewalls:
14	main:
15	pattern: ^/
16	form_login:
17	provider: fos_userbundle
18	#acceder por defecto a esta pagina /universidad
19	
20	default_target_path: /inicio
21	
22	#csrf_provider: form.csrf_provider
23	logout:
24	path: /logout
25	target: /login
26	
27	# default_target_path: /universidad
28	anonymous: true
29	#security: true
30	
31	frontend:
32	pattern: ^/*
33	anonymous: true
34	form_login:
35	provider: usuarios
36	login_path: /login
37	use_forward: false
38	check_path: /login_check
39	failure_path: null
40	logout: true
41	
42	
43	access_control:
44	- { path: ^/login\$, role: IS_AUTHENTICATED_ANONYMOUSLY }
45	- { path: ^/register, role: IS_AUTHENTICATED_ANONYMOUSLY }
46	- { path: ^/resetting, role: IS_AUTHENTICATED_ANONYMOUSLY }
47	- { path: ^/persona/new, role: IS_AUTHENTICATED_ANONYMOUSLY }
48	- { path: ^/persona/edit, role: [ROLE_ESTUDIANTE, ROLE_SECRETARIA, ROLE_DOCENTE,
49	- { path: ^/persona/create, role: [IS_AUTHENTICATED_ANONYMOUSLY, ROLE_SECRETARIA,
50	- { path: ^/persona/update, role: [ROLE_SECRETARIA, ROLE_ESTUDIANTE, ROLE_DOCENTE,
51	
52	- { path: ^/universidad, role: ROLE_ADMINISTRADOR }
53	- { path: ^/area/, role: ROLE_ADMINISTRADOR }
54	
55	- { path: ^/peticionhomologacion/listPorMallaCurricular, role: ROLE_DOCENTE }
56	
57	- { path: ^/carrera/listByPersona, role: ROLE_SECRETARIA }
58	- { path: ^/carrera/listByMallaCurricular, role: [ROLE_SECRETARIA, ROLE_DOCENTE] }

```
59 - { path: ^/carrera/listByUniversidad, role: [ROLE_ESTUDIANTE, ROLE_SECRETARIA]}
60 - { path: ^/carrera/, role: [ROLE_ADMINISTRADOR ]}
61 - { path: ^/persona/, role: ROLE_ADMINISTRADOR }
62
63
64 - { path: ^/peticionhomologacion/new, role: [ROLE_ESTUDIANTE, ROLE_USER, ROLE_SECRETARIA] }
65 - { path: ^/peticionhomologacion/, role: [ROLE_DOCENTE, ROLE_SECRETARIA, ROLE_ESTUDIANTE,
66
67
68 - { path: ^/mallacurricular, role: ROLE_ADMINISTRADOR }
69
70
71
72 role_hierarchy:
73     ROLE_ADMIN:     ROLE_USER
74 #     ROLE_SUPER_ADMIN: ROLE_ADMIN
75     ROLE_DOCENTE: [ROLE_ESTUDIANTE, ROLE_DOCENTE]
76
```

3. ARQUITECTURA MVC

3.1 Descripción de la arquitectura MVC

El término MVC proviene de tres palabras que se utilizan dentro del ambiente de desarrollo de software: Model – View – Controller, en castellano Modelado, Vista y Controlador. Esta arquitectura permite dividir las aplicaciones en tres grandes capas:

1. Vista (V): Todo lo que se refiera a la visualización de la información, el diseño, colores, estilos y la estructura visual de las páginas.
2. Modelado (M): Es el responsable de la conexión a la base de datos y la manipulación de los datos mismos. Esta capa está pensada para trabajar con los datos así como también obtenerlos, pero no mostrarlos, ya que la capa de presentación de datos es la vista.
3. Controlador (C): Su responsabilidad es procesar y mostrar los datos obtenidos por el Modelado. Es decir, este último trabaja de intermediario entre los otros dos, encargándose también de la lógica de negocio.

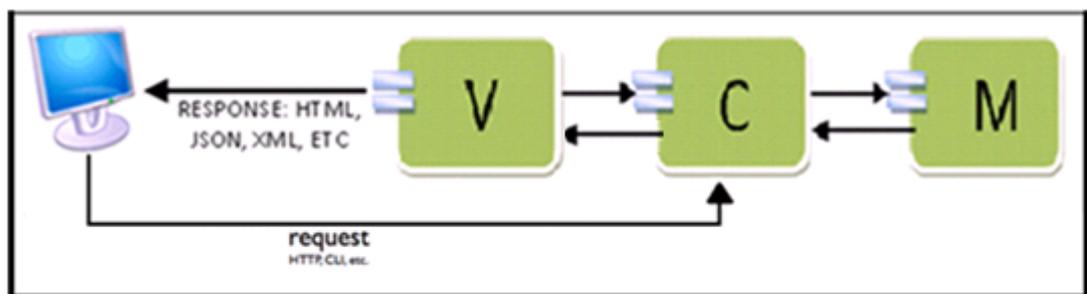


Figura 3. Arquitectura MVC (Modelo Vista Controlador)

El cliente envía una señal llamada REQUEST o Petición, (Ver Figura 3) ésta es interceptada por el Controlador quien realiza las validaciones necesarias, procesamiento de dichos datos y lógica de negocio asociadas a esa petición del cliente. El Controlador envía datos al Modelado, por ejemplo para ser guardados en una base de datos y/o los obtiene dependiendo de la solicitud del usuario para finalmente enviarlos a la Vista a fin de ser mostrador nuevamente al cliente a través de un RESPONSE o respuesta.

3.2 MVC en symfony2

3.2.1 VISTA

La capa de interfaz es la encargada de “comunicarse” con el usuario (Ver Figura 4). Recibe información desde el sistema y desde el usuario. En esta aplicación se usaran las herramientas que facilita symfony. El framework Symfony utiliza twig para representar la “Vista”, que es la capa de presentación.

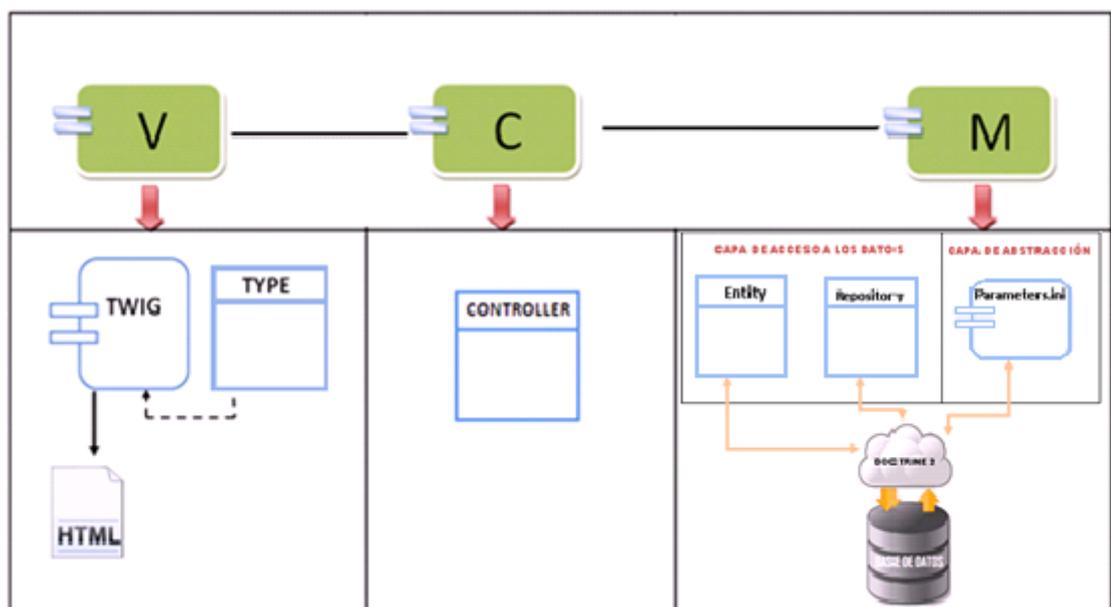


Figura 4. Arquitectura MVC en Symfony2

Twig permite separar las plantillas de forma jerárquica para evitar repetir el código, es decir separar esto en dos capas. La primera es el programa con la lógica de negocio para resolver el problema específico de esa página, mientras que la otra es una página que no contenga el mencionado código sino solo lo necesario para mostrar los datos a los usuarios.

Twig es un sistema de plantillas para la programación PHP¹² que viene incluido en Symfony2, es open source.

¹² **PHP**, acrónimo del inglés Hypertext Pre-processor. Lenguaje de programación usado generalmente en la creación de contenidos para sitios web.

- Symfony2 Traduce las plantillas twig a código PHP optimizado y reduciendo en mucho la cantidad de código que generaría usando plantillas PHP.
- Twig tiene un sistema de evaluación de código que permite evitar que los usuarios puedan modificar el diseño de las plantillas.
- Twig permite definir etiquetas, funciones y servicios.

La siguiente tabla muestra la plantilla edit.html.twig, utilizada en el proyecto para generar la pantalla Editar Área.

**TABLA II
PLANTILLAS TWIG**

homologa/ src/ Unl/ HomologacionBundle/ Resources/ views/ Area/ edit.html.twig	
1	{% block title %}Edición de Areas{% endblock %}
2	{% block greensectiontitle %}EDITAR AREA{% endblock %}
3	{% block body %}
4	<h2>Registre la información del Área</h2>
5	<form id="editForm" action="{{ entity.id ? path('area_update', { 'id': entity.id }) :
6	{{ form_errors(edit_form) }}
7	{{ form_row(edit_form.nombre) }}
8	{{ form_row(edit_form.description) }}
9	{% macro prototype(carrera) %}
10	<tr>
11	<td class="center">{{ form_widget(carrera.codigo) }}</td>
12	<td>{{ form_widget(carrera.nombre) }}</td>
13	<td><div id="controls" style="text-align: center;"><a title="Quitar carrera del área"
14	</tr>
15	{% endmacro %}
16	
17	<div id="records_table">
18	<table id="carreras-table" padding="0" margin="0">
19	<thead>
20	<tr>
21	<th colspan="3">
22	<div id="controls">
23	Carreras
24	<a title="Agregar una carrera al área" href="#" id="add-carrera"
25	</div>
26	</th>
27	
28	</tr>
29	<tr>
30	<th>Codigo</th>
31	<th>Nombre</th>
32	<th>Acciones</th>
33	</tr>
34	</thead>
35	<tbody id="carreras-table-body">
36	codigo-data-prototype="{{ form_widget(edit_form.carreras.get('prototype').codigo)
37	nombre-data-prototype="{{ form_widget(edit_form.carreras.get('prototype').nombre)
38	[# iterate over each existing carrera and render its field: codigo, nombre, universidad #]
39	{% for carrera in edit_form.carreras %}

```

40         {{_self.prototype(carrera)}}
41         {% endfor %}
42     </tbody>
43 </table>
44 </div>
45
46 <div id="controls">
47     <a title="Guardar datos y carreras del área" id="saveButton" href="#">Guardar</a>
48     <a title="Cancelar y volver a la lista de áreas" href="{{ path('area') }}">Cancelar</a>
49 </div>
50 </form>
51 {% endblock %}
52

```

3.2.2 MODELO

El Modelo se subdivide en la capa de acceso a los datos y en la capa de abstracción de la base de datos (Ver Figura 4.). Así, las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar consultas. Si se cambia de base de datos solamente es necesaria la capa de abstracción. En la capa de modelo Symfony trabaja con un ORM en este caso Doctrine2.

La siguiente tabla muestra la clase Entity/Área.php, utilizada en el proyecto para generar la tabla Área en la Base de Datos

**TABLA III
ENTITY**

```

homologa/ src/ Unl/ HomologacionBundle/ Entity/ Area.php
1 <?php
2
3 namespace Unl\HomologacionBundle\Entity;
4
5 use Doctrine\ORM\Mapping as ORM;
6 use Symfony\Component\Validator\Constraints as Assert;
7 use Doctrine\Common\Collections\ArrayCollection;
8
9
10 /**
11  * @ORM\Entity(repositoryClass="Unl\HomologacionBundle\Repository\AreaRepository")
12  */
13 class Area
14 {
15     /**
16      * @ORM\Id
17      * @ORM\Column(type="bigint")
18      * @ORM\GeneratedValue(strategy="AUTO")
19      */
20     protected $id;
21
22     /**
23      * @ORM\Column(type="string", length=100)

```

```

24     * @Assert\NotBlank()
25     */
26     protected $nombre;
27
28     /**
29     * @ORM\Column(type="text")
30     */
31     protected $description;
32
33     /**
34     * @ORM\OneToMany(targetEntity="Carrera", cascade={"all"}, mappedBy="area")
35     */
36     protected $carreras;
37
38     /**
39     * Get id
40     *
41     * @return bigint
42     */
43     public function getId()
44     {
45         return $this->id;
46     }
47
48     /**
49     * Set nombre
50     *
51     * @param string $nombre
52     */
53     public function setNombre($nombre)
54     {
55         $this->nombre = $nombre;
56     }
57
58     /**
59     * Get nombre
60     *
61     * @return string
62     */
63     public function getNombre()
64     {
65         return $this->nombre;
66     }
67
68     /**
69     * Set description
70     *
71     * @param text $description
72     */
73     public function setDescription($description)
74     {
75         $this->description = $description;
76     }
77
78     /**
79     * Get description
80     *
81     * @return text
82     */
83     public function getDescription()
84     {
85         return $this->description;
86     }

```

```

89     }
90     public function __toString()
91     {
92         return $this->id.' '.$this->nombre;
93     }
94
95     public function __construct()
96     {
97         $this->carreras = new \Doctrine\Common\Collections\ArrayCollection();
98     }
99
100    /**
101     * Add carreras
102     *
103     * @param Uni\HomologacionBundle\Entity\Carrera $carreras
104     */
105    public function addCarrera(\Uni\HomologacionBundle\Entity\Carrera $carrera)
106    {
107        $this->carreras->add($carrera);
108        $carrera->setArea($this);
109    }
110
111
112    public function setCarreras($carreras)
113    {
114        foreach ($carreras as $carrera) {
115            $carrera->setArea($this);
116        }
117        $this->carreras = $carreras;
118    }
119
120    /**
121     * Get carreras
122     *
123     * @return Doctrine\Common\Collections\Collection
124     */
125    public function getCarreras()
126    {
127        return $this->carreras;
128    }
129 }
130 }

```

3.2.3. CONTROLADOR

La capa de negocio o aplicación actúa de mediados entre la capa de presentación y la capa de persistencia (Ver Figura 4.), contiene la implementación de la funcionalidad asociada a los casos de uso. Aunque no almacena datos, todos los datos antes de ir a la capa de almacenamiento pasan por aquí para ser validados, o tratados si es necesario. Los datos cuando vuelven de la capa de persistencia también pasan por aquí antes de llegar a las vistas. En symfony estos serían los “controllers” un tipo de clases necesario.

La siguiente tabla muestra la clase AreaController.php, utilizada en el proyecto.

**TABLA IV
CONTROLLER**

homologa/ src/ Unl/ HomologacionBundle/ Controller/ AreaController.php	
1	<?php
2	
3	namespace Unl\HomologacionBundle\Controller;
4	
5	use Symfony\Bundle\FrameworkBundle\Controller\Controller;
6	use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;
7	use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
8	use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;
9	use Unl\HomologacionBundle\Entity\Area;
10	use Unl\HomologacionBundle\Entity\Carrera;
11	use Unl\HomologacionBundle\Form\AreaType;
12	use Unl\HomologacionBundle\Form\CarreraType;
13	
14	/**
15	* Area controller.
16	*
17	* @Route("/area")
18	*/
19	class AreaController extends Controller {
20	
21	/**
22	* Lists all Area entities.
23	*
24	* @Route("/", name="area")
25	* @Template()
26	*/
27	public function indexAction() {
28	
29	\$sample = new Area();
30	\$sample->setNombre('');
31	
32	\$searchForm = \$this->createForm(new AreaType(), \$sample);
33	\$request = \$this->getRequest();
34	\$searchForm->bindRequest(\$request);
35	
36	\$em = \$this->getDoctrine()->getEntityManager();
37	\$entities = \$em->getRepository('UnlHomologacionBundle:Area')->findByNombre(\$sample-
38	
39	return array('entities' => \$entities,
40	'search_form' => \$searchForm->createView());
41	}
42	
43	/**
44	* Finds and displays a Area entity.
45	*
46	* @Route("/{id}/show", name="area_show")
47	* @Template()
48	*/
49	public function showAction(\$id) {
50	\$em = \$this->getDoctrine()->getEntityManager();
51	
52	\$entity = \$em->getRepository('UnlHomologacionBundle:Area')->find(\$id);
53	
54	if (!\$entity) {
55	throw \$this->createNotFoundException('No se encuentra el Area.');
56	}
57	
58	\$deleteForm = \$this->createDeleteForm(\$id);
59	

```

60         return array(
61             'entity' => $entity,
62             'delete_form' => $deleteForm->createView(),);
63     }
64
65     /**
66      * Displays a form to create a new Area entity.
67      *
68      * @Route("/new", name="area_new")
69      * @Template()
70      */
71     public function newAction() {
72         $entity = new Area();
73         $form = $this->createForm(new AreaType(), $entity);
74
75         return array(
76             'entity' => $entity,
77             'edit_form' => $form->createView()
78         );
79     }
80
81     /**
82      * Creates a new Area entity.
83      *
84      * @Route("/create", name="area_create")
85      * @Method("post")
86      * @Template("UnlHomologacionBundle:Area:new.html.twig")
87      */
88     public function createAction() {
89         $entity = new Area();
90         $request = $this->getRequest();
91         $form = $this->createForm(new AreaType(), $entity);
92         $form->bindRequest($request);
93
94         if ($form->isValid()) {
95             $em = $this->getDoctrine()->getEntityManager();
96             $em->persist($entity);
97             $em->flush();
98
99             return $this->redirect($this->generateUrl('area', array('id' => $entity->getId())));
100         }
101
102         return array(
103             'entity' => $entity,
104             'edit_form' => $form->createView()
105         );
106     }
107
108     /**
109      * Agrega carrera al Area actual
110      *
111      * @Route("/carreraAdd", name="carrera_add")
112      * @Template("UnlHomologacionBundle:Area:edit.html.twig")
113      */
114     public function addCarreraAction() {
115         $entity = $this->get('session')->get('area');
116         if ($entity != NULL) {
117             $carrera = new Carrera();
118             $carrera->setUniversidad(null);
119             $entity->addCarrera($carrera);
120             $editForm = $this->createForm(new AreaType(), $entity);
121
122             // $deleteForm = NULL;
123             if ($entity->getId() != NULL) {
124                 $deleteForm = $this->createDeleteForm($entity->getId());

```

```

121     }
122
123     $request = $this->getRequest();
124     $editForm->bindRequest($request);
125
126     return array(
127         'entity' => $entity,
128         'edit_form' => $editForm->createView(),
129         'delete_form' => $deleteForm->createView(),
130     );
131 } else {
132     return new \Symfony\Component\HttpFoundation\Response("NO SE ENCONTRO EL AREA");
133 }
134 }
135
136 /**
137  * Displays a form to edit an existing Area entity.
138  *
139  * @Route("/{id}/edit", name="area_edit")
140  * @Template()
141  */
142 public function editAction($id) {
143     $em = $this->getDoctrine()->getEntityManager();
144
145     $entity = $em->getRepository('UnlHomologacionBundle:Area')->find($id);
146
147     if (!$entity) {
148         throw $this->createNotFoundException('Unable to find Area entity.');
```

```

186     $editForm->bindRequest($request);
187
188
189     if ($editForm->isValid()) {
190
191         // filtra $originalCarreras para evitar aquellas que ya no estan presentes
192         foreach ($entity->getCarreras() as $carrera) {
193             foreach ($originalCarreras as $key => $toDel) {
194                 if ($toDel->getId() === $carrera->getId()) {
195                     unset($originalCarreras[$key]);
196                 }
197             }
198         }
199     }
200
201     // Se quita la relacion entre carrera y area
202     foreach ($originalCarreras as $carrera) {
203         // remove the Task from the Tag
204         $carrera->setArea(null); //->removeElement($task);
205         $em->persist($carrera);
206         //$em->remove($carrera);
207     }
208
209     $em->persist($entity);
210     $em->flush();
211
212     return $this->redirect($this->generateUrl('area', array('id' => $id)));
213 }
214
215 return array(
216     'entity' => $entity,
217     'edit_form' => $editForm->createView(),
218     //'delete_form' => $deleteForm->createView(),
219 );
220 }
221
222 /**
223  * Deletes a Area entity.
224  *
225  * @Route("/{id}/delete", name="area_delete")
226  * @Method("post")
227  */
228 public function deleteAction($id) {
229     $form = $this->createDeleteForm($id);
230     $request = $this->getRequest();
231
232     $form->bindRequest($request);
233
234     if ($form->isValid()) {
235         $em = $this->getDoctrine()->getEntityManager();
236         $entity = $em->getRepository('UniHomologacionBundle:Area')->find($id);
237
238         if (!$entity) {
239             throw $this->createNotFoundException('Unable to find Area entity.');
```

```
251         ->add('id', 'hidden')
252         ->getForm()
253     ;
254     }
255 }
```

4. MAPEO OBJETO RELACIONAL

4.1 Qué es ORM

Las siglas ORM provienen de Object-Relational-Mapping o Mapeo entre Objetos y Relaciones.

Es una técnica de programación que permite vincular los objetos usados en el modelo de la aplicación con una base de datos relacional. (Ver Figura 5)

En las bases de datos relacionales lo que se guardan son datos primitivos, no se puede guardar los objetos que se crean en la aplicación, entonces lo que se hace es convertir los datos del objeto en datos primitivos para que puedan ser almacenados en las tablas correspondientes de la base de datos.

El mapeo objeto-relacional (ORM) se encarga en convertir los objetos en datos primitivos para almacenarlos y viceversa.

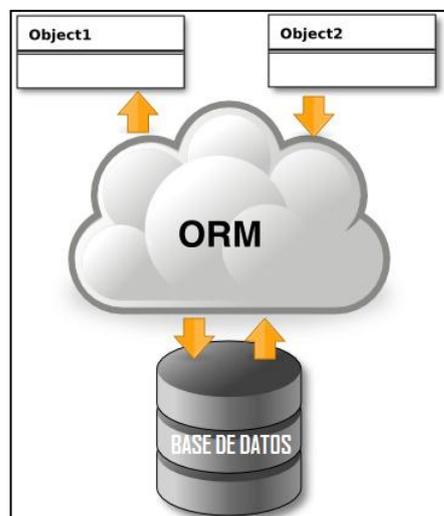


Figura 5. Funcionamiento de un ORM

4.2 Casos de ORM

Según su Multiplicidad

Multiplicidad 1-1

Cada objeto A está asociado con cero o un objeto B y, cada objeto B está asociado con cero o un objeto A.

En el modelo de objetos, este tipo de relación, se representa como una propiedad de tipo de datos de Usuario. Así es común accederla vía `setObjetoB()`, `getObjetoB()`.

En el modelo relacional, cualquiera de las 2 tablas relacionadas implementará una columna con el `Oid`¹³ de la otra tabla, y esta columna será la clave foránea para relacionarlas.

Multiplicidad 1-n

Cada objeto A puede estar asociado con cero o más objetos B, pero cada objeto B está asociado con cero o un objeto A.

En el modelo de objetos, este tipo de relación, se representa como una colección o array de tipo de datos de usuario. Así es común accederla vía `addObjetosB(ObjetoB)`, `removeObjetosB(ObjetoB)`.

Según su navegabilidad

Navegabilidad unidireccional

Una relación unidireccional es aquella que sabe con qué objetos está relacionado, pero dichos objetos no conocen al objeto original.

Navegabilidad bidireccional

¹³ **OID**, acrónimo del inglés Object Identifier (identificador de objeto). utilizado para nombrar un objeto

Una relación bidireccional existe cuando los objetos en ambos extremos de la relación saben del objeto en el extremo contrario.

4.3 Ventajas y desventajas

VENTAJAS

- Rapidez en el desarrollo. La mayoría de las herramientas actuales permiten la creación del modelo por medio del esquema de la base de datos, leyendo el esquema, se puede crear el modelo adecuado.
- Abstracción de la base de datos. Al utilizar un sistema ORM, se puede separar totalmente el sistema de Base de datos que se utiliza con la aplicación; y así si en un futuro se cambia de motor de bases de datos, este cambio no afectará a la aplicación,
- Reutilización. Permite utilizar los métodos de un objeto de datos desde distintas zonas de la aplicación, incluso desde aplicaciones distintas.
- Seguridad. Los ORM suelen implementar sistemas para evitar tipos de ataques como pueden ser los SQL¹⁴ injections. (método de infiltración de código intruso, insertado en el medio del código bueno)
- Mantenimiento del código. Facilita el mantenimiento del código debido a la correcta ordenación de la capa de datos, permite que el mantenimiento del código sea más sencillo.
- Lenguaje propio para realizar las consultas. Estos sistemas de mapeo traen su propio lenguaje para hacer las consultas, lo que hace que los usuarios dejen de utilizar las sentencias SQL para que pasen a utilizar el lenguaje propio de cada herramienta.

DESVENTAJAS

- Tiempo utilizado en el aprendizaje. Este tipo de herramientas suelen ser complejas por lo que su correcta utilización lleva un tiempo que hay que emplear en ver el funcionamiento correcto

¹⁴ **SQL**, acrónimo del inglés Structured Query Language, (Lenguaje de Consulta Estructurado)

- Aplicaciones algo más lentas. Esto es debido a que todas las consultas que se hagan sobre la base de datos, el sistema primero deberá de transformarlas al lenguaje propio de la herramienta, luego leer los registros y por último crear los objetos.

4.4 Producto ORM para symfony2.0

Se distinguen dos ORM para symfony2:

Doctrine

Es un framework ORM para PHP 5.2 y posterior, y trabaja con su lenguaje DQL (Doctrine Query Language) que está inspirado en el HQL¹⁵ de Hibernate. Para crear el modelo, Doctrine crea una clase por tabla y le indica mediante PHP el tipo de datos que se almacena.

Propel

Es otro framework ORM para PHP 5 y superior, se puede acceder y modificar los datos de la base de datos utilizando la lógica de programación orientada a objetos, en vez de utilizar los clásicos Select y Updates de SQL.

4.5 Doctrine 2

Doctrine 2 es un ORM (Object-Relational Mapper) para PHP. La principal tarea del mapeador es traducir de forma transparente los objetos PHP en filas de una base de datos relacional y viceversa. Una de las características claves es escribir Queries en un dialecto propio similar al SQL, llamado Doctrine Query Language (DQL)

Doctrine, se encarga de tratar con la base de datos desde la conexión, generación de SQL, manipulación de datos, transacciones y desconexión.

Symfony2, integra Doctrine2, proporcionando todo el soporte necesario para utilizarlo sin tener que preocuparse por la configuración del mismo.

¹⁵ **HQL**, acrónimo del inglés Hibernate Query Language, que es un lenguaje de consulta sobre objetos,

4.5.1 Persistencia de los datos

Doctrine permite persistir objetos completos a la base de datos y recuperar objetos completos desde la base de datos. Esto funciona asignando una clase PHP a una tabla de la base de datos, y las propiedades de esa clase PHP a las columnas de la tabla (Ver Figura 6):

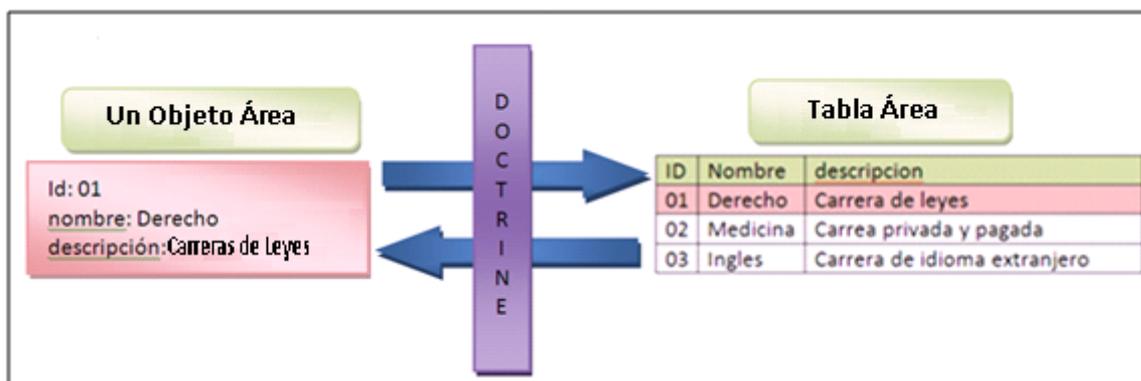


Figura 6. Muestra la persistencia de objetos con Doctrine.

El mapeo en los ORM (Object Relational Mapping) se suele hacer en XML pero Doctrine admite 3 diferentes sintaxis (Anotaciones, XML, YAML).

La siguiente tabla muestra el mapeo realizado de clase Área.php en el proyecto, utilizando Anotaciones.

**TABLA V
MAPEO**

```

homologa/ src/ Un/ HomologacionBundle/ Entity/ Area.php
1  <?php
2  namespace Un\HomologacionBundle\Entity;
3
4  use Doctrine\ORM\Mapping as ORM;
5  use Symfony\Component\Validator\Constraints as Assert;
6  use Doctrine\Common\Collections\ArrayCollection;
7
8
9  /**
10 * @ORM\Entity(repositoryClass="Un\HomologacionBundle\Repository\AreaRepository")
11 */
12 class Area
13 {
14     /**
15     * @ORM\Id
16     * @ORM\Column(type="bigint")
17     * @ORM\GeneratedValue(strategy="AUTO")

```

```

18     */
19     protected $id;
20
21     /**
22     * @ORM\Column(type="string", length=100)
23     * @Assert\NotBlank()
24     */
25     protected $nombre;
26
27     /**
28     * @ORM\Column(type="text")
29     */
30     protected $description;
31
32
33     /**
34     * @ORM\OneToMany(targetEntity="Carrera", cascade={"all"}, mappedBy="area")
35     */
36     protected $carreras;
37 }

```

4.5.2 Consulta de objetos

Symfony utiliza junto con doctrine una serie de formas de acceder a los datos. Lo más normal es utilizar repositorios. Estos tienen funciones de búsqueda que se heredan de una clase Repositorio del framework (de ahí lo interesante de tener id en la BBDD), pero aun más interesante es crear otras búsquedas pudiendo hacer una búsqueda según las necesidades del desarrollo.

Para crear búsquedas Doctrine puede hacerlo de tres formas diferentes:

1. Una es creando consultas nativas en SQL con el método del gestor de entidades "createNativeQuery".
2. También puedes usar DQL directamente con el método "createQuery"
3. Y la tercera opción es crear una instancia al creador de consultas "QueryBuilder" que permite poner por secciones lo que se quiere hacer y el propio Doctrine generara internamente la DQL y la lanzara.

El utilizar DQL, no permite que los ataques "SQL Injection" porque los valores son pasados en su forma tipo, esto hace que no puedan incluirse sentencias sino solo valores. Lo malo es que puede resultar algo complejo de entender, especialmente para consultas complejas.

La siguiente tabla muestra un ejemplo de la utilización de DQL directamente con el método "createQuery", usado en la clase AreaRepository.php para crear búsquedas.

TABLA VI
DQL (createQuery)

homologa/ src/ Unl/ HomologacionBundle/ Repository/ AreaRepository.php	
1	<?php
2	
3	namespace Unl\HomologacionBundle\Repository;
4	
5	use Doctrine\ORM\EntityRepository;
6	
7	/**
8	* AreaRepository
9	*
10	* This class was generated by the Doctrine ORM. Add your own custom
11	* repository methods below.
12	*/
13	class AreaRepository extends EntityRepository
14	{
15	public function findByNombre(\$nombre){
16	\$em = \$this->getEntityManager();
17	\$query = \$em->createQuery('SELECT a FROM UnlHomologacionBundle:Area a WHERE a.nombre
18	\$query->setParameter('nombre', \$nombre.'%');
19	\$entities = \$query->getResult();
20	return \$entities;
21	}
22	}

5. SUBVERSION

SVN es un sistema de control de versiones usado para que varios desarrolladores puedan trabajar en un mismo proyecto en forma más o menos ordenada. Tiene una arquitectura cliente servidor con controles de concurrencia para cuando varios desarrolladores están trabajando en el mismo archivo y funciona más o menos así. En algún servidor se monta un repositorio SVN. En este lugar se van a registrar los cambios (revisiones) y los logs que se vayan generando. El cliente de SVN se baja una copia local de alguna revisión (generalmente la última), el desarrollador hace los cambios y los sube al servidor para que estén disponibles para los otros desarrolladores editando un comentario de lo modificado.

Entre los servidores svn/git para almacenar los proyectos de forma gratuita se encuentran: Repositorio SVN Assembla, Repositorio SVN Google, Repositorio SVN MyVersionControl, Repositorio SVN ProjXpert, Unfuddle, BeanStalk, ProjectLocker.

5.1 Repositorio SVN google

Google proporciona un servicio gratuito de Alojamiento de Proyectos, Google Code para lo cual se debe utilizar una cuenta de Google y acceder a este servicio.

Google Code permite:

- Crear proyectos instantáneos sobre cualquier tema,
- Saber información muy detallada acerca de quien realiza un cambio y cuando lo realiza,
- Consultar código fuente integrado y utilizar herramientas de revisión de código para facilitar la visualización de código, la revisión de contribuciones y el mantenimiento de una base de código de gran calidad,
- Marcar como destacados y actualizar flujos que facilitan el seguimiento de los proyectos y los desarrolladores que te interesan.

e. Materiales y Métodos

1. Métodos utilizados

1.1 Método deductivo

El método deductivo permite partir de generalizaciones, es decir de teorías y categorías, para poder explicar adecuadamente hechos concretos o particulares.

Durante el desarrollo del presente proyecto, la investigación de teorías fue imprescindible para descubrir los nuevos conceptos asociados a metodologías y herramientas de programación, tales como el concepto de framework, orm, plantillas, etc.

Una vez comprendido el contexto teórico en el que se fundamentan las herramientas para construir implementaciones de dichos conceptos, deducir que Symfony, Doctrine o Twig son solo aplicaciones de dichos conceptos en el ámbito del desarrollo basado en el lenguaje PHP.

El método deductivo permitió también la aplicación de los conceptos de la ingeniería de software durante la etapa de diseño para proponer las soluciones específicas de cada uno de los componentes. El estudio primario de teoría permitió que esos conceptos se vean aplicados en la construcción de la solución para el sistema de homologación

Este método también permitió estructurar el presente informe, puesto que todo el detalle de lo planteado, intenta primero definir un contexto teórico general que posteriormente muestra la aplicación concreta de dicho conocimiento.

1.2 Método inductivo

El método inductivo se uso durante la etapa de determinación de requerimientos y análisis partiendo de la experiencia particular de los usuarios y de las entrevistas mantenidas con ellos para generar un modelo de dominio general y aplicable a todas las situaciones específicas encontradas.

A partir de la experiencia obtenida con el desarrollo de la aplicación y de los inconvenientes presentados a lo largo del mismo, se pudo obtener conocimiento general acerca de la aplicación del framework, el mismo que fue usado durante el planteamiento de las conclusiones de este trabajo, las cuáles fueron redactadas y especificadas como conocimiento general.

1.3 Método experimental

Este método se usó a lo largo del desarrollo de la aplicación para generar las pruebas iniciales para la aplicación de los diversos conceptos aprendidos bajo el método deductivo.

La experimentación para mostrar el comportamiento de las herramientas de programación se basó en la práctica bajo condiciones controladas de trabajo, lo cual permitió establecer la mejor forma de aplicación de las herramientas en el contexto del proyecto.

2. Instrumentos para recolección de información

2.1 Entrevista

Entrevistas, estas se efectuaron cada vez que fueron necesarias para aseverar los avances en cada una de las fases del proyecto, las mismas fueron realizadas con las autoridades y personal requerido.

2.2 Observaciones

Para poder determinar la situación actual de la institución en lo referente al proceso de los tramites de homologación, como es el medio en el cual se desarrolla y con qué recursos cuenta actualmente para poder llevar en marcha el desarrollo de nuestra investigación.

3. Metodología de desarrollo

Para llevar a cabo las diferentes fases en el desarrollo del presente proyecto se utilizó la Metodología ICONIX, por ser la que más se adapta a las necesidades del proyecto. A continuación se describen las diferentes fases que implicó el desarrollo con esta metodología.

3.1 Descripción de ICONIX

Análisis de requerimientos

Para poder determinar los requerimientos se mantuvo constantes diálogos directos con los funcionarios de secretaria, docentes del A.E.I.R.N.N.R y el Coordinador de la Carrera de Sistemas quienes supieron brindar valiosa información para poder determinar los requerimientos del sistema, y así poder elaborar el Modelo de Dominio, un Diagrama de Clases simplificado. Con este modelo, se realizó un pequeño prototipo de la interfaz gráfica, para luego identificar los casos de uso del sistema y los actores involucrados; para esto se realizó los diagramas de casos de uso.

Análisis y diseño preliminar

En esta etapa se realizó la descripción de los casos de uso, con un flujo principal de acciones y los posibles flujos alternos, luego se continúa con la elaboración de los diagramas de robustez por cada caso de uso, estos diagramas muestran las interacciones entre los objetos participantes de cada caso de uso.

Posteriormente con estos diagramas y los casos de uso descritos se actualizó el modelo de Dominio realizado en la fase anterior.

Esta etapa sirvió para dar una solución lógica al problema, indicando una posible guía de cómo va a funcionar el sistema en forma general.

Diseño

En esta etapa se elaboraron los diagramas de secuencia, los mismos que se obtienen a partir de la descripción de los de caso de uso.

Luego se actualizan el diagrama de clase permitiendo tener un acercamiento claro del funcionamiento de la aplicación

Implementación

En esta etapa además de la programación modular e integración, se realizaron test unitarios de código así como pruebas de validación con los usuarios.

f. Resultados

1. ANÁLISIS DE REQUERIMIENTOS

Esta primera etapa de la metodología ICONIX, permitió conocer el proceso que se sigue en las distintas actividades realizadas por el personal del AEIRNNR, y los principales inconvenientes que se presentan al momento de ejecutarlas, además a través de ella se pudo determinar el proceso para el trámite de homologación se los realizan de forma manual, lo que lo convierte en un proceso lento y tedioso, dando como resultado la demora y aglomeración de trámites de homologación.

Por otro lado se pudo recolectar información que fue muy útil al momento de realizar el análisis sobre el proceso que se lleva a cabo al efectuar un trámite de homologación, y posteriormente permitió desarrollar varias tareas que se encuentran inmersas dentro de la etapa de análisis de requerimientos, entre ellas se puede mencionar:

- ✦ Determinación de requerimientos del sistema
- ✦ Obtención del modelo de dominio
- ✦ Identificación de los caos de uso
- ✦ Diagramas de casos de uso

Todas estas tareas se detallan más adelante con sus respectivos resultados

A continuación se presenta el informe de Especificación de Requisitos del Software, que se basa en el Estándar IEEE-STD-830-1998.

2. ESPECIFICACIONES DE LOS REQUISITOS DEL SOFTWARE

1. Introducción

A través del presente documento se realiza una especificación de Requisitos de Software (ERS) para el Sistema de Apoyo al Trámite de Homologación para el AEIRNNR de la Universidad Nacional de Loja. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE. Práctica recomendada para la Especificación de Requisitos de Software ANSI/IEEE 830/1998.

1.1 Propósito

El propósito de esta Especificación de Requisitos de Software (ERS), es definir las especificaciones funcionales, no funcionales para el desarrollo de un sistema de Apoyo al trámite de Homologación, que permita la homologación tanto entre carreras de la UNL como de otras universidades. Este sistema será utilizado por estudiantes, profesores, secretario(a) de la carrera, y coordinador(a) de la carrera.

1.2 Alcance

Esta especificación de requisitos está dirigida al desarrollo de una herramienta informática de apoyo al trámite de Homologación para el AEIRNNR, el cual es el objetivo principal del presente proyecto. Teniendo como principales acciones a realizar: Administración de Peticiones de Homologación, Administración de Mallas Curriculares, Emisión del Resultado de la Homologación, Visualización e Impresión de los Resultados de la Homologación

1.3 Definiciones, siglas y abreviaturas

TABLA VII
ANÁLISIS DE REQUERIMIENTOS- DEFINICIÓN DE SIGLAS Y ABREVIATURAS

Nombre	Descripción
ERS	Especificación de Requisitos de Software
AEIRNN	Área de Energía Industria y Recursos Naturales no Renovables
UNL	Universidad Nacional de Loja
SIST.	Sistema.
RQF	Requerimiento Funcional.
RQNF	Requerimiento no Funcional.

1.4 Referencias

TABLA VIII
ANÁLISIS DE REQUERIMIENTOS- REFERENCIAS

Referencia	Título	Ruta
IEEE	Standard IEEE 830-1990	http://www.ctr.unican.es/asignaturas/is1/IEEE830_esp.pdf

1.5 Apreciación global

El presente documento consta de tres secciones.

En la primera sección se encuentra la introducción, la misma que proporciona una visión general de la especificación de recursos del software.

En la segunda sección se encuentra una descripción general del sistema, con el fin de conocer las principales funciones que este debe realizar, los datos asociados, y los factores, restricciones, supuestos, y dependencias que afectan al desarrollo, sin entrar en excesivos detalles

En la tercera sección se encuentran los requisitos que debe satisfacer el sistema, los cuales se definen detalladamente.

2. Descripción global

2.1 Perspectiva del producto

El Sistema de Homologación a desarrollar es independiente, con lo que tiene total autonomía, ya que no actúa con otro sistema.

- **Interfaces con el usuario**

La interfaz con la que interactuará el usuario es en entorno Web y será manejada por los estudiantes, secretaria, docente, y coordinador de cada carrera del AEIRNNR. Para su correcto acceso se deberá introducir la dirección correcta en un navegador Web. Tras la conexión con la página principal de la aplicación, el estudiante tendrá la opción de registrarse para tener acceso a los servicios de la aplicación. La secretaria, docente, y coordinador de cada carrera del AEIRNNR serán registrados por el Administrador del sistema para poder tener acceso a los servicios de la aplicación.

La interfaz para el estudiante tendrá la opción de permitirle realizar una Petición de Homologación, Visualizar el estado en el que se encuentra la Petición realizada.

La interfaz para la secretaria permitirá visualizar todas las Peticiones de Homologación realizadas, registrar una Nueva Petición de Homologación, Editar una Petición de Homologación existente, Asignar una Petición de Homologación a un Docente del AEIRNNR, Visualizar el estado en el que se encuentra la Petición de Homologación e Imprimir el Resultado de la Petición de Homologación.

La interfaz para el docente permitirá visualizar las Peticiones de Homologación asignadas a su persona, realizar la Homologación de estas Peticiones de homologación, visualizar el resultado de la Homologación realizada, e Imprimir un reporte o resultado de la Homologación realizada.

El administrador tendrá la opción de: crear los diferentes tipos de usuarios del sistema y asignarles cualquier rol al usuario creado; editar los datos de los usuarios del sistema, con la excepción Nombre de Usuario y Clave; Crear Universidades, Carreras, Áreas, Mallas Curriculares o así mismo podrá editar los datos de Universidades, Carreras, Áreas y Mallas Curriculares existentes.

A través de los requerimientos de los usuarios se pudieron realizar los siguientes prototipos (pantallas) de interfaces para el usuario, los cuales podrán cambiar según el desarrollo del sistema.

INGRESAR SOLICITUD HOMOLOGACION [x]

Cedula

Nombres

Apellidos

Carrera Origen

Universidad Origen

Carrera

Carrera Destino

Figura 7. Prototipo de pantalla ingresar solicitud de homologación



Figura 8. Prototipo de pantalla consultar petición de homologación

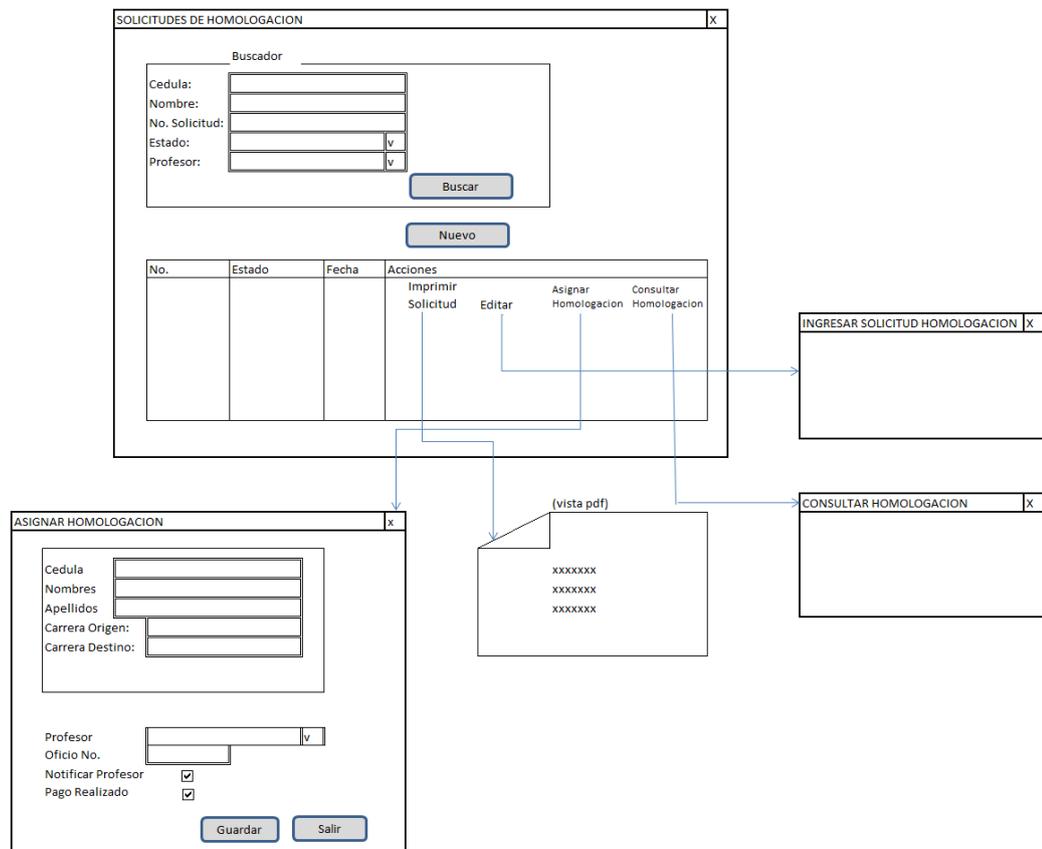


Figura 9. Prototipo de pantalla solicitudes de homologación y asignar homologación.

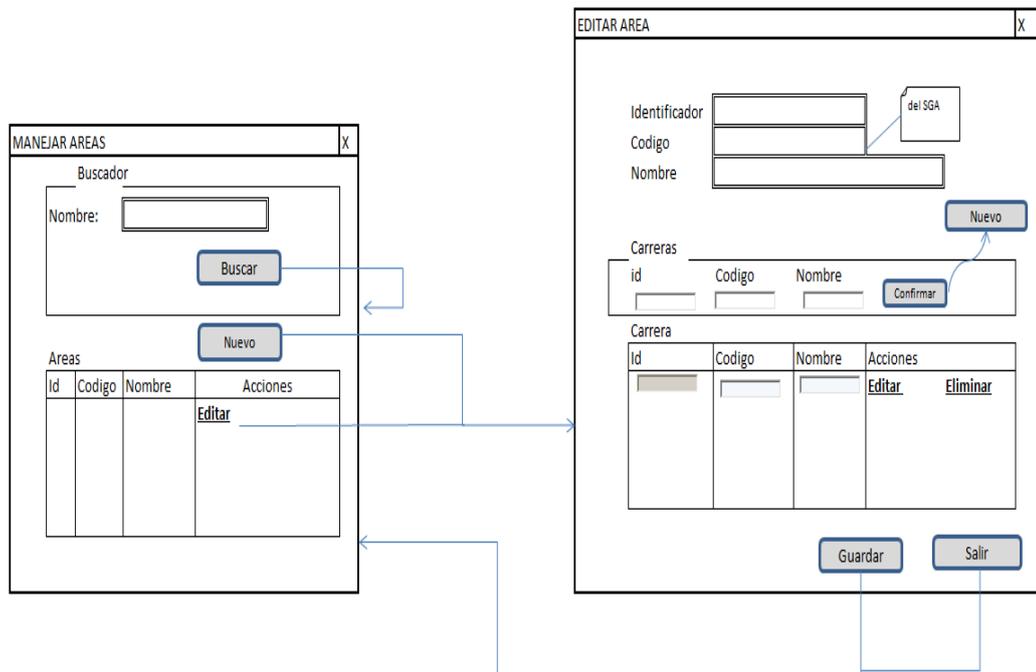


Figura 10. Prototipo de pantalla manejar áreas y editar área.

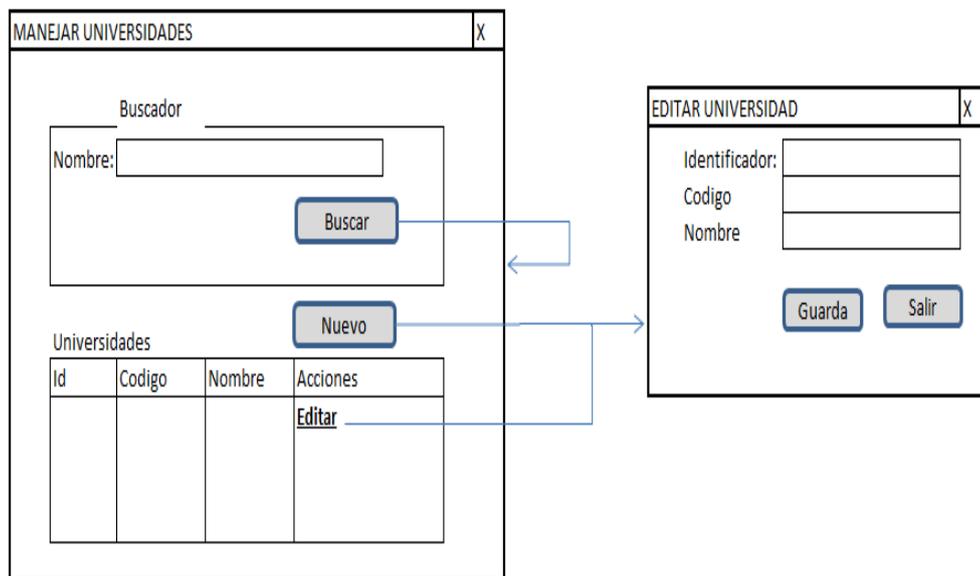


Figura 11. Prototipo de pantalla manejar universidades y editar universidad.

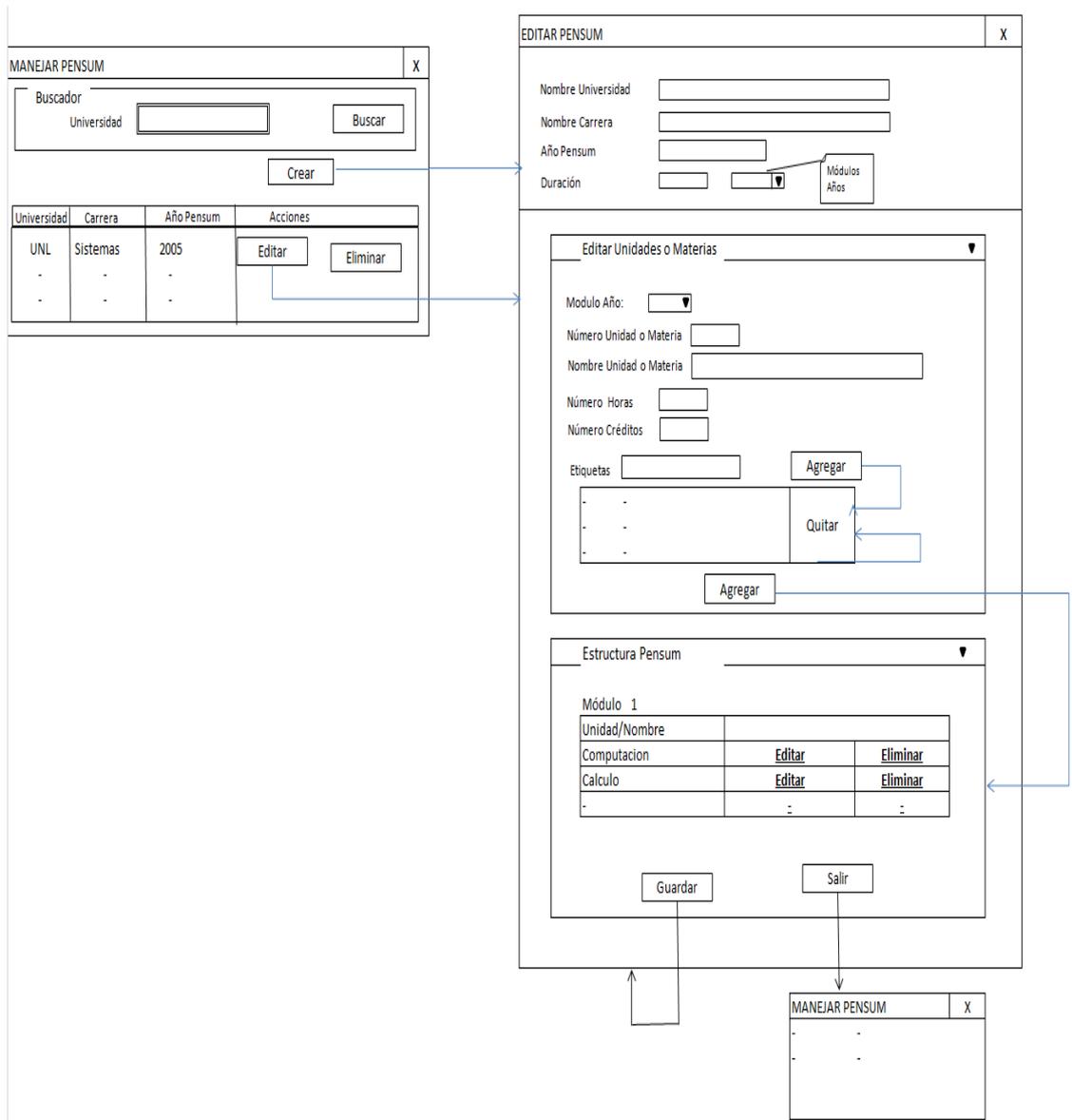


Figura 12. Prototipo de pantalla manejar pensum y editar pensum

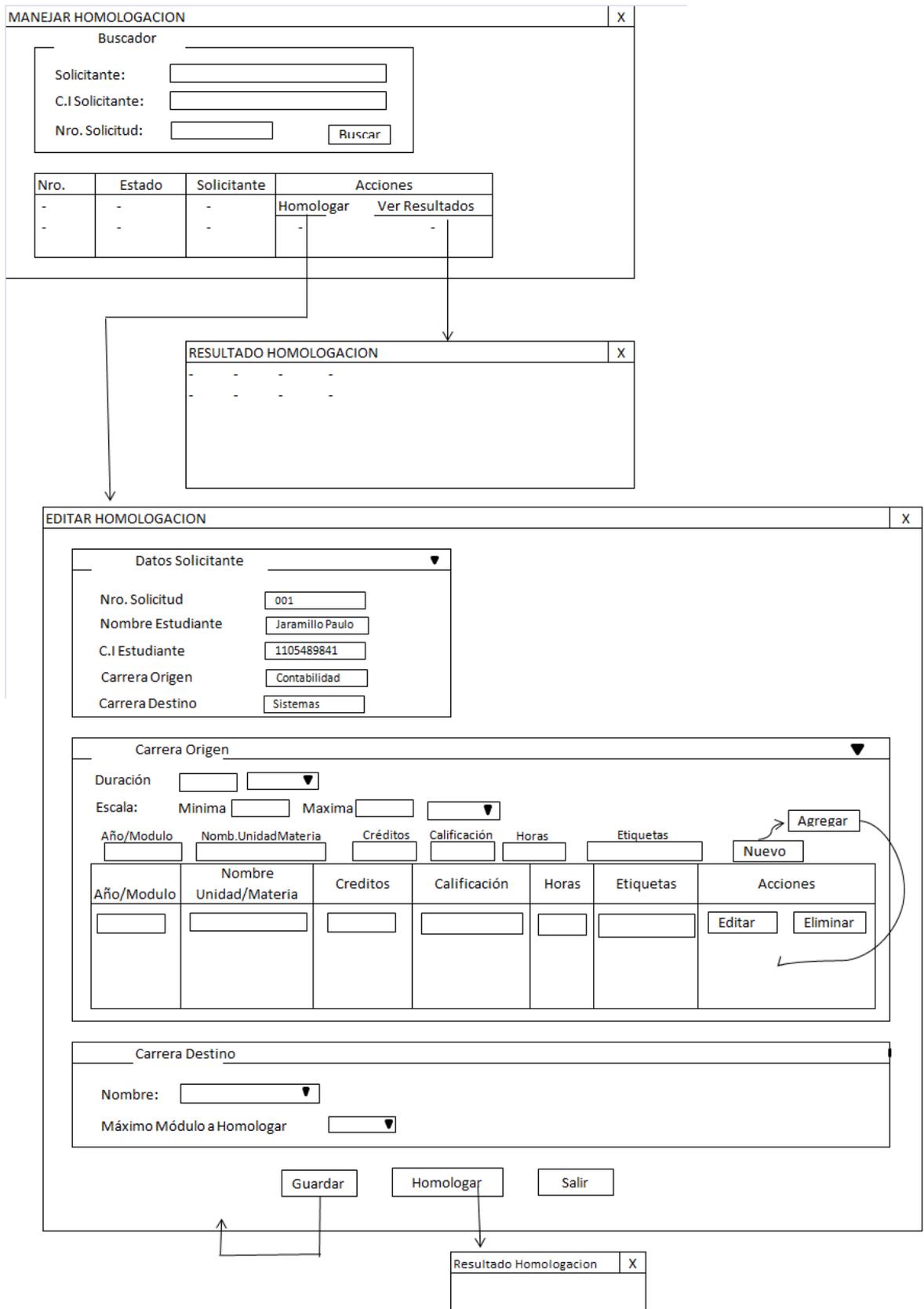


Figura 13. Prototipo de pantalla manejar homologación y editar homologación.

Resultado Homologacion

Datos del Peticionario

Datos Solicitante

Nombres y Apellidos: Maria Reyes

Universidad de Procedencia: Universidad del Pacifico

Carrera de Procedencia: Contabilidad

Periodo Estudios Univer. Procedencia: 2010

Pensum Estudios Procedencia	Mailla Curricular a ser Admitido	Nro. Creditos	Calificaci on	Porcentaj e >=80%	Reconoce		Observaciones	Ver Detalles
					Si	No		
Algebra Lineal	Modulo III U1 Matemáticas	13	21/30	90%	<input type="checkbox"/>	<input type="checkbox"/>		
	U2 Algebra Lineal				<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	U3 Calc. Diferencial				<input type="checkbox"/>	<input type="checkbox"/>		
Fisca Mate. Discretas	U5 Fisica	13	20/30	90%	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	U6 Mate Discretas	5		20%	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
Estadística	Modulo IV Estadística	10	23/30	90%	<input type="checkbox"/>	<input type="checkbox"/>		

Imprimir Salir

Figura 14. Prototipo de pantalla resultado de homologación

- **Interfaces con el hardware**

La interfaz hardware necesitará el uso de un computador debidamente configurado con conexión de internet, con sus respectivos dispositivos de entrada y salida como lo son mouse, teclado e impresora.

- **Interfaces con el software**

El usuario de la aplicación requerirá de un sistema operativo en el cual se pueda utilizar un navegador web como puede ser Internet Explorer, Mozilla o Chrome.

En el servidor se utilizará una base de datos para el correcto almacenamiento de la información relativa al soporte del sistema.

- **Interfaces de comunicaciones**

Para la comunicación se requerirá de un dispositivo que soporte acceso a internet y un protocolo de transferencia de recursos HTTP debidamente configurados. Además será necesario que el servidor soporte un gran número de conexiones externas simultáneas.

- **Restricciones de memoria**

Para el funcionamiento de la aplicación por parte de los usuarios no se necesitara un gran uso de memoria. No obstante, en el servidor será necesaria una mayor capacidad en los dispositivos de almacenamiento para la información requerida.

- **Funcionamientos**

Para que la aplicación funcione correctamente será necesario un servidor conectado permanentemente a internet. Ya que la aplicación utiliza tanto una base de datos como un servidor web es posible que se produzcan algunos fallos ajenos a la aplicación, en tal caso será competencia de personal de mantenimiento solucionar estos inconvenientes técnicos.

Con respecto al usuario de la aplicación será necesario que conozca la URL exacta a la cual debe conectarse para que pueda hacer uso de los servicios de la aplicación, además deberá tener conocimientos mínimos de navegación web, como una correcta configuración de su sistema local.

2.2 Funciones del producto

Esta sección se la ha dividido en relación a los usuarios de la aplicación así tenemos las siguientes funciones asignadas:

Funciones del administrador:

Administrar usuarios

Administrar áreas

Administrar carreras

Administrar mallas curriculares

Funciones de la secretaria(o):

Administrar las peticiones de homologación

Funciones del docente:

Realizar homologaciones

Emitir resultados de las homologaciones

Funciones del estudiante:

Realizar peticiones de homologación

Consultar estado de la petición de homologación

2.3. Características del usuario

**TABLA IX
ANÁLISIS DE REQUERIMIENTOS- CARACTERÍSTICAS DEL USUARIO**

Tipo de usuario	Administrador
Formación	Informática
Actividades	Control y Manejo del sistema en general

Tipo de usuario	Secretaria(o)
Formación	Secretariado
Actividades	Editar, Asignar y Consultar Peticiones de Homologación

Tipo de usuario	Docente
Formación	Educador
Actividades	Realizar el proceso de Homologación, visualizar el resultado de la homologación e imprimir este resultado

Tipo de usuario	Estudiante(o)
Formación	Bachiller
Actividades	Solicitar Petición de Homologación. Visualizar en qué estado se encuentra la Petición de Homologación realizada.

2.4. Restricciones

Las restricciones o limitaciones que se pueden presentar en el desarrollo de la aplicación es la utilización de una herramienta nueva como lo es el Framework Symfony2 utilizado con el lenguaje de programación PHP, por lo que llevará un poco más de tiempo el desarrollo de la aplicación.

2.5 Atención y dependencias

Los requisitos antes descritos son estables, ya que si se realizan cambios en alguna de las herramientas necesarias para el funcionamiento correcto de la aplicación, se deberían reconfigurar todas las demás.

2.6 Prorratear los requisitos

Debido a las cualidades del proyecto, la aplicación será estable y no necesitará de ninguna otra entrega. Por lo que no se tiene previsto realizar modificaciones posteriores.

3. Requisitos específicos

Para la descripción de los requerimientos específicos se utiliza la Plantilla de A.1

3.1 Requisitos de las interfaces externas

3.1.1 Interfaces del usuario:

La interfaz del usuario será sencilla. Contendrá un título de la ventana, un menú y un área de contenidos.

Así mismo la interacción entre la aplicación y el usuario se realizará mediante pantallas que contengan: ventanas, formularios, botones, etiquetas, listas y menús

Desde el menú principal se accederá a cada una de las acciones que el usuario podrá realizar y estas se mostrarán en el área de contenidos. El menú contendrá al menos las siguientes secciones:

- Homologar
- Universidades
- Áreas
- Carreras
- Mallas Curriculares
- Usuarios

Cada una de estas secciones corresponde a alguna de las funciones requeridas, y por tanto sus contenidos se describen en la **Sección 3.2 Requisitos funcionales**.

3.1.2 Interfaces hardware

Se necesitará un computador que actuará de servidor web (conectado a internet) y servidor de bases de datos.

Para hacer uso de la aplicación las características del hardware que se tendrán en consideración son: equipos computacionales en perfecto estado, adaptadores de red, procesador 1.66 Ghz o superior, Memoria RAM mínima de 256 Mb o superior, Mouse, Teclado.

3.1.3 Interfaces del software

La presente aplicación va a ser desarrollada bajo el sistema operativo Ubuntu 11.10, sobre este sistema operativo se instalará Apache como servidor, la versión 2.2., MySQL versión 5.1.37 como gestor de base de datos, como lenguaje de implementación PHP versión 5.3.0 y el framework Symfony2; por último NetBeans ayudará a programar en PHP.

Los usuarios que accedan a la aplicación no necesitarán de un sistema operativo concreto, ya que para acceder a la aplicación se requerirá de un sistema operativo en el cual se pueda utilizar un navegador web entre los que más se emplean actualmente: Microsoft Internet Explorer 7, Mozilla Firefox 3.6 y Google Chrome 4.1. La aplicación va a ser desarrollada con una arquitectura de tres capas.

3.1.4 Interfaces de comunicación

La interfaz de comunicación que se necesitará para la aplicación será de un dispositivo que soporte acceso a internet y un protocolo de transferencia de recursos HTTP debidamente configurados para la parte de comunicación entre los computadores y el servidor. Además será necesario que el servidor soporte un gran número de conexiones externas simultáneas.

3.2 Requisitos funcionales

REQUERIMIENTOS FUNCIONALES

1. ADMINISTRADOR

TABLA X
ANÁLISIS DE REQUERIMIENTOS
REQUERIMIENTOS FUNCIONALES-ADMINISTRADOR

Número de requisito	RQF1.1
Nombre de requisito	El sistema debe Administrar las Mallas Curriculares de las diferentes carreras del AEIRNNR
Descripción del Requerimiento	Previo al registro e ingreso al sistema, el administrador podrá crear, editar y eliminar mallas curriculares, para la creación y/o edición el administrador deberá ingresar los siguiente datos (Nombre de Universidad, Nombre de la Carrera, Año de creación de la malla curricular, niveles, unidad de Tiempo, ya sean ciclos, módulos, años)
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF1.2
Nombre de requisito	El sistema debe administrar los contenidos de las diferentes unidades de cada módulo de las carreras del AEIRNNR
Descripción del Requerimiento	Previo al registro e ingreso al sistema, el administrador podrá crear, editar y eliminar unidades, así como los contenidos de las mismas. Para la creación y/o edición de las unidades el administrador deberá ingresar los siguientes datos (nombre de la unidad, número de horas, numero de créditos, escoger si esta unidad es obligatoria o no, e ingresar todos los contenidos de las mismas)
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF1.3
Nombre de requisito	El sistema debe crear menús personalizados de acuerdo a los roles de los usuarios
Descripción del Requerimiento	Previo al registro e ingreso al sistema el administrador podrá crear usuarios y asignarles los roles, correspondientes. El sistema al autenticar y validar el usuario automáticamente ingresara al sistemas y presentara el menú correspondiente al rol del usuario.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF1.4
Nombre de requisito	El sistema debe Validar el correcto ingreso de la información de los Pensum de cada carrera
Descripción del Requerimiento	Al realizar, la creación y/o edición de las unidades, descritas en el RQF 1.2 el sistema validara de forma automática, que los campos se han llenados con los datos requeridos (en el campo de nombre de la unidad y contenidos solo se podrá ingresar texto. En los campos de número de horas y número de créditos solo se podrá ingresar números). Así mismo se validara que el número de horas y créditos se excedan del límite requerido
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

2. SECRETARIA(O)

TABLA XI
ANÁLISIS DE REQUERIMIENTOS
REQUERIMIENTOS FUNCIONALES- SECRETARIA(O)

Número de requisito	RQF2.1
Nombre de requisito	El sistema debe permitir visualizar el estado del Proceso de Homologación (Registro de estados) a través de internet
Descripción del Requerimiento	Previo al registro e ingreso al sistema la secretaria(o) podrá visualizar un listado de las peticiones de homologación, las mismas que presentan el ultimo estado en el que se encuentra la homologación. También podrá acceder a visualizar el registro de estados, el cual presentara los estados por los que ha pasado la petición de homologación con las fechas.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF2.2
Nombre de requisito	El sistema debe Imprimir el resultado de la Petición de Homologación
Descripción del Requerimiento	Previo al registro e ingreso al sistema la secretaria(o) podrá ingresar a imprimir el resultado de una petición de homologación. Esta acción se activara solo si la petición de homologación tiene el estado de homologada
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF2.3
Nombre de requisito	El sistema debe visualizar los porcentajes de la homologación de las carreras
Descripción del Requerimiento	Previo al registro e ingreso al sistema la secretaria(o) podrá ingresar a visualizar el resultado de una petición de homologación. Esta acción se activara solo si la petición de homologación tiene el estado de homologada
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF2.4
Nombre de requisito	El sistema debe llevar un registro de docentes con sus respectivas homologaciones asignadas
Descripción del Requerimiento	Previo al registro e ingreso al sistema la secretaria(o) podrá visualizar un registro de todas las peticiones de homologación que se han realizado con los nombres de los docentes asignadas a cada petición de homologación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

3. DOCENTE

**TABLA XII
ANÁLISIS DE REQUERIMIENTOS
REQUERIMIENTOS FUNCIONALES- DOCENTE**

Número de requisito	RQF3.1
Nombre de requisito	El sistema debe permitir la homologación entre carreras del AEIRNNR de la UNL
Descripción del Requerimiento	Previo al registro e ingreso al sistema, y previo al ingreso de las mallas curriculares de las carreras de origen y destino el docente podrá escoger la acción de homologar en la lista de peticiones de homologación. Esta comparación entre los contenidos de las dos mallas curriculares el sistema lo hará de forma automática, presentando el resultado de la homologación con los porcentajes con los que fueron homologadas las unidades.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF3.2
Nombre de requisito	El sistema debe permitir la homologación de las carreras del AEIRNNR con carreras de otras universidades
Descripción del Requerimiento	Previo al registro e ingreso al sistema el docente, podrá ingresar mallas curriculares pertenecientes a la carrera origen del peticionario(estudiante), ya sea esta de la UNL o de otra universidad
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF3.3
Nombre de requisito	El sistema debe permitir realizar la homologación mediante contenidos(temas)
Descripción del Requerimiento	Previo al registro e ingreso al sistema y previo al ingreso de la malla origen el docente podrá escoger homologar una petición, para lo cual se crear un registro de homologación en el cual deberá ingresar los siguientes datos: escala mínima de calificación de la unidad origen, escala máxima de calificación de la unidad origen, el numero de módulos máximo a homologar(que depende del número de módulos(niveles) cursados por el peticionario en la universidad origen(universidad de procedencia), la calificación de cada unidad, y además podrá también ingresar contenidos o temas alternos en cada unidad con

	sinónimos o equivalentes a los contenidos de la malla curricular destino, para realizar una comparación más precisa entre contenidos.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF3.4
Nombre de requisito	El sistema debe permitir visualizar el estado del Proceso de Homologación (Registro de estados) a través de internet
Descripción del Requerimiento	Previo al registro e ingreso al sistema el docente podrá visualizar un listado de las peticiones de homologación que le han sido asignadas, las mismas que presentan el último estado en el que se encuentra la homologación. También podrá acceder a visualizar el registro de estados, el cual presentara los estados por los que ha pasado la petición de homologación con las fechas.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF3.5
Nombre de requisito	El sistema debe permitir Imprimir el resultado de la Homologación
Descripción del Requerimiento	Previo al registro e ingreso al sistema el docente podrá ingresar a imprimir el resultado de una petición de homologación. Esta acción se activara en el listado de las peticiones de homologación solo si la petición tiene el estado de homologada
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF3.6
Nombre de requisito	El sistema debe permitir al usuario(Docente) asignar contenidos(temas) de las unidades de cada carrera a homologar y almacenarlos
Descripción del Requerimiento	
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF3.7
Nombre de requisito	El sistema debe permitir visualizar los porcentajes de la homologación de las carreras
Descripción del Requerimiento	Previo al registro e ingreso al sistema el docente podrá ingresar a visualizar el resultado de una petición de homologación. Esta acción se activara solo si la petición de homologación tiene el estado de homologada
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF3.8
Nombre de requisito	El sistema debe validar el correcto ingreso de la información de los Pensum de cada carrera
Descripción del	Previo al registro e ingreso al sistema el docente podrá

Requerimiento	realizar, la creación y/o edición de las unidades de la carrera origen del peticionario. El sistema validara de forma automática, que los campos sean llenados con los datos requeridos (en el campo de nombre de la unidad y contenidos solo se podrá ingresar texto. En los campos de número de horas y número de créditos solo se podrá ingresar números). Así mismo se validara que el número de horas y créditos se excedan del límite requerido
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF3.10
Nombre de requisito	El sistema debe permitir utilizar un mínimo de contenidos(temas) para realizar la Homologación
Descripción del Requerimiento	
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

4. ESTUDIANTE

**TABLA XIII
ANÁLISIS DE REQUERIMIENTOS
REQUERIMIENTOS FUNCIONALES- ESTUDIANTE**

Número de requisito	RQF4.1
Nombre de requisito	El sistema debe permitir que el estudiante ingrese una solicitud de homologación vía internet
Descripción del Requerimiento	Previo al registro e ingreso al sistema el estudiante podrá realiza una petición de homologación en la cual deberá registrar los siguientes datos: Nombre de la Universidad Origen, nombre de la Carrera Origen, nombre de la carrera destino.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RQF4.2
Nombre de requisito	El sistema debe permitir visualizar el estado del Proceso de Homologación (Registro de estados) a través de internet
Descripción del Requerimiento	Previo al registro e ingreso al sistema el estudiante podrá visualizar un listado de las peticiones de homologación realizadas. Las mismas que presentan el último estado en el que se encuentra la homologación. También podrá acceder a visualizar el registro de estados, el cual presentara los estados por los que ha pasado la petición de homologación con las fechas.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

3.3 Requisitos de desarrollo

Entre los requisitos para el desarrollo de la presente aplicación, se tendrá dos computadores que servirán para el desarrollo de código de la aplicación.

El ciclo de vida elegido para desarrollar el sistema será considerado en cuatro etapas que son: análisis, diseño, codificación y prueba.

El tiempo de respuesta a consultas, actualizaciones, y modificaciones de la información es un estimado de 10 segundos dependiendo del computador a emplear.

3.4 Requisitos del banco de datos lógicos

Se contará con una BBDD, la misma que almacenará la información que utilizará la presente aplicación.

Tanto los usuarios como el administrador podrán accederán a la Base de Datos cada vez que requieran consultar información o modificarla.

3.5 Restricciones de diseño

La aplicación no tiene ningún tipo de restricción por estándares o limitaciones del hardware.

El formato empleado para los informes será PDF.

3.6 Atributos del sistema de software

3.6.1 Fiabilidad

El aplicación permitirá almacenar datos de manera correcta y completa en la base de datos, además de ofrecerá seguridad y confidencialidad.

3.6.2 Disponibilidad

La aplicación estará disponible en todo momento para mayor comodidad del usuario ya que es una aplicación web.

3.6.3 Seguridad

La aplicación tiene diferentes perfiles de usuario: Administrador, Estudiante, Secretaria(o) y Docente, lo cual permitirá implementar un esquema de seguridad y restricciones en el acceso a la base de datos.

3.6.4 Portabilidad

La presente aplicación se desarrollará bajo la arquitectura Cliente/Servidor además se tendrá en cuenta las distintas plataformas utilizables y se ha intentado no utilizar nada en concreto de ningún sistema para obtener una compatibilidad con todas las plataformas posibles.

3.7 Organizar los requisitos específicos.

3.7.1 Modo del sistema

En la presente aplicación no se dispondrá de ningún modo de sistema.

3.7.2 Clases de usuario

Las clases de usuario, con las que se consideran son: Administrador, Estudiante, Docente y Secretaria(o)

3.7.3 objetos

En la aplicación se podrá considerar como objetos en cuanto a personas a: Administrador, Estudiante, Docente y Secretaria(o) ya que son quienes utilizarán la aplicación y en cuanto a objetos dentro del sistema tendremos Universidades, Carreras, Mallas Curriculares, etc.

3.7.4 Rasgo

Para la presente aplicación los rasgos principales serían la agregación, consulta y modificación en la base de datos.

3.7.5 Estímulo

Dependerá de la información obtenida para proceder a realizar la acción adecuada en la aplicación.

3.7.6 Contestación

Confirmaciones de peticiones de homologación, respuestas de consultas realizadas.

RESUMEN DE REQUERIMIENTOS DEL SISTEMA DE HOMOLOGACIÓN

REQUERIMIENTOS FUNCIONALES

1. ADMINISTRADOR

TABLA XIV
ANÁLISIS DE REQUERIMIENTOS
RESUMEN DE REQUERIMIENTOS FUNCIONALES- ADMINISTRADOR

CÓDIGO	FUNCIÓN	CATEGORÍA
RQF1.1	El sistema debe Administrar los Pensum de las diferentes carreras del AEIRNNR	Evidente
RQF1.2	El sistema debe Administrar los contenidos de las diferentes unidades de cada módulo de las carreras del AEIRNNR	Evidente
RQF1.4	El sistema debe crear menús personalizados de acuerdo a los roles de los usuarios	Evidente
RQF1.5	El sistema debe Validar el correcto ingreso de la información de los Pensum de cada carrera	Oculto

2. SECRETARIA

TABLA XV
ANÁLISIS DE REQUERIMIENTOS
RESUMEN DE REQUERIMIENTOS FUNCIONALES- SECRETARIA

CÓDIGO	FUNCIÓN	CATEGORÍA
RQF2.1	El sistema debe visualizar el estado del Proceso de Homologación (Registro de estados) a través de internet	Evidente
RQF2.2	El sistema debe Imprimir el resultado de la Petición de Homologación	Evidente
RQF2.3	El sistema debe visualizar los porcentajes de la homologación de las carreras	Evidente

RQF2.4	El sistema debe llevar un registro de docentes con sus respectivas homologaciones asignadas	Evidente
RQF2.5	El sistema debe visualizar el resultado de la homologación vía internet	Evidente

3. DOCENTE

**TABLA XVI
ANÁLISIS DE REQUERIMIENTOS
RESUMEN DE REQUERIMIENTOS FUNCIONALES- DOCENTE**

CÓDIGO	FUNCIÓN	CATEGORÍA
RQF3.1	El sistema debe permitir la homologación entre carreras del AEIRNNR de la UNL	Evidente
RQF3.2	El sistema debe permitir la homologación de las carreras del AEIRNNR con carreras de otras universidades	Evidente
RQF3.3	El sistema debe permitir realizar la homologación mediante contenidos(temas)	Oculto
RQF3.4	El sistema debe permitir visualizar el estado del Proceso de Homologación (Registro de estados) a través de internet	Evidente
RQF3.5	El sistema debe permitir Imprimir el resultado de la Homologación	Evidente
RQF3.6	El sistema debe permitir al usuario(Docente) asignar contenidos(temas) de las unidades de cada carrera a homologar y almacenarlos	Evidente
RQF3.7	El sistema debe permitir visualizar los porcentajes de la homologación de las carreras	Evidente

RQF3.8	El sistema debe validar el correcto ingreso de la información de los Pensum de cada carrera	Oculto
RQF3.9	El sistema debe permitir visualizar el resultado de la homologación vía internet	Evidente
RQF3.10	El sistema debe permitir utilizar un mínimo de contenidos(temas) para realizar la Homologación	Oculto

4. ESTUDIANTE

**TABLA XVII
ANÁLISIS DE REQUERIMIENTOS
RESUMEN DE REQUERIMIENTOS FUNCIONALES- ESTUDIANTE**

CÓDIGO	FUNCIÓN	CATEGORÍA
RQF4.1	El sistema debe permitir que el estudiante ingrese una solicitud de homologación vía internet	Evidente
RQF4.2	El sistema debe permitir visualizar el estado del Proceso de Homologación (Registro de estados) a través de internet	Evidente

RESUMEN DE REQUERIMIENTOS NO FUNCIONALES

TABLA XVIII
ANÁLISIS DE REQUERIMIENTOS
RESUMEN DE REQUERIMIENTOS NO FUNCIONALES

CÓDIGO	REQUERIMIENTO NO FUNCIONAL
RQNF.1	El sistema debe ser aplicable a cualquier carrera
RQNF.2	El sistema debe ser multiusuario
RQNF.3	Restringir el ingreso de datos erróneos que no corresponden al tipo definido.
RQNF.4	Mostrar información de ayuda para el usuario.
RQNF.5	Mostrar etiquetas claras para campos, botones, barras, menús.
RQNF.6	El sistema debe permitir el guardado automático del resultado obtenido en la comparación de las dos mallas curriculares.
RQNF.7	El sistema debe permitir trabajar a través de teclado y mouse
RQNF.8	La interfaz de Usuario debe ser amigable con el usuario. Las ventanas deben estar orientadas a una navegación rápida y comprensible.
RQNF.9	El tiempo de respuesta del sistema dependerá del equipo en que funcione el sistema.

GLOSARIO DE TÉRMINOS

**TABLA XIX
GLOSARIO DE TÉRMINOS**

TÉRMINO	CATEGORÍA	COMENTARIO
Universidad	Clase	Entidad Educativa que brinda enseñanza a la juventud
Área	Clase	Unidad Organizativa de una Universidad; corresponde a un gran campo científico o profesional y en ella se imparten las enseñanzas de una o varias carreras. La máxima autoridad de un área es el Director.
Carrera	Clase	Ente organizativo de un área que dirige la formación de una profesión.
Nivel	Clase	Hace referencia a los diferentes niveles de educación que se encuentran en una carrera. Estos pueden ser: Pregrado, Postgrado, Maestría, Artesanal. Etc.
Malla Curricular	Clase	Hace referencia a las mallas curriculares tanto del A.E.I.R.N.N de la UNL como de otras Universidades.
Persona	Clase	Individuo de la especie humana, hombre o mujer
Usuario	Clase	Persona que tiene acceso al sistema
Malla Curricular	Clase	Conjunto de materias/unidades que se deben dictar en un determinado nivel
Rol	Clase	Acciones que puede realizar un determinado usuario
Materia/Unidad	Clase	Conocimientos relacionados a un tópico específico bajo una sola denominación para determinado curso
Petición Homologación	Clase	Solicitud de Homologación que la puede realizar un usuario del sistema con el rol de estudiante o secretaria

TÉRMINO	CATEGORÍA	COMENTARIO
Registro Homologación	Clase	Registro de las unidades o materias homologadas
Contenido	Clase	Etiquetas empleadas en cada una de las unidades para el proceso de homologación
Unidad de Tiempo	Clase	Hace referencia al tiempo en estudio, esto puede ser: ciclo, modulo, semestre, año, etc.
Administrar Homologación	Caso de Uso	Descripción del Proceso en el cual el administrador crea, modifica o elimina la malla curricular origen; ejecuta el proceso (algoritmo) de homologación para posteriormente presentar los resultados de la solicitud de homologación seleccionada.
Editar Homologación	Caso de Uso	Descripción del proceso en el cual se edita específicamente los datos de la carrera origen.
Administrar Malla Curricular	Caso de Uso	Descripción del Proceso a través del cual el administrador pueda crear, modificar o eliminar las mallas curriculares de la carrera destino.
Editar Malla Curricular	Caso de Uso	Descripción del proceso en el cual el administrador creará una malla curricular de una determinada carrera destino
Editar Unidad o Materia	Caso de Uso	Descripción del Proceso en el cual el administrador creará unidades o materias de para asignarlas a una malla curricular en una determinada carrera destino.
Autenticación o Autenticación		Es la confirmación de algo o alguien como auténtico. Este término se refiere al acto de verificar que la persona que se conecta a una aplicación es la que se espera, y debe tener los derechos, permisos y/o privilegios que le corresponden.
PHP		Es un acrónimo de <i>Hypertext Pre-processor</i> , es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. PFC: Pedro Alfaro Fernández Página 7

TÉRMINO	CATEGORÍA	COMENTARIO
HTML		Siglas de HyperText Markup Language (<i>Lenguaje de Marcado de Hipertexto</i>), es el lenguaje de marcado (junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación) predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.
DB		Acrónimo de Data Base.
Interfaz		Se refiere a una conexión física y funcional entre dos aparatos o sistemas independientes, en el caso de Interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, en nuestro caso nuestra aplicación software.
Symfony		Es un completo framework para aplicaciones web de proyectos PHP.
ORM		Del inglés Object-Relational-Mapping, mapeo objeto-relacional en castellano, es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.
Doctrine		Es un ORM para PHP que ofrece persistencia para objetos PHP.
Apache		Es un servidor muy popular.
Twig		Engine de plantillas para PHP que incluye symfony.
MVC		O modelo vista controlador es un modelo de abstracción en el desarrollo del software.

DIAGRAMA DE CASOS DE USO

Diagrama de Caso de Uso de Uso Peticionario (Estudiante)

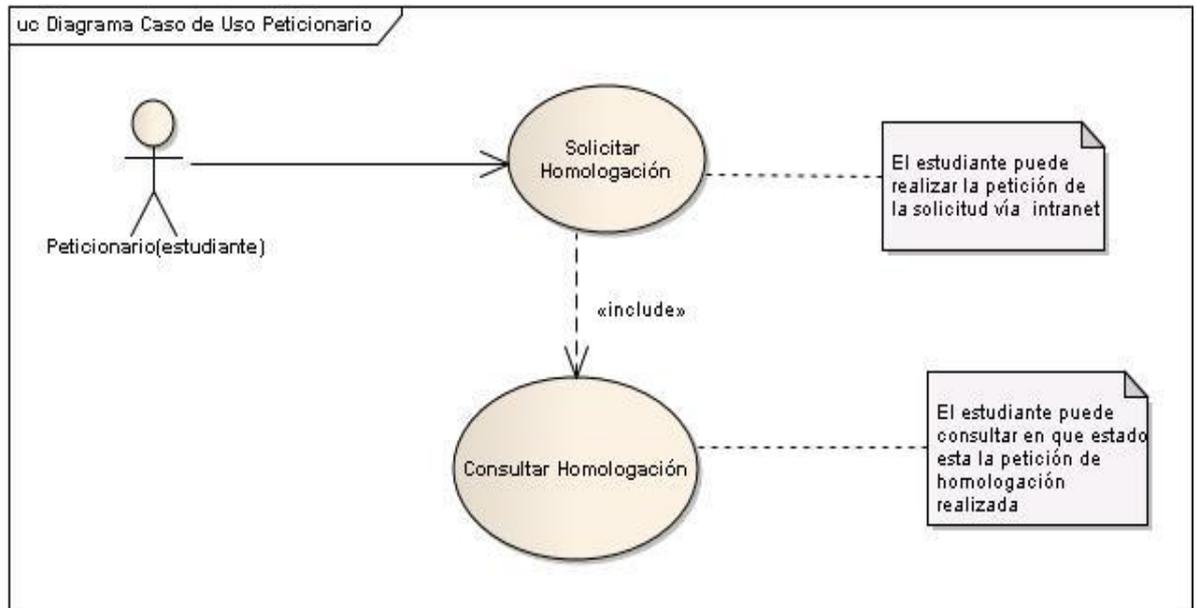


Figura 15. Diagrama de caso de uso del peticionario

Diagrama de Caso de Uso de Uso Administrador de Homologaciones (Secretario)

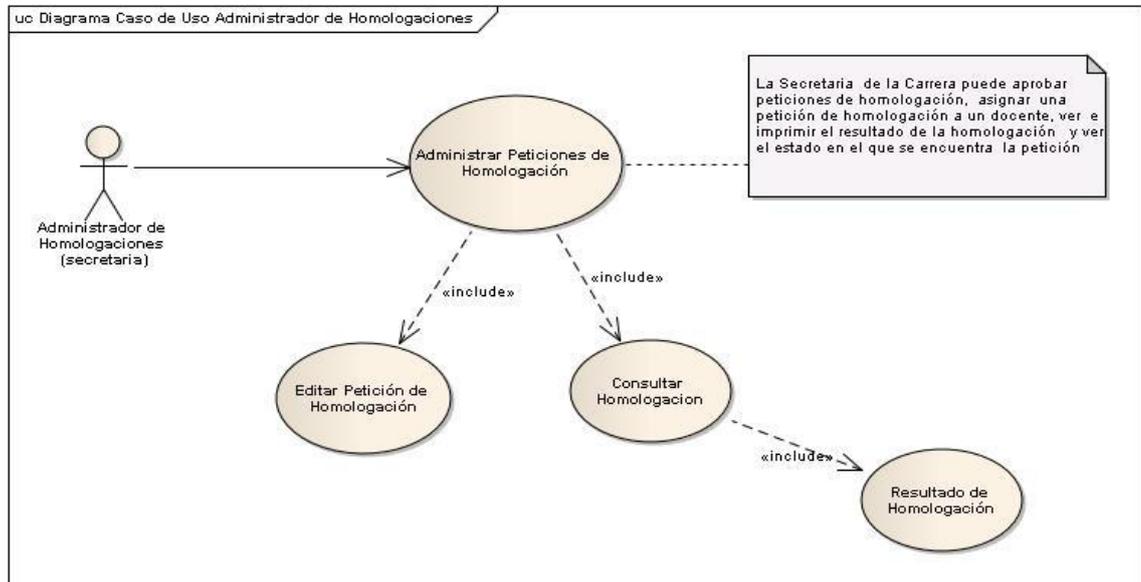


Figura 16. Diagrama de caso de uso del administrador de homologaciones

Diagrama de Caso de Uso de Uso Homologador (Docente)

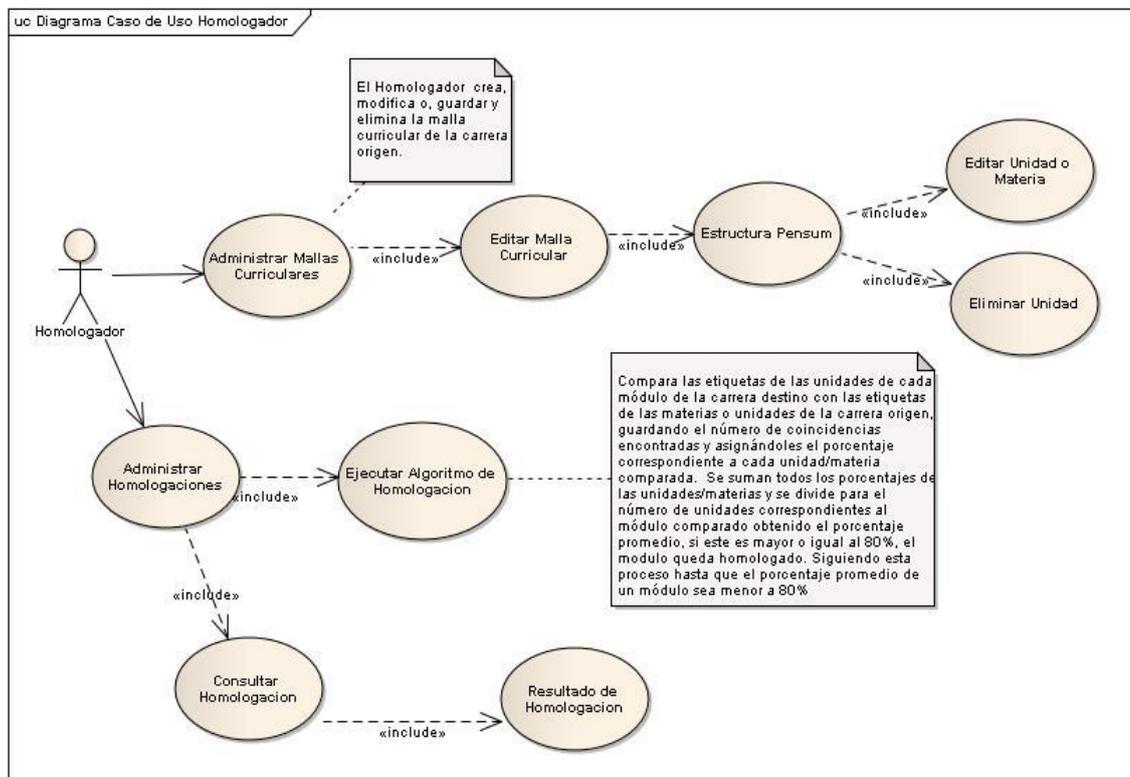


Figura 17. Diagrama de caso de uso del docente

Diagrama de Casos de Uso Administrador

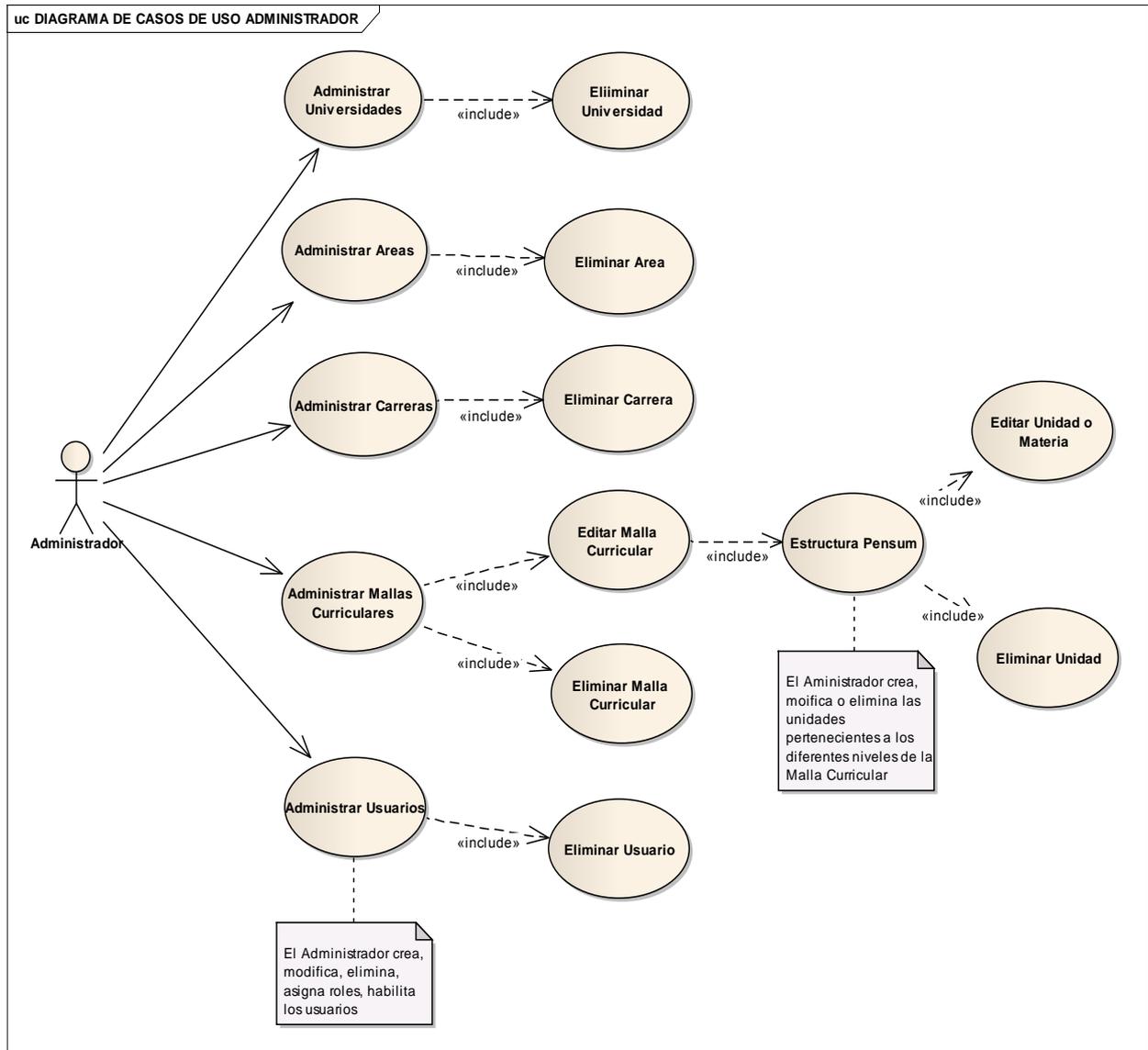


Figura 18. Diagrama de caso de uso del administrador

ANÁLISIS Y DISEÑO PRELIMINAR

En esta etapa de la metodología ICONIX, se realiza la descripción de los Casos de Uso con sus correspondientes flujos principales y alternos. La presentación de los Casos de Uso se la realiza con su respectiva Pantalla del sistema, Caso de Uso y Diagrama de Secuencia.

CASO DE USO: Ingresar Petición de Homologación

The screenshot displays the 'Homologa' web application interface. At the top, there is a green header with the application logo and the name 'Homologa'. On the right side of the header, it shows the user's name 'Usuario: maria' and a 'Cerrar sesión' link. Below the header, there is a navigation menu with links for 'Inicio', 'Homologar', and 'Mi perfil'. The main content area features a search form titled 'Buscar Peticiones Homologación' with input fields for 'Nombre' and 'Cedula', and a 'Buscar' button. Below the search form is a table titled 'Peticiones de Homologación' with a 'Nueva Petición Homologación' button. The table contains the following data:

No.	Nombre Apellido	Estado	Fecha	Acciones
1	María del Cisne Mera	Asignado	2012-12-23	Consultar
10	María del Cisne Mera	Homologado	2013-02-21	Consultar
28	María del Cisne Mera	Solicitado	2013-01-11	Consultar
37	María del Cisne Mera	Solicitado	2013-01-15	Consultar
38	María del Cisne Mera	Solicitado	2013-01-15	Consultar
39	María del Cisne Mera	Solicitado	2013-01-15	Consultar
40	María del Cisne Mera	Solicitado	2013-01-15	Consultar
41	María del Cisne Mera	Solicitado	2013-01-17	Consultar

Figura 19. Pantalla ingresar petición de homologación

TABLA XX
CASO DE USO: INGRESAR PETICIÓN DE HOMOLOGACIÓN

Caso de Uso:	Ingresar Petición de Homologación
Precondición:	- El Peticionario debe haber ingresado al sistema
Post Condición:	- La información de la petición de Homologación queda modificada de alguna manera
Referencia a Requerimientos:	RQ7, RQ8, RQ9, RQ10.
CURSO BÁSICO	
1. El Peticionario elige la Opción <Homologar> en la pantalla “Inicio”	
2. El sistema presenta la lista de peticiones de homologación realizadas, en la pantalla “Peticiones de Homologación”	
3. El Peticionario elige la opción <Nueva Petición de Homologación> en la pantalla “Peticiones de Homologación”	
4. El sistema recupera los datos de la persona peticionaria y crea una petición; y los presenta en la pantalla “Edición de Petición de Homologación”	
5. El sistema llama al fragmento “ Editar Petición de Homologación ”	
6. El caso de uso finaliza.	
CURSO ALTERNO	
3A. EL PETICIONARIO DESEA CONSULTAR EL ESTADO DE LA HOMOLOGACIÓN.	
1. El sistema llama al fragmento “Consultar Petición de Homologación”	
2. El caso de uso finaliza	

DIAGRAMA DE SECUENCIA

Caso de Uso: Ingresar Petición de Homologación

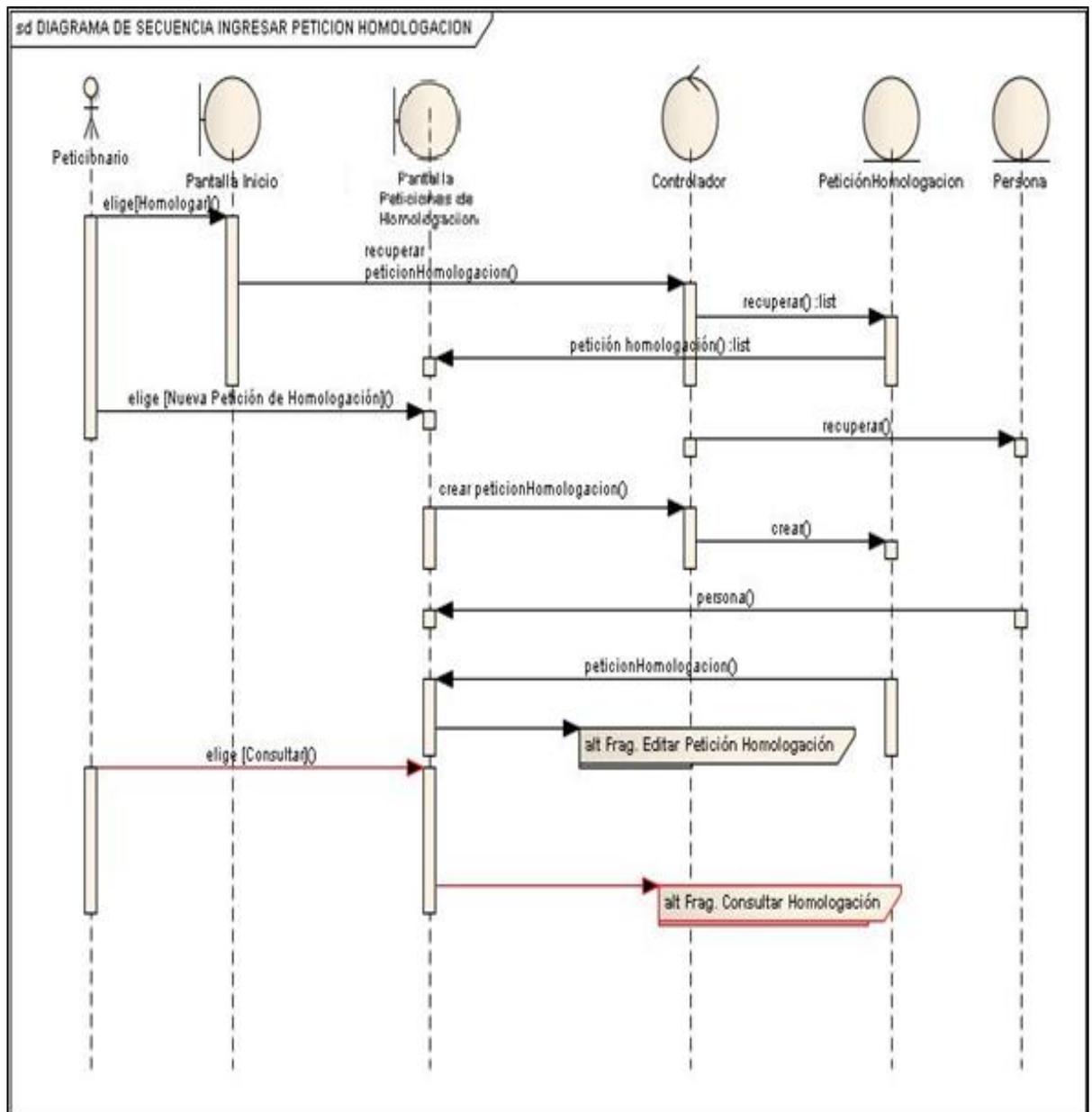


Figura 20. Diagrama de secuencia ingresar petición de homologación

PANTALLA EDICIÓN DE PETICIÓN DE HOMOLOGACIÓN

Figura 21. Pantalla edición de petición de homologación

FRAGMENTO: Editar Petición de Homologación

TABLA XXI
FRAGMENTO: EDITAR PETICIÓN DE HOMOLOGACIÓN

Fragmento:	Editar Petición de Homologación
Caso de Uso Referido:	Ingresar Petición De Homologación
Curso Básico	
1. El Peticionario escoge la universidad origen de la lista desplegable en la pantalla “Edición de Petición Homologación”	
2. El sistema recupera y carga las carreras correspondientes a la universidad seleccionada, en la pantalla “Edición de Petición de Homologación”.	
3. El Peticionario escoge la carrera origen de la lista desplegable en la pantalla “Edición de Petición Homologación”	
4. El Peticionario escoge la carrera destino de la lista desplegable y elige la	

opción <Guardar> en la pantalla “Edición de Petición de Homologación”
5. El sistema valida que la petición Ingresada tenga datos en los campos identificación, nombres, apellidos, teléfono y email; y que la universidad y carrera origen tengan escogido un valor
6. El sistema guarda la petición creada y cambia el estado de la petición a “Solicitado”
7. El caso de uso finaliza
Cursos Alternos
2A. LA UNIVERSIDAD DE ORIGEN NO ESTA DISPONIBLE EN LA LISTA DESPLEGABLE
<ol style="list-style-type: none"> 1. El Peticionario escoge la opción Otra de la lista desplegable Universidad Origen en la pantalla “Edición de Petición Homologación” 2. El sistema habilita el campo Universidad Origen y carga las opciones de Carrera Origen en la pantalla “Edición de Petición Homologación” 3. El Peticionario de Homologación ingresa el nombre de la Universidad Origen y escoge la opción Otra de la lista desplegable de Carrera Origen, en la pantalla “Edición de Petición Homologación” 4. El sistema habilita el campo Carrera Origen en la pantalla “Edición de Petición Homologación” 5. El peticionario ingresa el nombre de la Carrera Origen en la pantalla “Edición de Petición Homologación”. 6. Continúa con el paso 4 del Curso Básico
5A. EL SISTEMA PRESENTA ERROR AL VALIDAR LOS DATOS INGRESADOS
<ol style="list-style-type: none"> 1. El sistema carga y presenta la pantalla “Edición de Petición de Homologación” con los campos con error con asterisco(*) y de color rojo 2. Continúa con el paso 1 del Curso Básico.

DIAGRAMA DE SECUENCIA

Fragmento: Editar Petición de Homologación

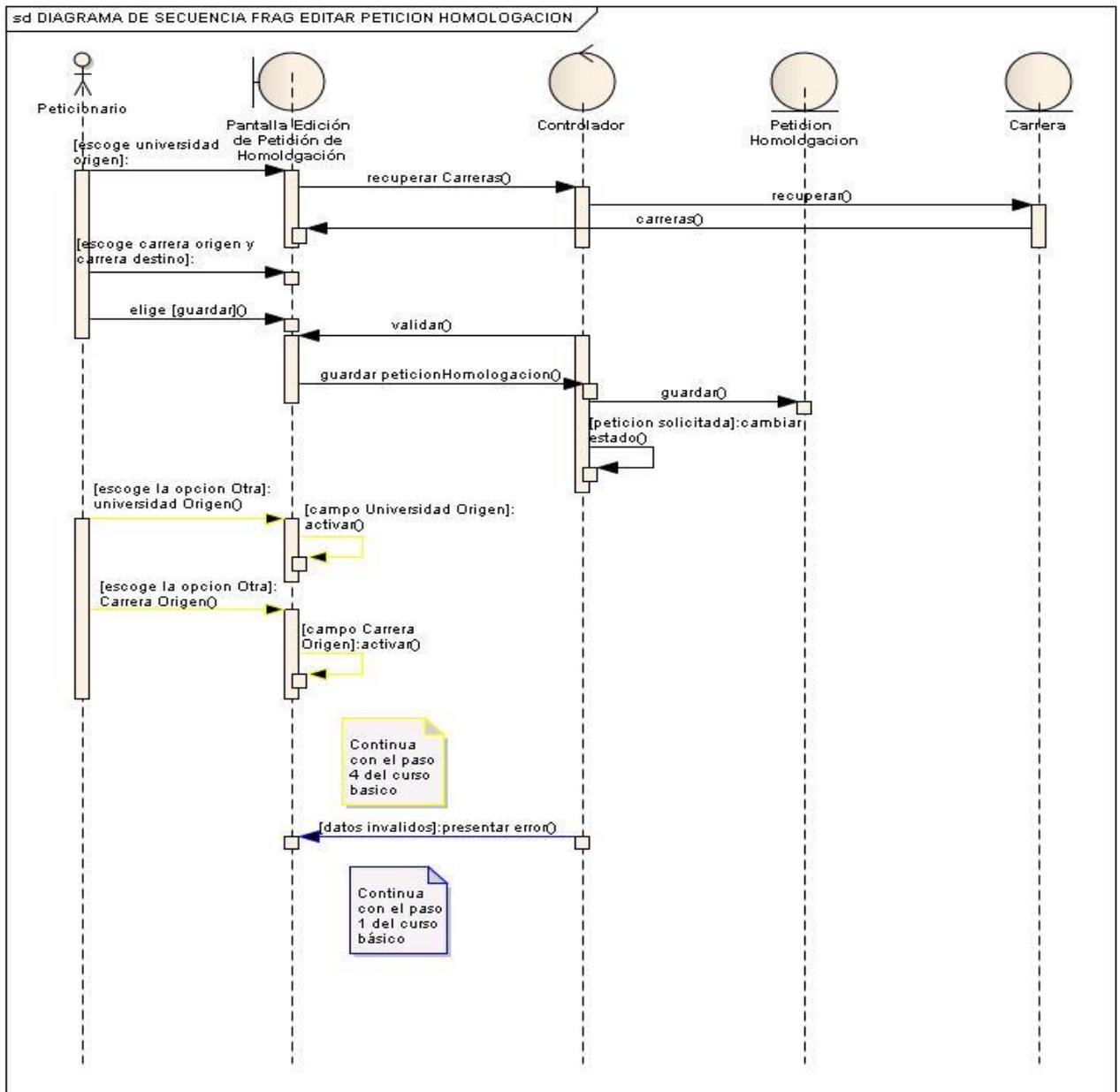


Figura 22. Diagrama de secuencia editar petición de homologación.

PANTALLA CONSULTAR PETICIÓN DE HOMOLOGACIÓN

Homologa

Usuario: secre
Cerrar sesión

Inicio Homologar Mi perfil

Petición de Homologación

eduardo reyes Numero de Oficio

Fecha	Descripción	Observaciones
2013-01-06	Solicitado	
2013-01-06	Asignado	
2013-01-08	Homologado	
2013-01-19	Asignado	

Inicio Ver Resultado Imprimir

Universidad Nacional de La Plata
Martes 19 de 2013

Figura 23. Pantalla consultar petición de homologación para el secretario o docente

Homologa

Usuario: maria
Cerrar sesión

Inicio Homologar Mi perfil

Petición de Homologación

María del Cisne Mera Numero de Oficio 1

Fecha	Descripción	Observaciones
2012-12-18	Solicitado	
2012-12-19	Asignado	
2012-12-23	Asignado	
2012-12-23	Asignado	

Inicio

Universidad Nacional de La Plata
Martes 19 de 2013

Figura 24. Pantalla consultar petición de homologación para el estudiante

FRAGMENTO: Consultar Petición de Homologación

TABLA XXII
FRAGMENTO: CONSULTAR PETICIÓN DE HOMOLOGACIÓN

Fragmento:	Consultar Petición de Homologación
Caso de Uso Referido:	Petición de Homologación
Curso Básico	
1.	El peticionario elige la opción <Consultar> en la pantalla “Petición de Homologación”
2.	El sistema recupera la Petición Homologación seleccionada con los estados por los que ha pasado esta petición
3.	El sistema valida que el estado de la petición de homologación sea diferente a homologado
4.	El sistema presenta la petición de homologación con los estados por los cuales ha pasado, con el botón <Inicio> activado en la pantalla “Consultar”.
5.	El Peticionario revisa los datos presentados en la pantalla “Consultar”
6.	El caso de uso finaliza
CURSOS ALTERNOS	
2A.	El sistema valida que el estado de la petición es holomogada
1.	El sistema valida que el estado de la Petición es “Homologada”
2.	El Sistema activa los botones <Ver Resultado> e <Imprimir> en la pantalla “Consultar”
3.	El Peticionario revisa los datos presentados y escoge la opción <Ver Resultado> en la pantalla “Consultar”
4.	El sistema llama al fragmento “Resultado de Homologación”
5.	Continúa con el paso 6 de curso básico

DIAGRAMA DE SECUENCIA

Fragmento: Consultar Petición de Homologación

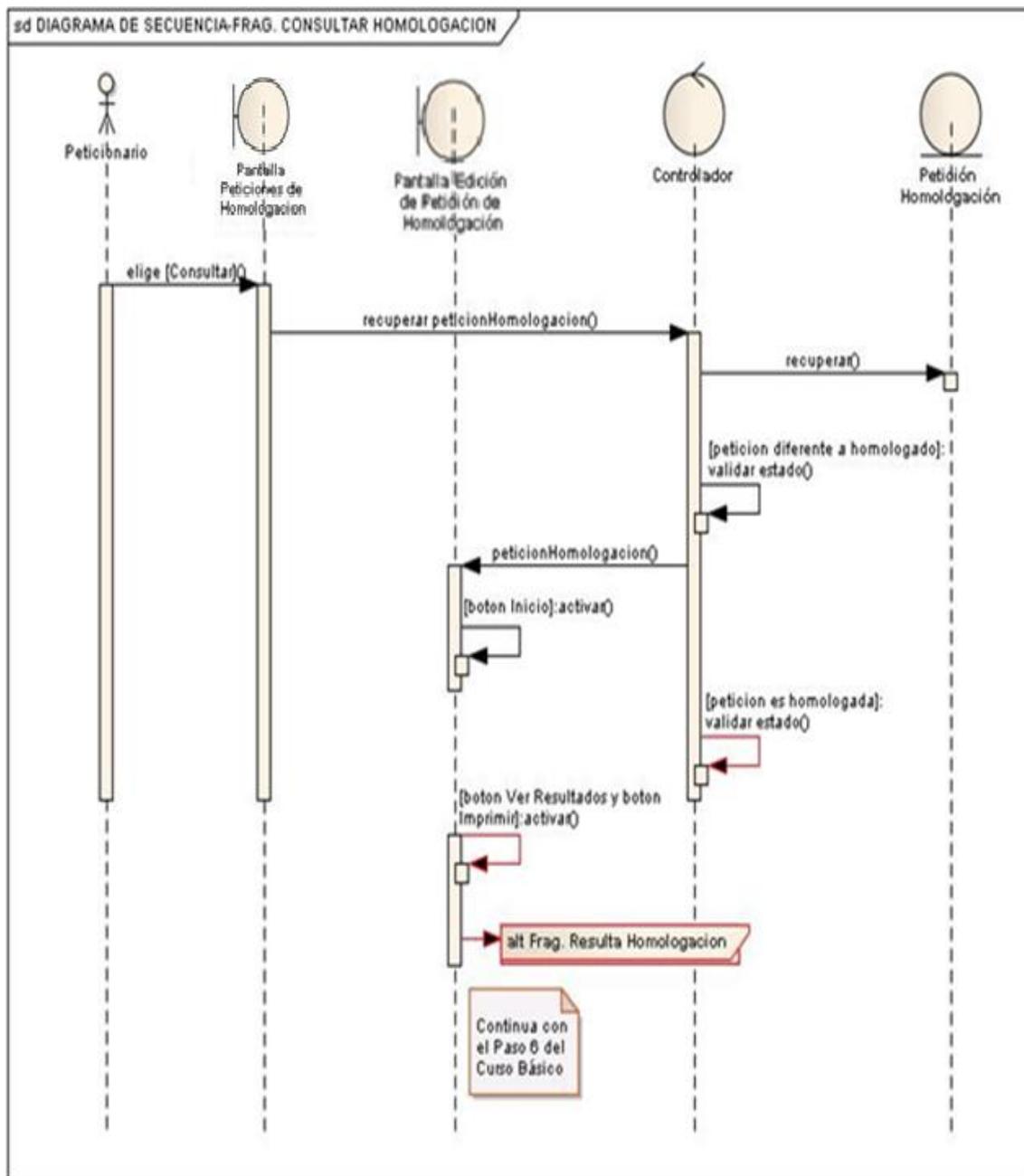


Figura 25. Diagrama de secuencia consultar petición de homologación

PANTALLA PETICIONES DE HOMOLOGACIÓN

Homologa

Usuario: secre
Cerrar sesión

Inicio Homologar MI perfil

Buscar Peticiones Homologación

Nombre
Cedula
Buscar

Peticiones de Homologación Nueva Peticion Homologacion

No.	Nombre Apellido	Estado	Fecha	Acciones
11	eduardo reyes	Asignado	2013-01-19	Asignar Editar Archivar Consultar
53	fred reyes	Homologado	2013-03-07	Asignar Editar Archivar Consultar
47	Jorge Eduardo Armijos Valverde	Asignado	2013-02-24	Asignar Editar Archivar Consultar
8	Juan Vasquez	Homologado	2013-01-09	Asignar Editar Archivar Consultar
29	karina valarezo	Asignado	2013-01-22	Asignar Editar Archivar Consultar
30	karina valarezo	Homologado	2013-03-05	Asignar Editar Archivar Consultar
33	karina valarezo	Asignado	2013-01-22	Asignar Editar Archivar Consultar
50	leo leo	Asignado	2013-02-28	Asignar Editar Archivar Consultar

Figura 26. Pantalla peticiones de homologación para el secretario

PANTALLA EDITAR PETICIÓN DE HOMOLOGACIÓN

Homologa

Usuario: lola
Cerrar sesión

Inicio Homologar MI perfil

Petición de Homologación

Cedula
Nombre
Apellido
Telefono
Email
Universidad
Origen
Carrera
Origen
Carrera
Destino
Guardar Cancelar

Universidad Nacional de Loja
Homologa 1.0.0 2013

Figura 27. Pantalla editar petición de homologación.

CASO DE USO: Administrar Peticiones de Homologación

TABLA XXIII
CASO DE USO: ADMINISTRAR PETICIONES DE HOMOLOGACIÓN

Caso de Uso:	Administrar Peticiones de Homologación
Precondición:	El Administrador de Homologaciones debe haber ingresado al sistema
Post Condición:	La petición de homologación queda modificada de alguna manera
Referencia a Requerimientos:	RQ 7, RQ.8, RQ 9, RQ 10, RQ 15, RQ 17
CURSO BÁSICO	
1.	El Administrador de Homologaciones elige la opción < Homologar > en la pantalla "Inicio"
2.	El sistema recupera las peticiones de homologación existentes y las presenta en la pantalla "Peticiones de Homologación"
3.	El Administrador de Homologaciones elige la opción <Nueva Petición Homologación> en la pantalla "Peticiones de Homologación"
4.	El sistema crea un usuario y una petición y presenta la pantalla "Peticiones de Homologación"
5.	El Administrador de Homologaciones ingresa los datos del peticionario: Identificación, Nombre, Apellido, Teléfono, email, Nombre de Usuario y Clave en la pantalla "Ingresar Petición Homologación"
6.	El sistema llama al fragmento " Editar Petición de Homologación "
7.	El caso de uso finaliza
CURSOS ALTERNOS	
3A.	EL ADMINISTRADOR DE HOMOLOGACIONES DESEA EDITAR UNA PETICIÓN DE HOMOLOGACIÓN EXISTENTE

1. EL Administrador de Homologaciones selecciona una petición y elige la opción <Editar> en la pantalla “Petición de Homologación”
2. El sistema recupera la petición y la presenta en la pantalla “Edición de Petición de Homologación”
3. Continúa con el paso 6 del curso básico

3B. EL ADMINISTRADOR DE HOMOLOGACIONES DESEA ASIGNAR UNA HOMOLOGACIÓN

1. El Administrador de Homologaciones selecciona una petición y elige la opción <Asignar> en la pantalla “Petición de Homologación”
2. El sistema recupera la petición seleccionada y la presenta en la pantalla “Asignar Petición Homologación” con los campos de los datos de la Petición desactivados
3. El Administrador de Homologaciones escoge el nombre del profesor, ingresa el No. de oficio y elige la opción <Guardar> en la pantalla “Asignar Petición Homologación”
4. El sistema valida que la petición tenga datos en el campo No. de oficio y que el profesor tenga escogido un valor.
5. El sistema guarda la petición, envía una notificación al profesor seleccionado, cambia el estado de la solicitud a “Asignada” y presenta la lista de peticiones de homologación en la pantalla “Ingresar Petición de Homologación”
6. Continúa con el paso 7 del curso básico

3C. EL ADMINISTRADOR DE HOMOLOGACIONES DESEA CONSULTAR EL ESTADO O EL RESULTADO DE LA HOMOLOGACIÓN

1. EL Administrador de Homologaciones selecciona una petición y elige la opción <Consultar> en la pantalla “Petición de Homologación”
2. El sistema llama al fragmento “**Consultar Petición de Homologación**”
3. Continúa con el paso 7 de curso básico

3E. EL ADMINISTRADOR DE HOMOLOGACIONES DESEA BUSCAR UNA PETICIÓN

1. El Administrador de Homologaciones ingresa alguno de los campos requeridos para la búsqueda como son (cédula, nombre o apellido) y elige la opción

<Buscar> en la pantalla “ Ingresar Petición de Homologación”

2. El sistema busca la o las peticiones y la presenta en la pantalla “Peticiones de Homologación”
3. Continúa con el paso 7 del curso básico

DIAGRAMA DE SECUENCIA

Caso de Uso: Administrar Peticiones de Homologación

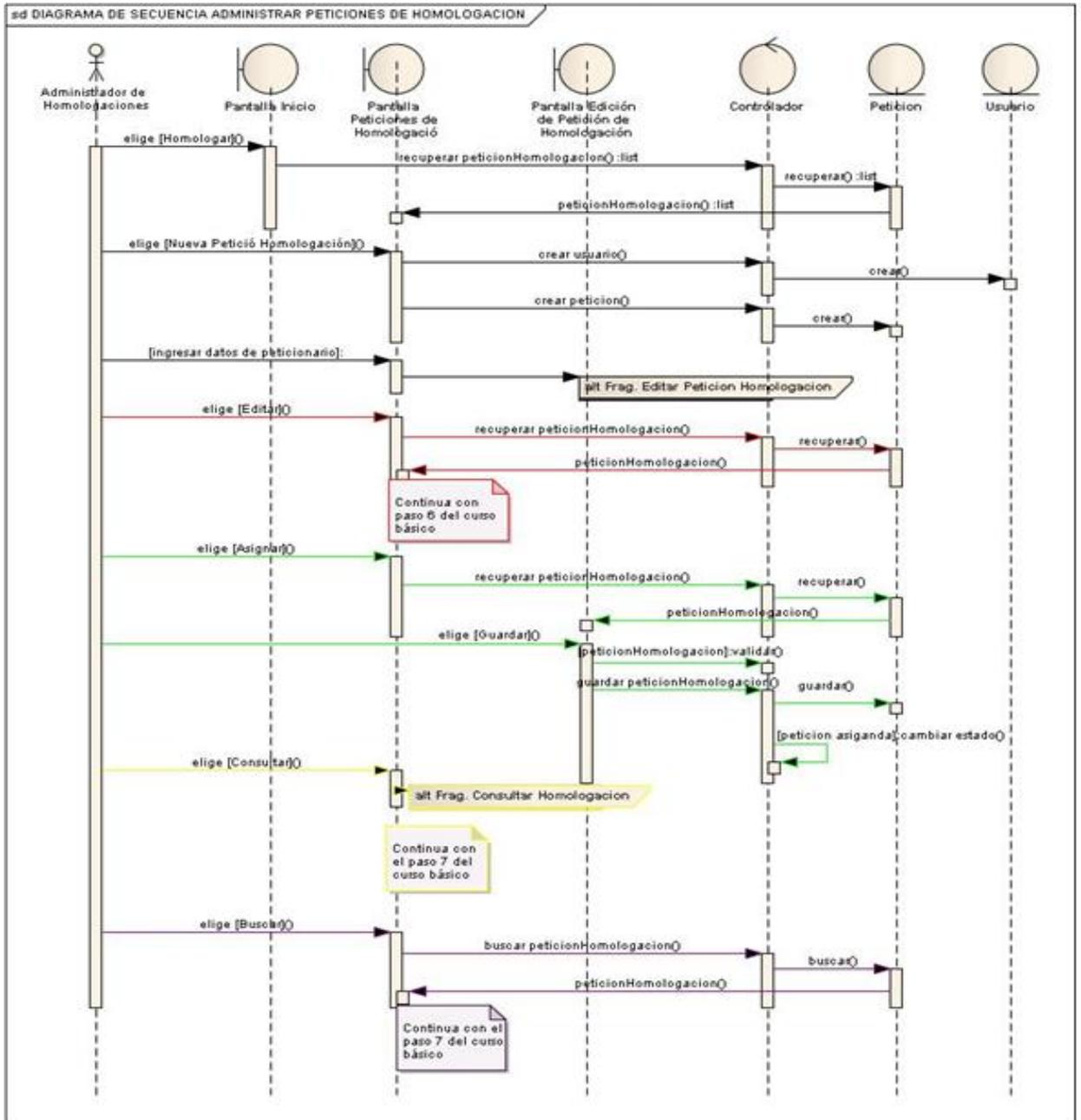


Figura 28. Diagrama de secuencia administrar peticiones de homologación

PANTALLA PETICIONES DE HOMOLOGACIÓN

The screenshot shows the 'Homologa' web application interface. At the top, there is a green header with the 'Homologa' logo and the text 'Usuario: edison' and 'Cerrar sesión'. Below the header is a navigation bar with links for 'Inicio', 'Homologar', and 'Mi perfil'. The main content area is titled 'Buscar Peticiones Homologación' and contains a search form with two input fields: 'Nombre' and 'Cedula', and a 'Buscar' button. Below the search form is a table titled 'Peticiones de Homologación' with the following data:

No.	Nombre Apellido	Estado	Fecha	Acciones
11	eduardo reyes	Asignado	2013-01-19	Consultar Visualizar
53	fred reyes	Homologado	2013-03-07	Consultar Visualizar
8	Juan Vasquez	Homologado	2013-01-09	Consultar Visualizar
50	leo leo	Asignado	2013-02-28	Consultar Homologar
42	Lorena Elizabeth Ordoñez Cartuche	Asignado	2013-01-18	Consultar Homologar
44	Luis Lozano	Homologado	2013-02-21	Consultar Visualizar

Figura 29. Pantalla peticiones de homologación para docente

PANTALLA HOMOLOGAR PETICIÓN DE HOMOLOGACIÓN

Usuario: milton
Cerrar sesión

[Inicio](#) | [Homologar](#) | [Mi perfil](#)

Petición de Homologación

Numero de
Oficio
Cedula
Nombre
Apellido
Carrera
Origen
Malla Origen
Carrera
Destino
Malla
Destino
Tipo de
Escala
Escala
Minima
Escala
Máxima
Numero de
Modulo

Nivel	Materia	Horas	Creditos	Calificacion	Contenidos	Observaciones
<input type="text" value="1"/>	<input type="text" value="Matematicas 1"/>	<input type="text" value="30"/>	<input type="text" value="6"/>	<input type="text" value="10"/>	<input type="text" value="operaciones t"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Fisica Basica"/>	<input type="text" value="30"/>	<input type="text" value="7"/>	<input type="text" value="10"/>	<input type="text" value="poleas, movin"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Geometria"/>	<input type="text" value="20"/>	<input type="text" value="4"/>	<input type="text" value="8"/>	<input type="text" value="planos, forma"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Bases Quimicas"/>	<input type="text" value="30"/>	<input type="text" value="2"/>	<input type="text" value="9.46"/>	<input type="text" value="compuestos c"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Investigacion"/>	<input type="text" value="25"/>	<input type="text" value="2"/>	<input type="text" value="7"/>	<input type="text" value="trabajos de ca"/>	<input type="text"/>
<input type="text" value="2"/>	<input type="text" value="Matematicas 2"/>	<input type="text" value="40"/>	<input type="text" value="6"/>	<input type="text" value="8"/>	<input type="text" value="matrices"/>	<input type="text"/>
<input type="text" value="2"/>	<input type="text" value="Fisica Avanzada"/>	<input type="text" value="40"/>	<input type="text" value="7"/>	<input type="text" value="8"/>	<input type="text" value="vectores, lent"/>	<input type="text"/>
<input type="text" value="2"/>	<input type="text" value="Algebra Lineal"/>	<input type="text" value="28"/>	<input type="text" value="4"/>	<input type="text" value="8"/>	<input type="text" value="sistemas de e"/>	<input type="text"/>
<input type="text" value="2"/>	<input type="text" value="Geometria"/>	<input type="text" value="30"/>	<input type="text" value="6"/>	<input type="text" value="9"/>	<input type="text" value="angulos, circul"/>	<input type="text"/>
<input type="text" value="2"/>	<input type="text" value="Investigacion 2"/>	<input type="text" value="40"/>	<input type="text" value="7"/>	<input type="text" value="7"/>	<input type="text" value="problematiza"/>	<input type="text"/>
<input type="text" value="3"/>	<input type="text" value="Electronica Aplicada"/>	<input type="text" value="40"/>	<input type="text" value="7"/>	<input type="text" value="8"/>	<input type="text" value="amplificacion"/>	<input type="text"/>
<input type="text" value="3"/>	<input type="text" value="Informatica 1"/>	<input type="text" value="20"/>	<input type="text" value="4"/>	<input type="text" value="8"/>	<input type="text" value="computadore"/>	<input type="text"/>
<input type="text" value="3"/>	<input type="text" value="Legislacion"/>	<input type="text" value="20"/>	<input type="text" value="2"/>	<input type="text" value="8"/>	<input type="text" value="xxxxx"/>	<input type="text"/>

Universidad Nacional de Loja
Homologa 1.0.0.2011

Figura 30. Pantalla homologar petición de homologación

CASO DE USO: Administrar Homologaciones

TABLA XXIV
CASO DE USO: ADMINISTRAR HOMOLOGACIONES

Caso de Uso:	Administrar homologaciones
Precondición:	El homologador debe haber ingresado al sistema y justificado su rol
Post Condición:	La información de las mallas curriculares queda modificada de alguna manera. Quedan homologados dos mallas curriculares
Referencia a Requerimientos:	RQ 1, RQ 2, RQ 5, RQ 10, RQ 13, RQ 18
Curso Básico	
1. El Homologador elige la opción < Homologar> en la pantalla “Inicio”	
2. El sistema recupera y presenta la lista de peticiones de homologación asignadas, en la pantalla “Peticiones de Homologación”	
3. El Homologador selecciona la petición de homologación y elige la opción <Homologar> en la pantalla “Peticiones de Homologación”	
4. El sistema valida que el estado de la petición de homologación sea diferente ha homologado.	
5. El sistema recupera la petición de homologación con los datos del peticionario desactivados, recupera las mallas de la carrera origen y las mallas de la carrera destino, y las presenta en la pantalla “Homologar Petición de Homologación”.	
6. El Homologador selecciona la malla de la carrera origen en la pantalla “Homologar Petición de Homologación”	
7. El sistema recupera las unidades o materias de la malla seleccionada y las presenta en la pantalla “Homologar Petición de Homologación”.	
8. El Homologador selecciona: la malla de la carrera destino, el tipo de escala de calificación (numérico o alfanumérico), ingresa la escala mínima y escala máxima de calificación, y el número de módulo de la carrera destino máximo a homologar, en la pantalla “Homologar Petición de Homologación”.	
9. El Homologador, ingresa las calificaciones, contenidos adicionales y opcionalmente observaciones de cada materia de la malla origen y elige la opción <Guardar> en la pantalla “Homologar Petición de Homologación”.	
10. El sistema valida los campos ingresados	
11. El sistema guarda la Homologación	
12. Finaliza Caso de Uso	
Cursos Alternos	

<p>3A. El Homologador desea consultar la petición de Homologación.</p> <ol style="list-style-type: none">1. El Homologador selecciona la petición de homologación y elige la opción <Consultar> en la pantalla “Peticiones de Homologación”.2. El sistema llama al fragmento “Consultar Petición de Homologación”3. Continúa con el paso 12 del Curso Básico
<p>4A. El sistema valida que el estado de la petición sea homologada.</p> <ol style="list-style-type: none">1. El sistema valida que el estado de la Petición es “Homologada”2. El sistema llama al fragmento “Resultado de Homologación”3. Continúa con el paso 12 de curso básico
<p>9A. El Homologador desea realizar la homologación de la petición seleccionada.</p> <ol style="list-style-type: none">1. El Homologador elige la opción <Homologar> en la pantalla “Homologar Petición de Homologación “2. El sistema llama al fragmento “Resultado de Homologación”3. Continúa con el paso 12 de curso básico

DIAGRAMA DE SECUENCIA

Caso de Uso: Administrar Homologaciones

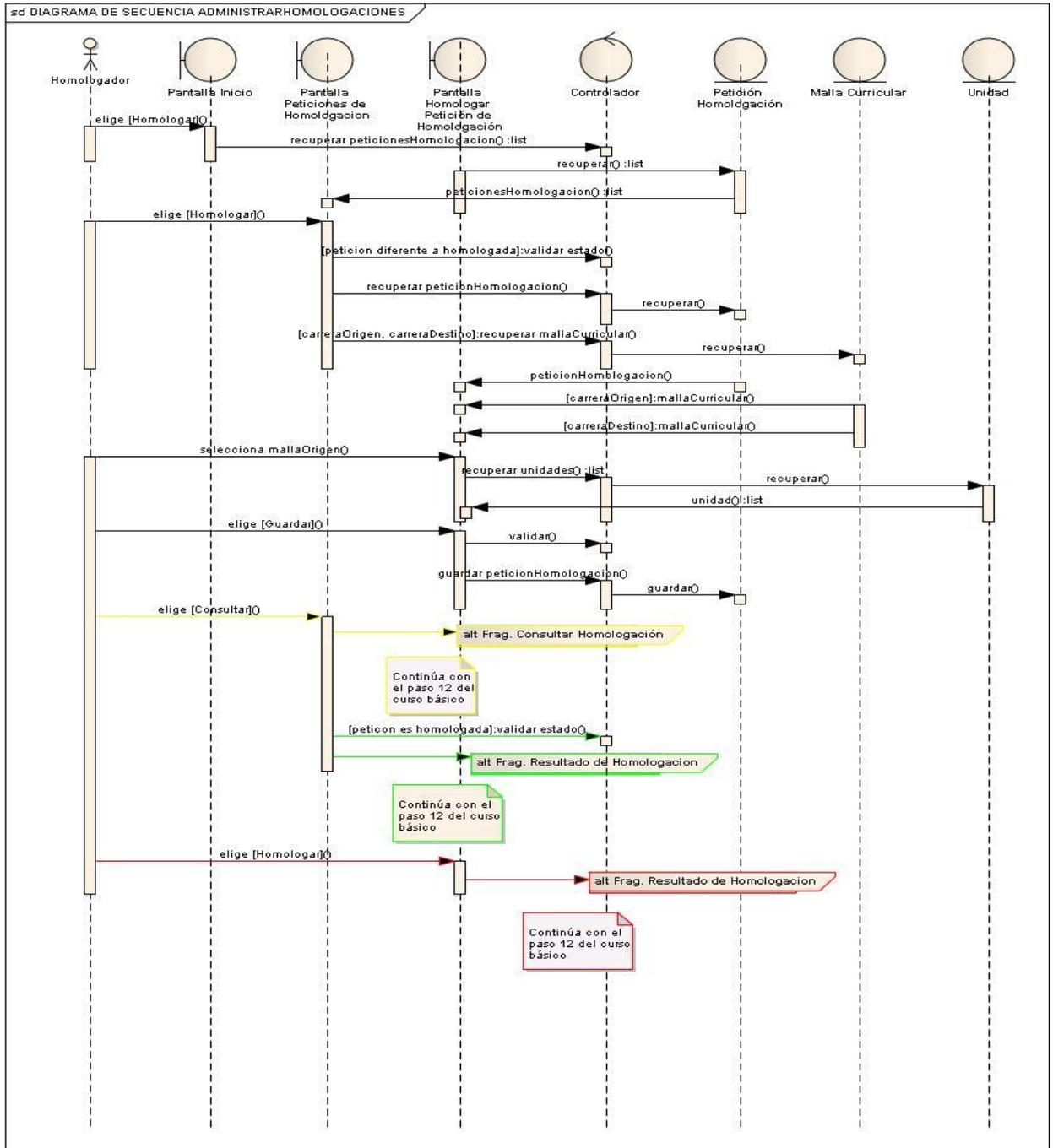


Figura 31. Diagrama de secuencia homologaciones

PANTALLA RESULTADO DE HOMOLOGACIÓN


Homologa
Usuario: edison
Cerrar sesión

[Inicio](#) | [Homologar](#) | [Mi perfil](#)
?

Resultado Petición de Homologación

Cedula
Nombre
Apellido
Telefono
Universidad
Origen
Carrera
Origen
Mallaorigen

Resumen la señorita estudiante segun el res

Modulo	Materia Destino	Materia Origen	Horas	Creditos	Calificacion	Porcentaje Cumplimiento	Temas	Observaciones
<input type="text" value="1"/>	<input type="text" value="Estadistica descripti"/>	<input type="text" value="Algebra"/>	<input type="text" value="48"/>	<input type="text" value="3"/>	<input type="text" value="28"/>	<input type="text" value="87.5"/>	<input type="text" value="x"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Estadistica descripti"/>	<input type="text" value="Calculo I"/>	<input type="text" value="48"/>	<input type="text" value="3"/>	<input type="text" value="28"/>	<input type="text" value="75"/>	<input type="text" value="x"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Fisica"/>	<input type="text" value="Contabilidad Genera"/>	<input type="text" value="80"/>	<input type="text" value="8"/>	<input type="text" value="28"/>	<input type="text" value="38.235294"/>	<input type="text" value="x"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Quimica"/>	<input type="text" value="Fundamentos Inform"/>	<input type="text" value="40"/>	<input type="text" value="4"/>	<input type="text" value="28"/>	<input type="text" value="21.739130"/>	<input type="text" value="x"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Fisica"/>	<input type="text" value="Contabilidad Genera"/>	<input type="text" value="80"/>	<input type="text" value="8"/>	<input type="text" value="28"/>	<input type="text" value="38.235294"/>	<input type="text" value="x"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Quimica"/>	<input type="text" value="Fundamentos Inform"/>	<input type="text" value="40"/>	<input type="text" value="4"/>	<input type="text" value="28"/>	<input type="text" value="21.739130"/>	<input type="text" value="x"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Quimica"/>	<input type="text" value="Logica matematica"/>	<input type="text" value="40"/>	<input type="text" value="4"/>	<input type="text" value="28"/>	<input type="text" value="30.434782"/>	<input type="text" value="x"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Proceso Investigati"/>	<input type="text" value="Lenguaje de alto nive"/>	<input type="text" value="100"/>	<input type="text" value="12"/>	<input type="text" value="28"/>	<input type="text" value="60"/>	<input type="text" value="x"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Fisica"/>	<input type="text" value="Metodologia y tecno"/>	<input type="text" value="80"/>	<input type="text" value="8"/>	<input type="text" value="28"/>	<input type="text" value="44.117647"/>	<input type="text" value="x"/>	<input type="text"/>
<input type="text" value="1"/>	<input type="text" value="Estadistica descripti"/>	<input type="text" value="Calculo II"/>	<input type="text" value="48"/>	<input type="text" value="3"/>	<input type="text" value="28"/>	<input type="text" value="87.5"/>	<input type="text" value="x"/>	<input type="text"/>


Universidad Nacional de Loja
Homologa 1.0 © 2013

Figura 32. Pantalla resultado previo de la petición homologación

PANTALLA RESULTADO DE HOMOLOGACIÓN FINALIZADA


Homologa
Usuario: secre
Cerrar sesión

Inicio
Homologar
Mi perfil
?

Resultado Petición de Homologación

Cedula

Nombre

Apellido

Telefono

Universidad

Origen

Carrera

Origen

Mallaorigen

Resumen

Modulo	Materia Destino	Materia Origen	Horas	Creditos	Calificacion	Porcentaje Cumplimiento	Temas	Observaciones
1	Matematicas	Matematicas 1	80	8	7	100	x	
1	Fisica	Fisica Basica	80	8	7	100	x	
1	Geometria Plana	Geometria	40	4	7	100	x	
1	Quimica	Bases Quimicas	40	4	7	100	x	
1	Proceso Investigativ	Investigacion	100	12	7	100	x	
2	Matematicas	Matematicas 2	80	8	7	100	x	
2	Fisica 1	Fisica Avanzada	80	8	7	100	x	
2	Algebra Lineal	Algebra Lineal	60	6	7	100	x	
2	Geometria Analitica	Geometria	60	6	7	100	x	
2	Proceso Investigativ	Investigacion 2	80	8	7	100	x	

Imprimir
Inicio


Universidad Nacional de Loja
Homologa 1.0 © 2013

Figura 33. Pantalla resultado de la petición de homologación

FRAGMENTO: Resultado de Homologación

TABLA XXV
FRAGMENTO: RESULTADO DE HOMOLOGACIÓN

Fragmento:	Resultado de Homologación
Caso de Uso Referido:	Manejar Homologación
Curso Básico	
1. El Homologador selecciona la petición de homologación y elige la opción <Homologar> en la pantalla “Peticiones de Homologación”	
2. El sistema valida que el estado de la petición de homologación sea diferente a Homologado.	
3. El sistema llama al fragmento “Ejecutar Homologación”	
4. El sistema activa los botones <Finalizar Homologación> y <Regresar> en la pantalla “Resultado Homologación”	
5. El homologador revisa los valores del porcentaje de cumplimiento y elige la opción <Finalizar Homologación> en la pantalla “Resultado Homologación”	
6. El sistema guarda la petición de homologación.	
7. El sistema cambia el estado de la petición de homologación a Homologado.	
8. Finaliza Caso de Uso	
Cursos Alternos	

1A. El sistema valida que el estado de la petición sea homologada.

1. El Homologador selecciona la petición de homologación y elige la opción <Homologar> en la pantalla “Petición de Homologación”
2. El sistema valida que el estado de la petición de homologación sea Homologado
3. El sistema recupera y presenta la petición de homologación con las materias destino y las materias origen que fueron homologadas y los valores correspondientes a los porcentajes de cumplimiento en la pantalla “Resultado Homologación”.
4. El sistema activa los botones <Imprimir> e <Inicio> en la pantalla “Resultado Homologación”.
5. Continúa con el paso 7 del curso básico

4A. El homologador desea cambiar las calificaciones de las materias origen o desea ingresar más contenidos para la homologación.

1. El homologador elige la opción <Regresar> en la pantalla “Resultado Homologación”
2. El sistema ejecuta el paso 9 del curso básico del Caso de Uso Administrar Homologaciones.
3. Continúa con el paso 7 del curso básico

DIAGRAMA DE SECUENCIA

Fragmento: Resultado de Homologación

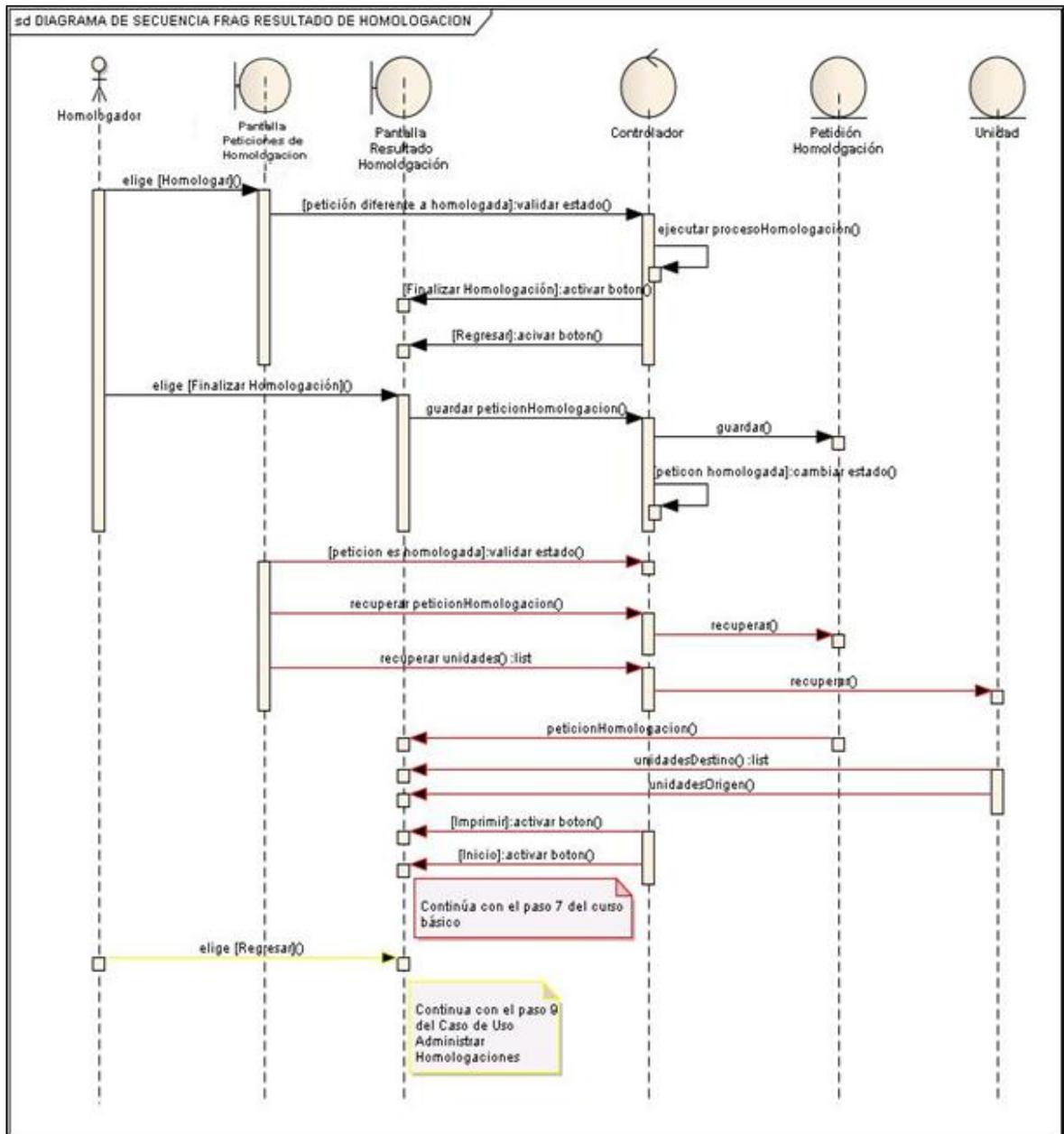


Figura 34. Diagrama de secuencia resultado de la petición de homologación

Fragmento: Ejecutar Homologación

**TABLA XXVI
FRAGMENTO: EJECUTAR HOMOLOGACIÓN**

Fragmento:	Ejecutar Homologación
Caso de Uso Referido:	Fragmento Resultado de Homologación.
Curso Básico	
1.	El sistema compara las etiquetas de cada unidad del Módulo 1 (hasta el Modulo n...) de la Carrera Destino con las contenidos de cada unidad (materia) de la Carrera Origen y guarda el número de contenidos iguales encontradas en cada unidad (materia) de la Carrera Destino.
2.	El sistema calcula el porcentaje de similitudes utilizando el número de contenidos máximo ingresado de cada unidad del Módulo 1 de la Carrera Destino con el número de contenidos iguales guardado en cada unidad.
3.	El sistema suma el porcentaje calculado de similitudes en el módulo 1 y lo divide para el número de unidades (materias) que tiene dicho modulo, si este resultado es mayor al 80% el sistema continua con el modulo siguiente.
4.	Finaliza caso de uso
CURSOS ALTERNOS	
3A. El resultado es igual a 80%	
1.	Si el resultado de la suma del porcentaje de similitudes es igual al 80% el sistema continua con el modulo siguiente.
2.	Caso de uso continúa con el paso 4 del curso básico.
3B. El resultado es menor a 80%	
1.	Si el resultado de la suma del porcentaje de similitudes es menor que el 80% el sistema guarda el modulo en el que se encuentra la comparación
2.	Caso de uso continúa con el paso 4 del curso básico.

PANTALLA ADMINISTRAR USUARIOS

Homologa Usuario: administrador12
Cerrar sesión

Inicio Universidad Área Carrera Malla Curricular Usuarios Mi perfil

Buscar Usuario

Nombre
Cedula
Habilitado
Buscar

Lista de Usuarios

Resultados Nuevo Usuario

Id	Apellido	Nombre	Cedula	Telefono	Acciones
22	ARROBO	JAVIER	1103735255	888	Eliminar Editar
23	ANDRADE	ANDRES	1103735444	666	Eliminar Editar
24	ACARO	PATRICIA	1103735222	55	Eliminar Editar
26	BENITEZ	CARLOS	1037334945	666	Eliminar Editar
27	CUENCA	SOLEDAD	1103735436	76554	Eliminar Editar
32	DIEGUEZ	GLENDA	1103765432	777	Eliminar Editar

Figura 35. Pantalla administrar usuarios

PANTALLA EDITAR USUARIO

Homologa Usuario: administrador12
Cerrar sesión

Inicio Universidad Área Carrera Malla Curricular Usuarios Mi perfil

Usuario

Apellido ARROBO
Nombre JAVIER
Cedula 1103735255
Telefono 888
Email SUUU@g.com

Nombre
Usuario

Rol ESTUDIANTE
SECRETARIA
DOCENTE
ADMINISTRADOR

Habilitado

Guardar Cancelar

Universidad Nacional de Loja
Buenos Aires, 12 de Mayo, 2012

Figura 36. Pantalla editar usuario

CASO DE USO: Administrar Usuarios

**TABLA XXVII
CASO DE USO: ADMINISTRAR USUARIOS**

Caso de Uso:	Caso de uso: administrar usuarios
Precondición:	El Administrador de área debe haber ingresado al sistema
Post Condición:	Usuarios quedan modificados de alguna manera.
Referencia a Requerimientos:	
Curso Básico	
1. El administrador de Área elige la opción < Usuarios> en la pantalla “Menú”	
2. El sistema carga y presenta la pantalla “Administrar Usuarios”	
3. El Administrador de Área selecciona un usuario de la lista de Usuarios y elige la opción <Editar> en la pantalla “Administrar Usuarios”	
4. El sistema recupera el usuario seleccionado y la presenta en la pantalla “Editar Usuario”	
5. El Administrador de Área ingresa los datos del usuario en la pantalla “Editar Usuario”	
6. El Administrador de Área elige la opción <Guardar> en la pantalla “Editar Usuario”	
7. El sistema valida que los campos requeridos no estén en blanco.	
8. El sistema guarda el usuario	
9. El caso de uso finaliza	
Cursos Alternos	
3A. EL ADMINISTRADOR DE AREA DESEA CREAR UN NUEVO USUARIO	
1. El Administrador de Área elige la opción <Nuevo Usuario> en la pantalla “Administrar Usuarios”	
2. El sistema crea un nuevo usuario y presenta la pantalla “Editar Usuario”	
3. Continúa con el paso 5 del curso básico	

<p>3B. EL ADMINISTRADOR DE AREA DESEA BUSCAR UN USUARIO</p> <ol style="list-style-type: none"> 1. El Administrador de Área ingresa el Nombre, la Cedula o el estado del Usuario a buscar y elige la opción <Buscar> en la pantalla “Administrar Usuarios” 2. El sistema busca el usuario y lo presenta en la pantalla “Administrar Usuarios” 3. Continúa con el paso 9 del curso básico
<p>3C. EL ADMINISTRADOR DE AREA DESEA ELIMINAR UN USUARIO</p> <ol style="list-style-type: none"> 1. El Administrador de Área elige la opción <Eliminar> en la pantalla “Administrar Usuarios” 2. El sistema llama al fragmento “Eliminar Usuario” 3. Continúa con el paso 9 del curso básico
<p>6A. EL ADMINISTRADOR NO DESEA GUARDAR EL USUARIO Y ELIGE LA OPCION CANCELAR</p> <ol style="list-style-type: none"> 1. El Administrador de Área elige la opción <Cancelar> en la pantalla “Editar Carrera” 2. continua con el paso 2 del curso básico
<p>8A. EL SISTEMA NO GUARDA EL USUARIO.</p> <ol style="list-style-type: none"> 1. El sistema no guarda el usuario 2. El sistema presenta los campos con error con asterisco (*) y con un mensaje en las campos que no pueden ser nulos en la pantalla “Editar Usuario” 3. continua con el paso 5 del curso básico

DIAGRAMA DE SECUENCIA

Caso de Uso: Administrar Usuarios

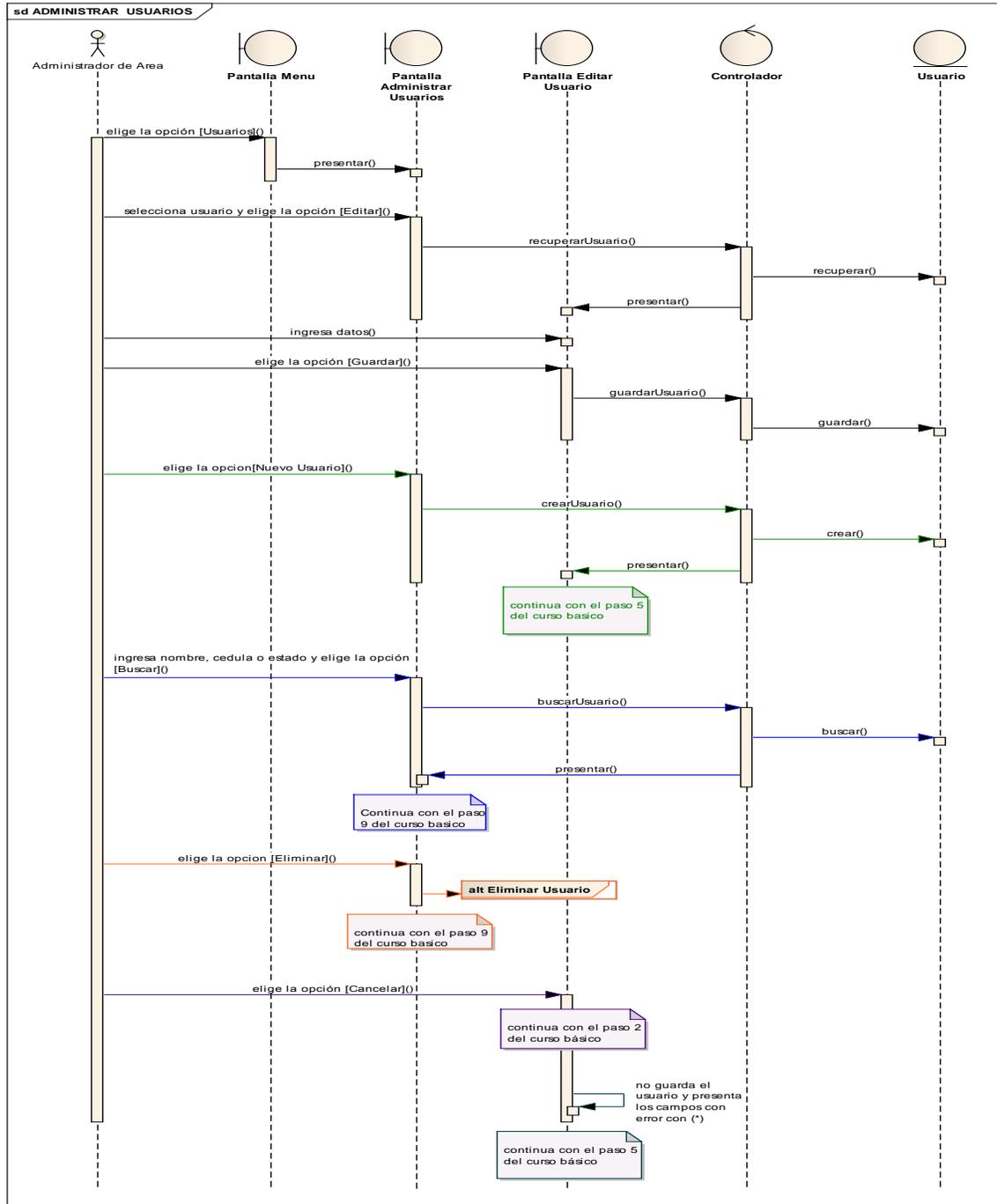


Figura 37. Diagrama de secuencia administrar usuarios

PANTALLA ELIMINAR USUARIO

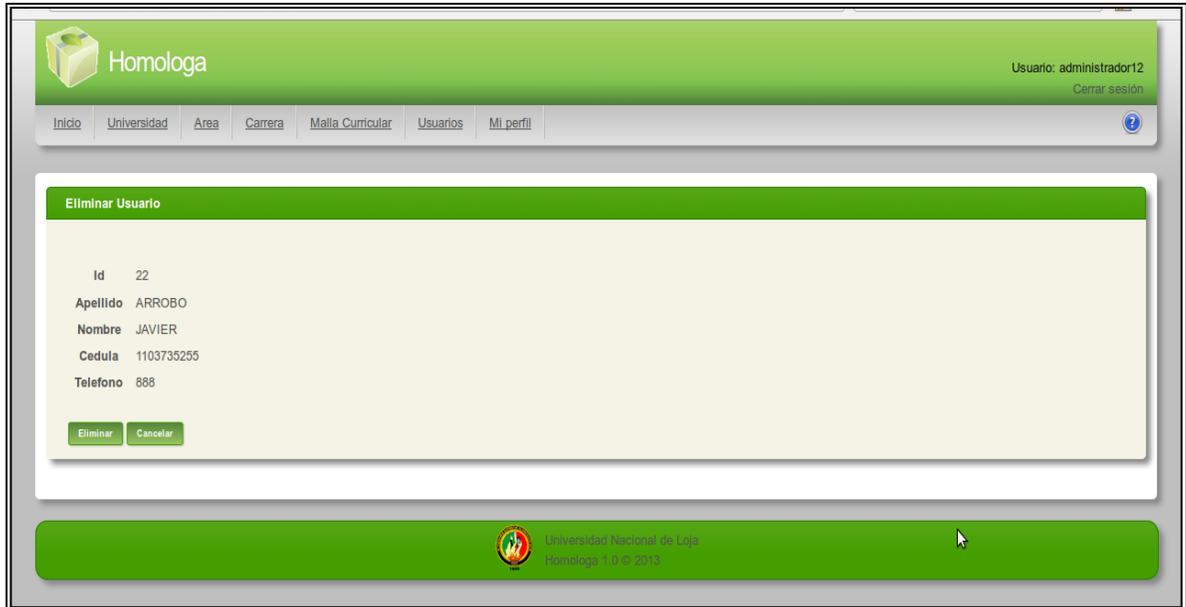


Figura 38. Pantalla eliminar usuario

FRAGMENTO: Eliminar Usuario

TABLA XXVIII
FRAGMENTO: ELIMINAR USUARIO

FRAGMENTO: ELIMINAR USUARIO	
CASO DE USO REFERIDO:	Administrar Usuarios.
Curso Básico	
1. El recupera los datos del usuario y los presenta en la pantalla “Eliminar Usuario”	
2. El administrador de Área visualiza los datos del usuario y elige la opción <Eliminar> en la pantalla “Eliminar Usuario”	
3. El sistema verifica si el usuario no posee peticiones de homologación asignadas y elimina el usuario seleccionado	
4. El sistema presenta la pantalla “Administrar Usuarios”	

5. El caso de uso finaliza
Cursos Alternos
<p>2A. LA ADMINISTRADOR DE AREA DESEA CANCELAR ELIMINAR USUARIO.</p> <ol style="list-style-type: none"> 1. El administrador de Área visualiza los datos del usuario y elige la opción <Cancelar> en la pantalla “Eliminar Usuario” 2. Continúa con el paso 4 del curso básico.
<p>3A. EL USUARIO TIENE UNA PETICIÓN DE HOMOLOGACIÓN ASIGNADA.</p> <ol style="list-style-type: none"> 1. El sistema presenta un mensaje (“usuario tiene peticiones de homologación asignadas, no se puede borrar”) en la pantalla “Administrar Usuarios” 2. El Administrador de Área elige la opción <Cancelar> en la pantalla “Eliminar Usuario” 3. Continúa con el paso 4 del curso básico

DIAGRAMA DE SECUENCIA

Fragmento: Eliminar Usuario

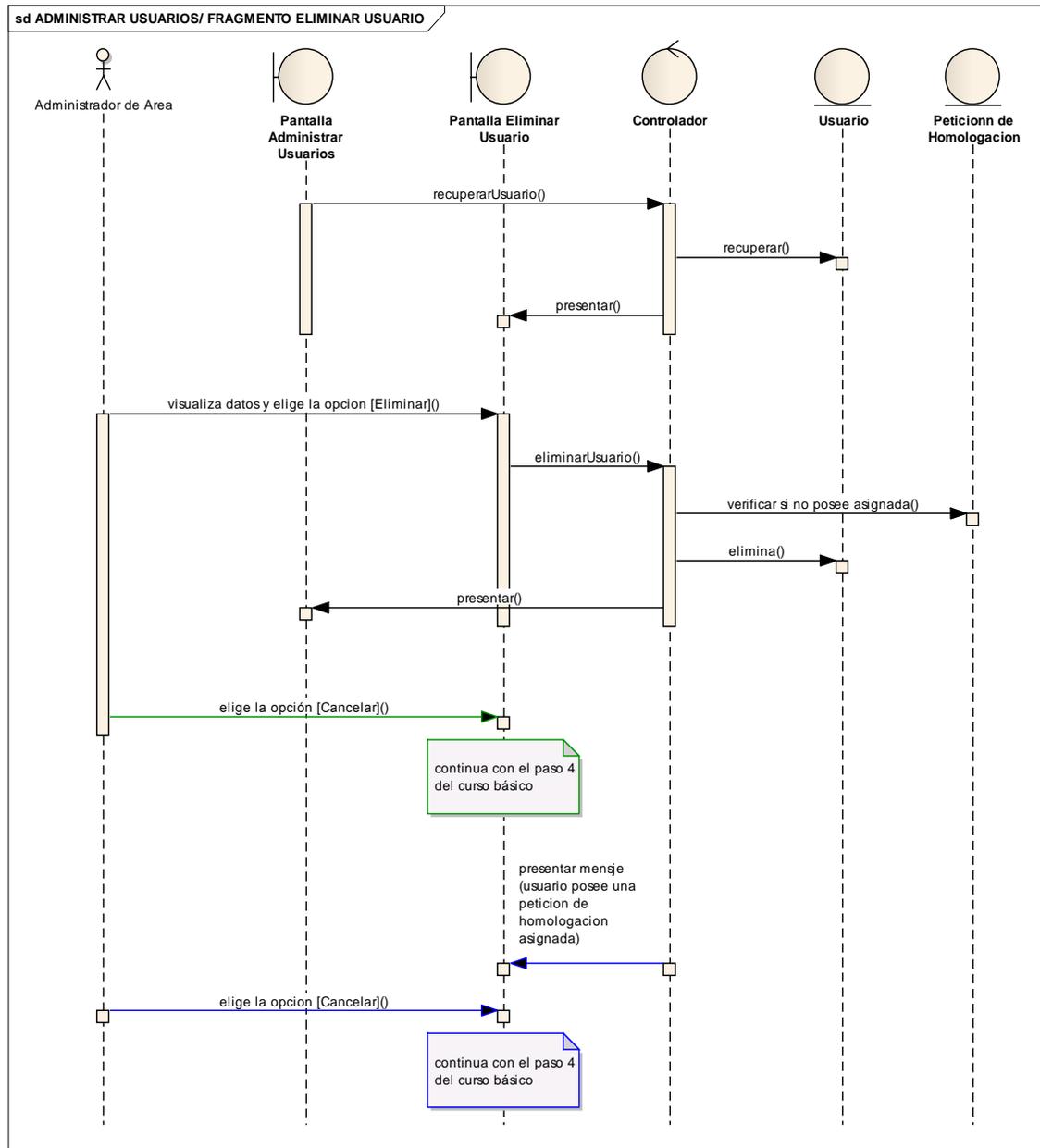


Figura 39. Diagrama de secuencia eliminar usuario

PANTALLA ADMINISTRAR UNIVERSIDADES

Homologa Usuario: administrador12
Cerrar sesión

Inicio Universidad Area Carrera Malla Curricular Usuarios Mi perfil

Buscar Universidad

Nombre

Buscar

Lista de Universidades

Resultados Nueva Universidad

Id	Codigo	Nombre	Descripción	Acciones
8	ESPOCH	Escuela Politecnica del Chimborazo	ESPOCH	Eliminar Editar
3	UNL	Universidad Nacional de Loja	Universidad Nacional de Loja	Eliminar Editar
4	UTPL	Universidad Nacional Tecnica Particular de Loja	Universidad Tecnica Loja	Eliminar Editar

Universidad Nacional de Loja
Versión: 1.0.0.2011

Figura 40. Pantalla administrar universidades

PANTALLA EDITAR AREA

Homologa Usuario: administrador12
Cerrar sesión

Inicio Universidad Area Carrera Malla Curricular Usuarios Mi perfil

Area

Nombre

Descripcion

Carreras Agregar

Codigo	Nombre	Acciones

Guardar Cancelar

Universidad Nacional de Loja
Versión: 1.0.0.2011

Figura 41. Pantalla editar área

CASO DE USO: Administrar Universidades

TABLA XXIX

CASO DE USO: ADMINISTRAR UNIVERSIDADES

Caso de Uso:	CASO DE USO: ADMINISTRAR UNIVERSIDADES
Precondición:	El Administrador de área debe haber ingresado al sistema
Post Condición:	Universidades quedan modificadas de alguna manera.
Referencia a Requerimientos:	RQ 12.
Curso Básico	
1. El administrador de Área elige la opción < Universidades> en la pantalla “Menú”	
2. El sistema carga y presenta la pantalla “Administrar Universidades”	
3. El Administrador de Área selecciona una universidad de la lista de universidades y elige la opción <Editar> en la pantalla “Administrar Universidades”	
4. El sistema recupera la universidad seleccionada y la presenta en la pantalla “Editar Universidad”	
5. El Administrador de Área ingresa los campos(código, nombre, descripción) en la pantalla “Editar Universidad”	
6. El Administrador de Área elige la opción <Guardar> en la pantalla “Editar Universidad”	
7. El sistema valida que la Universidad tenga datos en los campos Código y Nombre.	
8. El sistema guarda la universidad	
9. El caso de uso finaliza	
Cursos Alternos	
3A. EL ADMINISTRADOR DE AREA DESEA CREAR UNA NUEVA UNIVERSIDAD	
1. El Administrador de Área elige la opción <Nueva Universidad> en la pantalla “Administrar Universidades”	
2. El sistema crea una nueva universidad y presenta la pantalla “Editar Universidad”	
3. Continúa con el paso 5 del curso básico	

<p>3B. EL ADMINISTRADOR DE AREA DESEA BUSCAR UNA UNIVERSIDAD</p> <ol style="list-style-type: none"> 1. El Administrador de Área ingresa el nombre de la universidad a buscar y elige la opción <Buscar> en la pantalla “Administrar Universidades” 2. El sistema busca la universidad y la presenta en la pantalla “Administrar Universidades” 3. continua con el paso 9 del curso básico
<p>3C. EL ADMINISTRADOR DE AREA DESEA ELIMINAR UNA UNIVERSIDAD</p> <ol style="list-style-type: none"> 1. El Administrador de Área elige la opción <Eliminar> en la pantalla “Administrar Universidades” 2. El sistema llama al fragmento “Eliminar Universidad” 3. Continúa con el paso 9 del curso básico
<p>6A. EL ADMINISTRADOR NO DESEA GUARDAR LA UNIVERSIDAD Y ELIGE LA OPCION CANCELAR</p> <ol style="list-style-type: none"> 1. El Administrador de Área elige la opción <Cancelar> en la pantalla “Editar Universidad” 2. continua con el paso 2 del curso básico
<p>8A. EL SISTEMA NO GUARDA LA UNIVERSIDAD.</p> <ol style="list-style-type: none"> 1. El sistema no guarda la universidad 2. El sistema presenta los campos con error con asterisco (*) y con un mensaje en las campos que no pueden ser nulos en la pantalla “Editar Universidad” 3. continua con el paso 5 del curso básico

DIAGRAMA DE SECUENCIA

Caso de Uso: Administrar Universidades

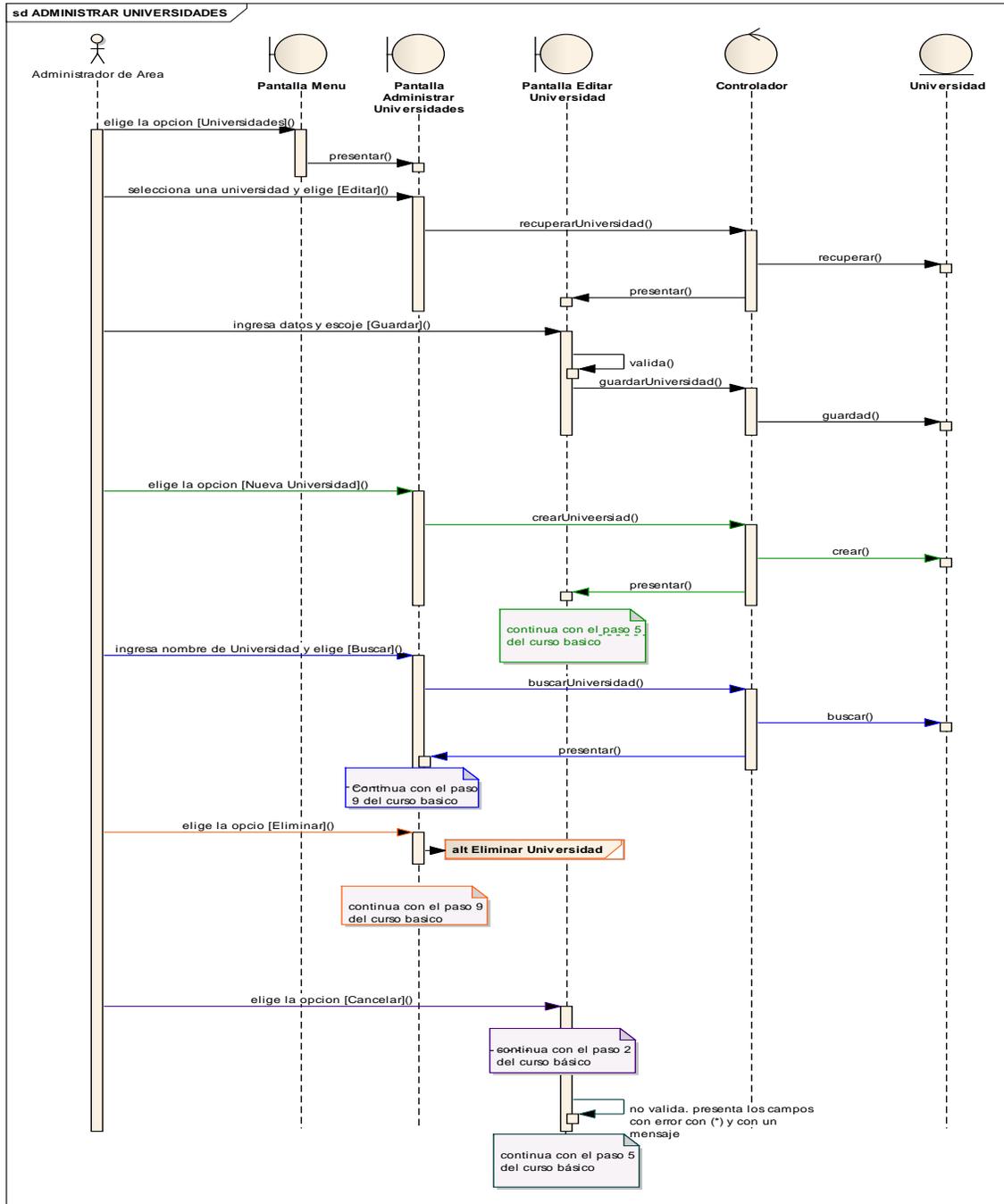


Figura 42. Diagrama de secuencia administrar universidades

PANTALLA ELIMINAR UNIVERSIDAD

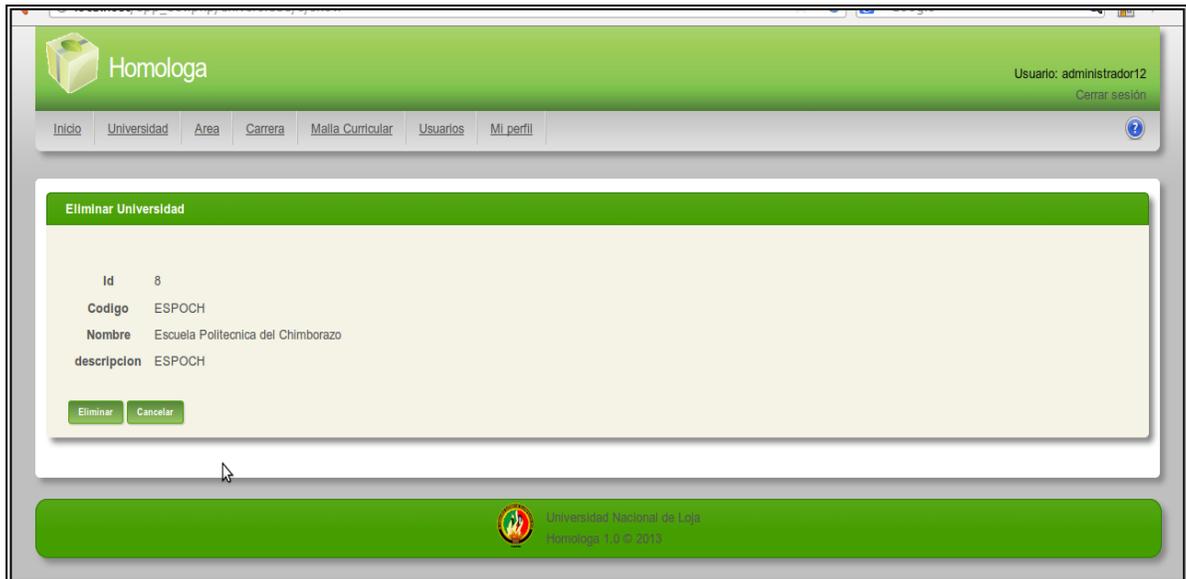


Figura 43. Pantalla eliminar universidades

FRAGMENTO: Eliminar Universidad

TABLA XXX
FRAGMENTO: ELIMINAR UNIVERSIDAD

Fragmento: Eliminar Universidad	
CASO DE USO REFERIDO:	Administrar Universidades.
Curso Básico	
1. El sistema recupera los datos de la universidad los presenta en la pantalla "Eliminar Universidad"	
2. El administrador de Área visualiza los datos de la universidad y elige la opción <Eliminar> en la pantalla "Eliminar Universidad"	
3. El sistema verifica si la universidad no posee carreras asignadas, elimina la universidad seleccionada	

4. El sistema presenta la pantalla “Administrar Universidades”
5. El caso de uso finaliza
Cursos Alternos
2A. La administrador de area desea cancelar eliminar universidad.
<ol style="list-style-type: none"> 1. El administrador de Área visualiza los datos de la universidad y elige la opción <Cancelar> en la pantalla “Eliminar Universidad” 2. Continúa con el paso 4 del curso básico.
3A. La universidad tiene carreras asignadas.
<ol style="list-style-type: none"> 1. El sistema presenta un mensaje (“Universidad tiene carreras, borre primero las carreras”) en la pantalla “Eliminar Universidad” 2. El Administrador de Área elige la opción <Cancelar> en la pantalla “Eliminar Universidad” 3. Continúa con el paso 4 del curso básico

DIAGRAMA DE SECUENCIA

Fragmento: Eliminar Universidad

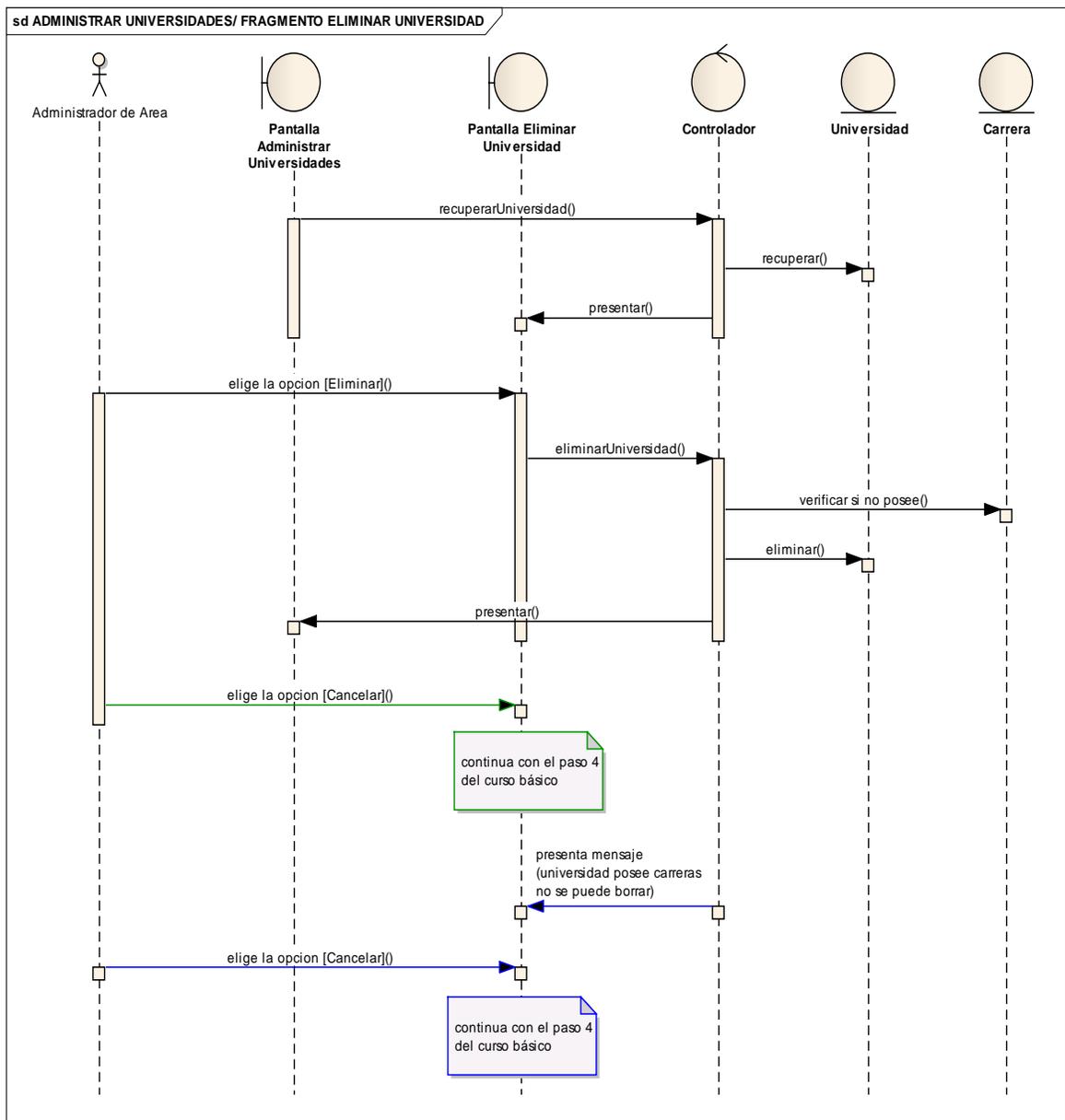


Figura 44. Diagrama de secuencia eliminar universidad

PANTALLA ADMINISTRAR ÁREAS

Homologa Usuario: administrador12
Cerrar sesión

Inicio Universidad Area Carrera Malla Curricular Usuarios MI perfil

Buscar Área

Nombre

Buscar

Lista de Áreas

Resultados Nueva Area

Id	Nombre	Descripción	Acciones
1	AERNIER	Area de Energia Industria y Recursos Materiales no Renovables	Eliminar Editar
2	JURIDICA		Eliminar Editar

Figura 45. Pantalla administrar áreas

PANTALLA EDITAR ÁREAS

Homologa Usuario: administrador12
Cerrar sesión

Inicio Universidad Area Carrera Malla Curricular Usuarios MI perfil

Area

Nombre AERNIER

Descripción Area de Energia Industria y Recursos Materiales

Carreras Agregar

Codigo	Nombre	Acciones
<input type="text" value="1"/>	<input type="text" value="Sistemas"/>	Quitar
<input type="text" value="2"/>	<input type="text" value="Electromecanica"/>	Quitar
<input type="text" value="1"/>	<input type="text" value="Minas"/>	Quitar

Guardar Cancelar

Figura 46. Pantalla editar área

CASO DE USO: Administrar Áreas

TABLA XXXI
CASO DE USO: ADMINISTRAR ÁREAS

Caso de Uso:	Caso De Uso: Administrar Áreas
Precondición:	El Administrador de Área debe haber ingresado al sistema.
Post Condición:	La información de alguna Área queda modificada de alguna manera.
Referencia a Requerimientos:	RQ 11.
Curso Básico	
1. El Administrador de Área elige la opción <Área> en la pantalla “Menú”	
2. El sistema carga y presenta la pantalla “ Administrar Áreas”	
3. El Administrador de Área selecciona una área y elige la opción <Editar> en la pantalla “Administrar Áreas”	
4. El sistema recupera el área seleccionada con las carreras pertenecientes a esta área y las presenta en la pantalla “Editar Área”	
5. El sistema llama al fragmento “ Editar Área ”	
6. El caso de uso finaliza	
Cursos Alternos	
3A. El administrador de area desea crear una nueva área	
1. El Administrador de Área elige la opción <Nueva Área> en la pantalla “Administrar Área”	
2. El sistema crea una nueva área y presenta la pantalla “Editar Área”	
3. Continúa con el paso 5 del curso básico	

3B. El administrador de area desea buscar un área

1. El Administrador de Área ingresa el nombre del área a buscar y escoge la opción <Buscar> en la pantalla “Administrar Áreas”
2. El sistema busca el área y la presenta en la pantalla “Administrar Áreas”
3. Continúa con el paso 6 del curso básico

3C. El administrador de area desea eliminar un área

1. El Administrador de Área elige la opción <Eliminar> en la pantalla “Administrar Áreas”
2. El sistema llama al fragmento “**Eliminar Área**”
3. Continúa con el paso 6 del curso básico

DIAGRAMA DE SECUENCIA

Caso de Uso: Administrar Áreas

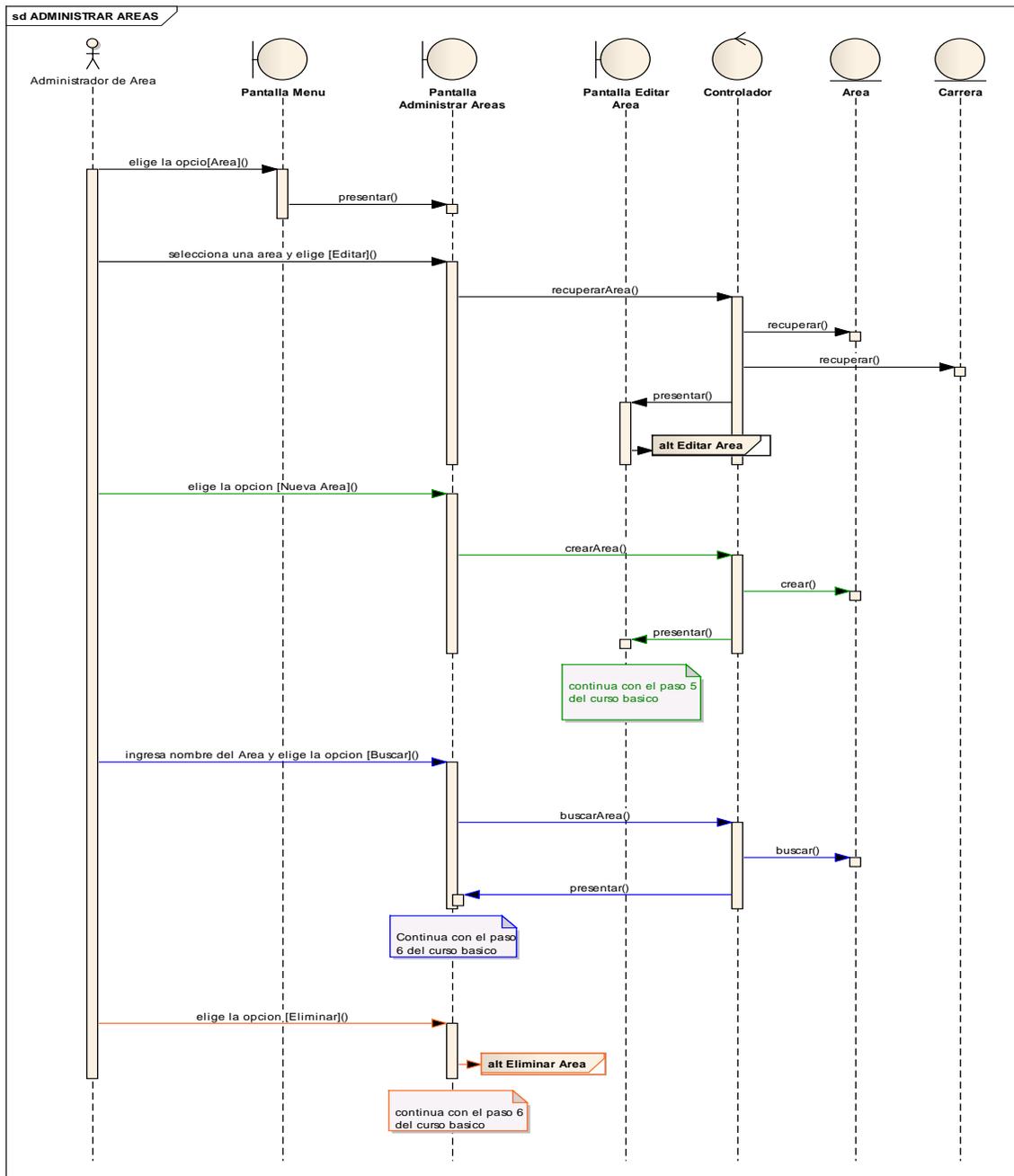


Figura 47. Diagrama de secuencia administrar áreas

PANTALLA ELIMINAR ÁREA

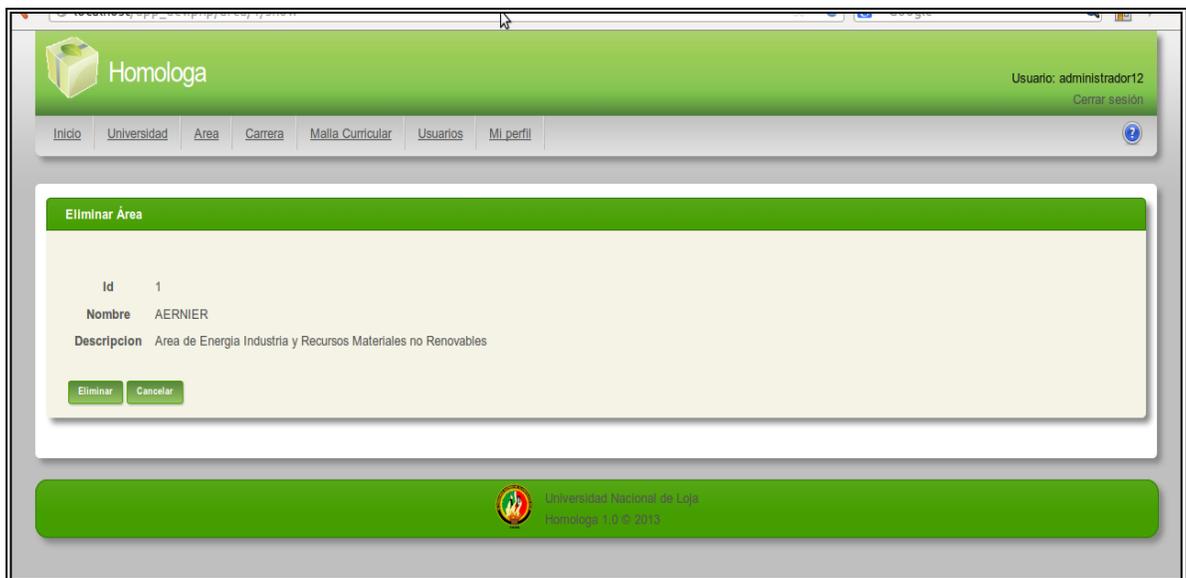


Figura 48. Pantalla eliminar área

FRAGMENTO: Eliminar Área

TABLA XXXII
FRAGMENTO: ELIMINAR ÁREA

Fragmento: Eliminar Área	
CASO DE USO REFERIDO:	ADMINISTRAR AREAS
Curso Básico	
1. El sistema recupera la área seleccionada y presenta los datos en la pantalla "Eliminar Área"	
2. El Administrador de Área visualiza los datos del área y elige la opción <Eliminar> en la pantalla "Eliminar Área"	
3. El sistema verifica si el área no posee carreras asignadas a una petición de homologación, y elimina el área seleccionada.	
4. El sistema presenta la pantalla "Administrar Áreas"	
5. El caso de uso finaliza	

Cursos Alternos	
2A. El administrador de area desea cancelar eliminar área	<ol style="list-style-type: none">1. El Administrador de Área elige la opción <Cancelar> en la pantalla “Eliminar Área”2. Continúa con el paso 4 del curso básico.
3A. El area tiene carreras asignadas a una petición de homologación.	<ol style="list-style-type: none">1. El sistema presenta un mensaje (“No se puede borrar esta área”) en la pantalla “Eliminar Área”.2. El Administrador de Área elige la opción <Cancelar> en la pantalla “Eliminar Área”3. Continúa con el paso 4 del curso básico.

DIAGRAMA DE SECUENCIA

Fragmento: Eliminar Área

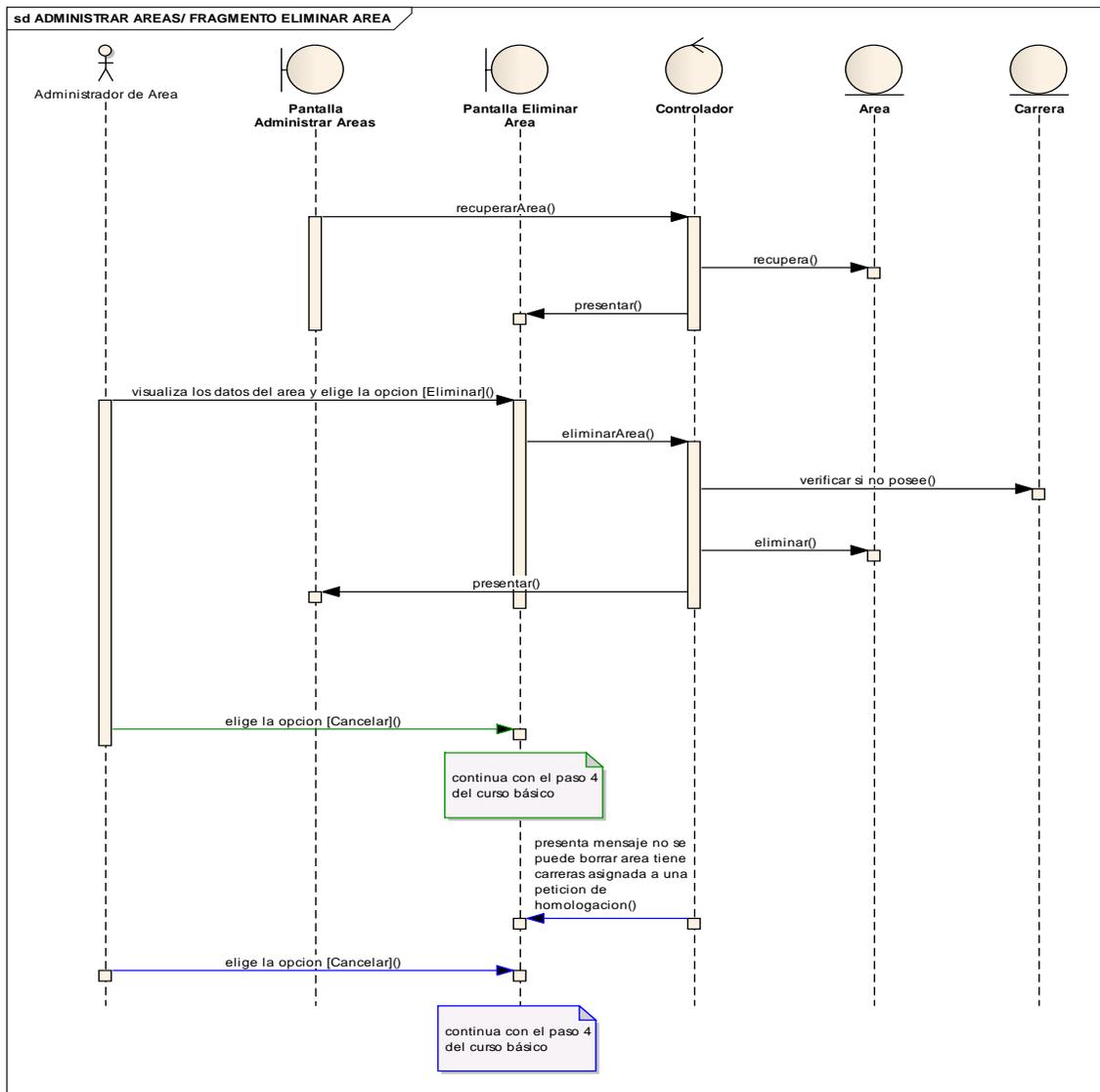


Figura 49. Diagrama de secuencia eliminar área

PANTALLA ADMINISTRAR MALLAS CURRICULARES

Homologa

Usuario: administrador12
Cerrar sesión

[Inicio](#) [Universidad](#) [Area](#) [Carrera](#) [Malla Curricular](#) [Usuarios](#) [Mi perfil](#)

Buscar Malla Curricula

Nombre

Buscar

Lista de Mallas Curriculares

Resultados [Nueva Malla Curricular](#)

Id	Universidad	Carrera	Año Pensun	Acciones
13	3 Universidad Nacional de Loja	5 Derecho	2008	Eliminar Editar
11	3 Universidad Nacional de Loja	5 Derecho	2000	Eliminar Editar
6	3 Universidad Nacional de Loja	4 Electromecanica	2012	Eliminar Editar
3	3 Universidad Nacional de Loja	4 Electromecanica	2001	Eliminar Editar
2	3 Universidad Nacional de Loja	3 Sistemas	2004	Eliminar Editar
1	3 Universidad Nacional de Loja	3 Sistemas	2003	Eliminar Editar

Figura 50. Pantalla Administrar Mallas Curriculares

CASO DE USO: Administrar Mallas Curriculares

TABLA XXXIII
CASO DE USO: ADMINISTRAR MALLAS CURRICULARES

Caso de Uso:	Administrar Mallas Curriculares
Precondición:	El Administrador de Área debe haber ingresado al sistema
Post Condición:	La información de alguna malla curricular queda modificada de alguna manera.
Referencia a Requerimientos:	RQ 3, RQ 4, RQ 6
Curso Básico	
1. El Administrador de Área elige la opción <Mallas Curriculares> en la pantalla “Menú”	
2. El sistema carga y presenta la pantalla “Administrar Mallas Curriculares”	
3. El Administrador de Área selecciona la malla curricular a modificar y elige la opción <Editar> en la pantalla “Administrar Mallas Curriculares”	
4. El sistema recupera los datos de la malla curricular seleccionada y los presenta en la pantalla “Editar Malla Curricular”	
5. El sistema llama al Fragmento “ Editar Malla Curricular ”	
6. Finaliza Caso de Uso	
Cursos Alternos	
3A. El administrador de área desea crear una nueva malla curricular	
1. El Administrador de Área elige la opción <Nueva Malla Curricular> en la pantalla “Administrar Mallas Curriculares”	
2. El sistema crea una nueva malla curricular y presenta la pantalla “Editar Malla Curricular”	
3. continua con el paso 5 del curso básico	
3B. El administrador de área desea buscar una malla curricular	
1. El Administrador de Área ingresa el nombre de la universidad y elige la opción <Buscar> en la pantalla “Administrar Mallas Curriculares”	
2. El sistema busca la malla curricular por el nombre de universidad ingresado y la presenta en la pantalla “Administrar Mallas curriculares”	
3. continua con el paso 6 del curso básico	

3C. El administrador de área desea eliminar una malla curricular

1. El Administrador de Área elige la opción <Eliminar> en la pantalla “Administrar Mallas Curriculares”
2. El sistema llama al fragmento “**Eliminar Malla Curricular**”
3. continua con el paso 6 del curso básico

DIAGRAMA DE SECUENCIA

Caso de Uso: Administrar Mallas Curriculares

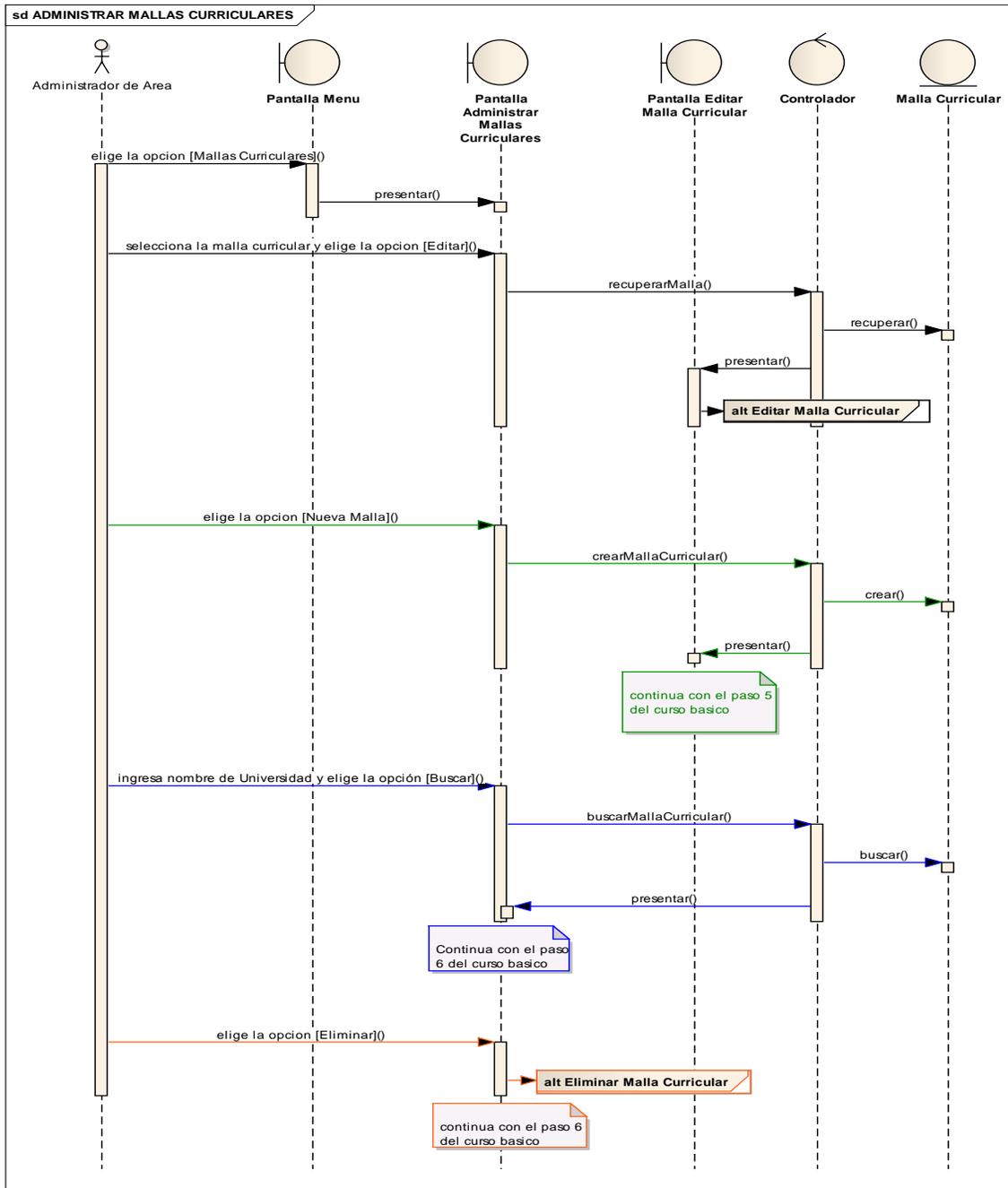


Figura 51. Diagrama de secuencia administrar mallas curriculares

PANTALLA EDITAR MALLA CURRICULAR

Homologa Usuario: administrador12
Cerrar sesión

Inicio Universidad Area Carrera Malla Curricular Usuarios Mi perfil

Malla Curricular

Universidad: 3 Universidad Nacional de Loja
Carrera: 3 Sistemas
Año: 2004
Duración: 11
Unidad de Tiempo: AÑOS

Niveles		Agregar
Numero de Nivel	Acciones	
1	Quitar	Unidades
2	Quitar	Unidades
1	Quitar	Unidades
2	Quitar	Unidades
3	Quitar	Unidades

Figura 52. Pantalla editar malla curricular

FRAGMENTO: Editar Malla Curricular

TABLA XXXIV
FRAGMENTO: EDITAR MALLA CURRICULAR

Fragmento: Editar Malla Curricular	
CASO DE USO REFERIDO:	ADMINISTRAR MALLAS CURRICULARES
Curso Básico	
1. El Administrador de Área selecciona la Universidad, Carrera y Unidad de Tiempo, e ingresa los datos requeridos de Año y Duración en la pantalla “Editar Malla Curricular”	
2. El Administrador de Área elige la opción <Guardar> en la pantalla “Editar Malla Curricular”	
3. El sistema valida que los campos requeridos tengan datos ingresados.	

4. El sistema actualiza la lista de niveles de la Malla Curricular
5. El sistema guarda la Malla Curricular.
6. El caso de uso finaliza
Cursos Alternos
2A. El administrador de área desea agregar un nuevo nivel a la malla curricular
<ol style="list-style-type: none"> 1. El Administrador de Área elige la opción <Agregar> en la pantalla “Editar Malla Curricular” 2. El sistema agrega el campos vacíos de Numero de Nivel en la lista de Niveles en la pantalla “Editar Malla Curricular” 3. El Administrador de Área ingresa el Numero de Nivel en la lista de Niveles en la pantalla editar “Editar Malla Curricular” 4. Continúa con el paso 2 del curso básico.
2B. El administrador de área desea editar el numero de nivel
<ol style="list-style-type: none"> 1. El Administrador de Área elige y hace clic en Numero de Nivel que desea editar, en la lista de niveles en la pantalla “Editar Malla Curricular” 2. El Administrador de Área ingresa el Numero de Nivel en la lista de Niveles en la pantalla editar “Editar Malla Curricular” 3. Continúa con el paso 2 del curso básico
2C. El administrador de área desea quitar un nivel de la malla curricular
<ol style="list-style-type: none"> 1. El Administrador de Área selecciona un nivel y elige la opción <Quitar> en la pantalla “Editar Malla Curricular” 2. El sistema quita el nivel de la lista de Niveles de la Malla en la pantalla “Editar Malla Curricular” 3. Continúa con el paso 2 del curso básico
2D. El administrador de área desea ir a la lista de unidades de un nivel de la malla curricular
<ol style="list-style-type: none"> 1. El Administrador de Área selecciona un nivel y elige la opción <Unidades> en la pantalla “Editar Malla Curricular” 2. El sistema llama al fragmento “Estructura Pensum” 3. Continúa con el paso 2 del curso básico

2E. El administrador de área desea cancelar editar malla curricular

1. El Administrador de Área elige la opción <Cancelar> en la pantalla “Editar Malla Curricular”
2. Continúa con el paso 6 del curso básico.

5A. El sistema no guarda la malla curricular

1. El sistema no guarda la malla curricular
2. El sistema presenta los campos con error con asterisco * y con un mensaje en los campos que no pueden estar en blanco en la pantalla “Editar Malla Curricular”
3. Continúa con el paso 1 del curso básico

DIAGRAMA DE SECUENCIA

Fragmento: Editar Malla Curricular

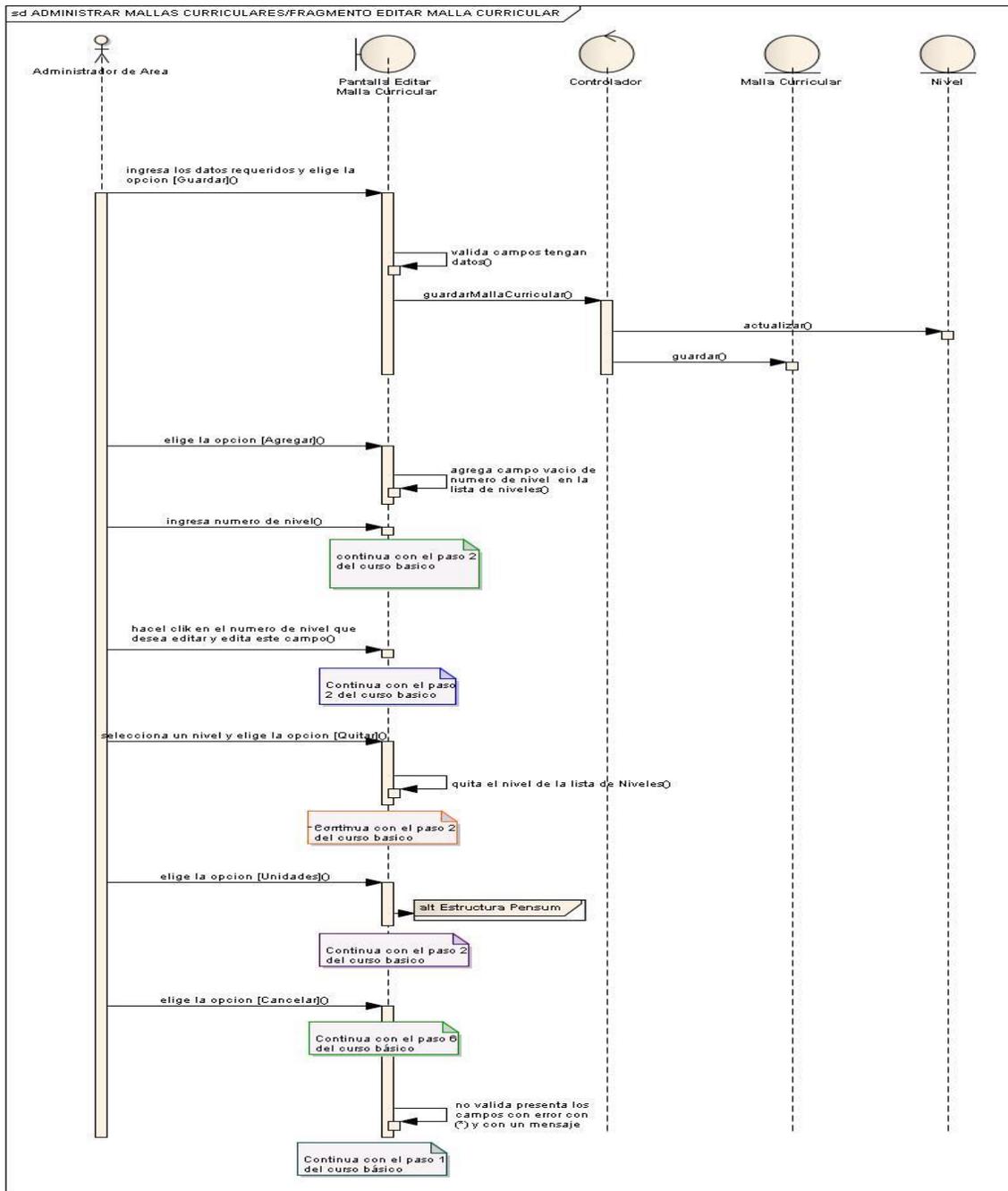


Figura 53. Diagrama de secuencia editar malla curricular

PANTALLA ELIMINAR MALLA CURRICULAR

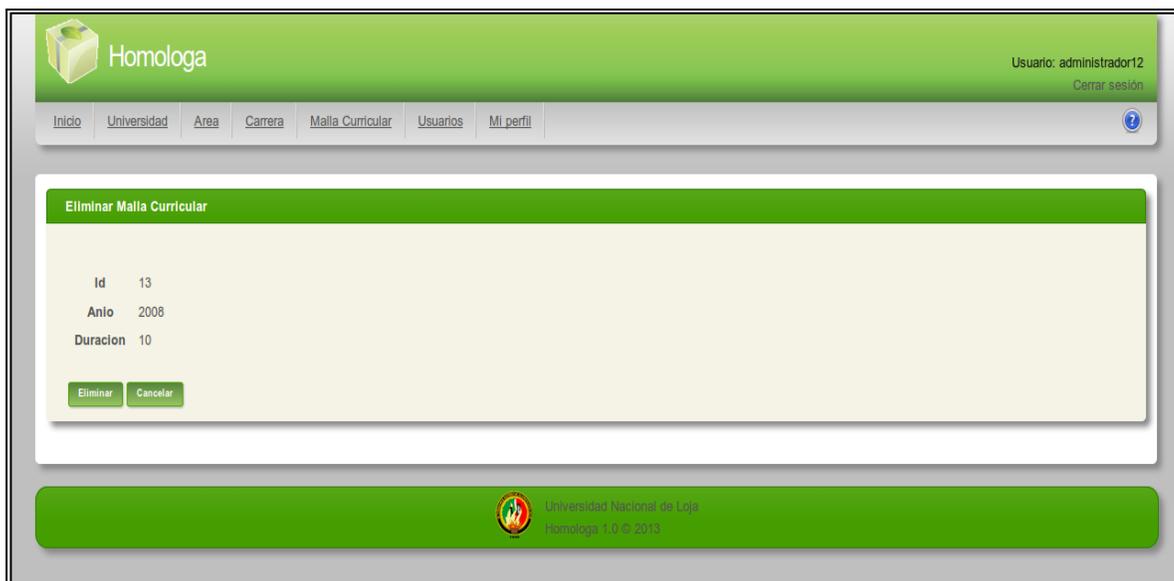


Figura 54. Pantalla eliminar malla curricular

FRAGMENTO: Eliminar Malla Curricular

TABLA XXXV
FRAGMENTO: ELIMINAR MALLA CURRICULAR

Fragmento: Eliminar Malla Curricular	
CASO DE USO REFERIDO:	ADMINISTRAR MALLAS CURRICULARES
Curso Básico	
1. El sistema recupera la malla curricular seleccionada y presenta los datos en la pantalla "Eliminar Malla Curricular"	
2. El Administrador de Área visualiza los datos de la malla curricular y elige la opción <Eliminar> en la pantalla "Eliminar Malla Curricular"	
3. El sistema verifica si la malla no está asignadas a una petición de homologación, y elimina la malla seleccionada.	
4. El sistema presenta la pantalla "Administrar Mallas Curriculares"	
5. El caso de uso finaliza	
Cursos Alternos	

2A. EL ADMINISTRADOR DE ÁREA DESEA CANCELAR ELIMINAR CURRICULAR	MALLA
<ol style="list-style-type: none">1. El Administrador de Área elige la opción <Cancelar> en la pantalla “Eliminar Malla Curricular”2. Continúa con el paso 4 del curso básico. 3A. LA MALLA CURRICULAR ESTA ASIGNADA A UNA PETICIÓN DE HOMOLOGACIÓN. <ol style="list-style-type: none">1. El sistema presenta un mensaje (“No se puede borrar esta malla curricular”) en la pantalla “Eliminar Área”.2. El Administrador de Área elige la opción <Cancelar> en la pantalla “Eliminar Malla Curricular”3. Continúa con el paso 4 del curso básico.	

DIAGRAMA DE SECUENCIA

Fragmento: Eliminar Malla Curricular

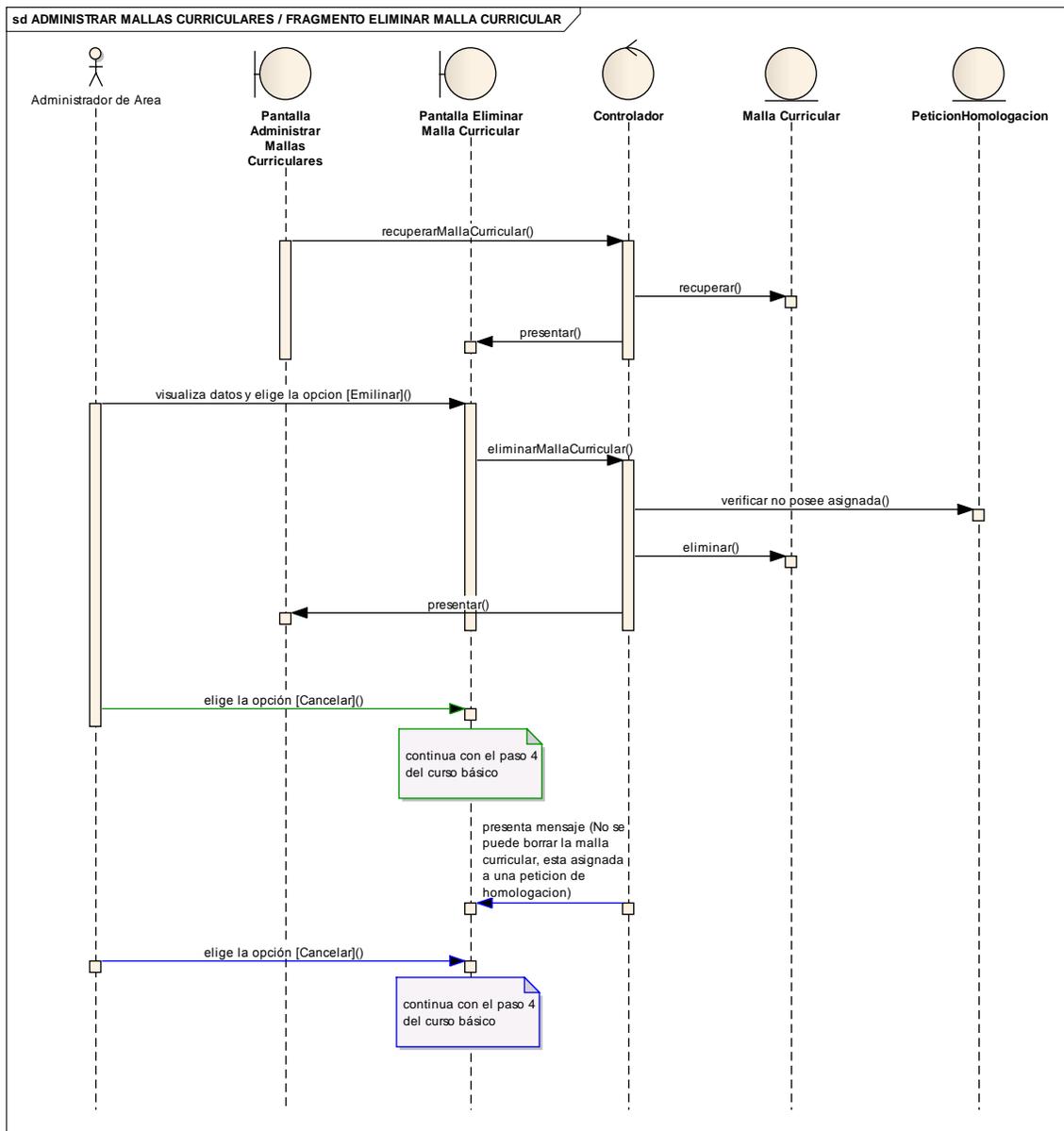


Figura 55. Diagrama de secuencia eliminar malla curricular

PANTALLA ESTRUCTURA PENSUM

The screenshot displays the 'Homologa' web application interface. At the top, there is a green header with the 'Homologa' logo and the user information 'Usuario: administrador12' and 'Cerrar sesión'. Below the header is a navigation menu with links: Inicio, Universidad, Area, Carrera, Malla Curricular, Usuarios, and Mi perfil. The main content area is divided into two sections. The first section, 'Buscar Unidad', features a search bar labeled 'Nombre' and a 'Buscar' button. The second section, 'Lista de Unidades', contains a table of results. The table has columns for 'Id', 'Nombre', and 'Acciones'. There are four rows of data, each with 'Eliminar' and 'Editar' buttons. A 'Nueva Unidad' button is located at the top right of the results section. At the bottom of the interface, there is a green footer with the logo of the Universidad Nacional de Loja.

Buscar Unidad

Nombre

Buscar

Lista de Unidades

Resultados Nueva Unidad

Id	Nombre	Acciones
36	Programacion Punto Net 1	Eliminar Editar
37	Analisis y Diseno Orientado a Objetos 1	Eliminar Editar
38	Base de Datos	Eliminar Editar
39	Etica Social y Profesional	Eliminar Editar

Malla Curricular

Universidad Nacional de Loja

Figura 56. Pantalla estructura pensum

FRAGMENTO: Estructura Pensum

**TABLA XXXVI
FRAGMENTO: ESTRUCTURA PENSUM**

Fragmento: Estructura Pensum	
CASO DE USO REFERIDO:	ADMINISTRAR MALLAS CURRICULARES
Curso Básico	
1. El sistema carga y presenta la lista de unidades del nivel de la malla curricular en la pantalla “Estructura Pensum”	
2. El Administrador de Área selecciona la unidad a modificar y elige la opción <Editar> en la pantalla “Estructura Pensum”	
3. El sistema recupera los datos de la unidad seleccionada y los presenta en la pantalla “Editar Unidad o Materia”	
4. El sistema llama al Fragmento “ Editar Unidad o Materia ”	
5. Finaliza Caso de Uso	
Cursos Alternos	
2A. El administrador de área desea crear una nueva unidad	
1. El Administrador de Área elige la opción <Nueva Unidad> en la pantalla “Estructura Pensum”	
2. El sistema crea una nueva unidad y presenta la pantalla “Editar Unidad o Materia”	
3. Continúa con el paso 4 del curso básico.	
2B. El administrador de área desea eliminar un unidad	
1. El Administrador de Área elige la opción <Eliminar> en la pantalla “Estructura Pensum”	
2. El sistema llama al fragmento “ Eliminar Unidad ”	
3. Continúa con el paso 5 del curso básico.	
2C. El administrador de área desea regresar a la malla curricular	
1. El Administrador de Área elige la opción <Malla Curricular> en la pantalla “Estructura Pensum”	
2. Continúa con el paso 5 del curso básico.	

DIAGRAMA DE SECUENCIA

Fragmento: Estructura Pensum

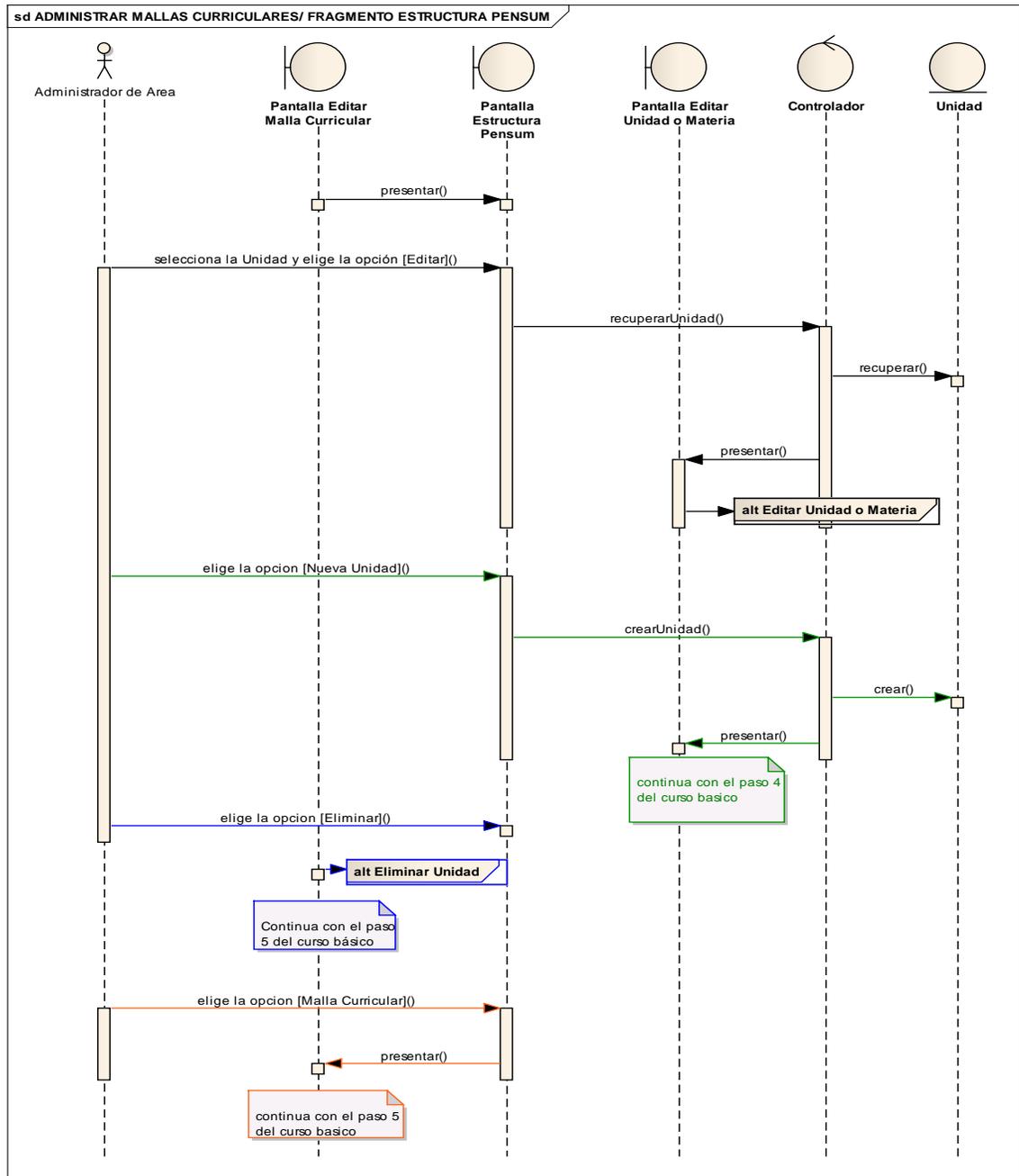


Figura 57. Diagrama de secuencia estructura pensum

PANTALLA EDITAR UNIDAD O MATERIA

The screenshot shows the 'Homologa' system interface. At the top, there is a navigation menu with options: Inicio, Universidad, Área, Carrera, Malla Curricular, Usuarios, and MI perfil. The user is logged in as 'administrador12' with a 'Cerrar sesión' link. The main content area is titled 'Unidad' and contains a form with the following fields:

- Nivel: 5
- Nombre: Gestion Empresarial
- Numerohoras: 64
- Numerocredito: 4
- Esobligatoria:

Below the form is a table titled 'Contenidos' with an 'Agregar' button. The table has two columns: 'Nombre' and 'Acciones'.

Nombre	Acciones
Empresa	Quitar
Subsistema Comercial	Quitar
Funcion Marketing	Quitar
Subsistema Financiero	Quitar

At the bottom of the form, there are 'Guardar' and 'Cancelar' buttons.

Figura 58. Pantalla editar unidad o materia

FRAGMENTO: Editar Unidad O Materia

TABLA XXXVII
FRAGMENTO: EDITAR UNIDAD O MATERIA

Fragmento: Editar Unidad O Materia	
CASO DE USO REFERIDO:	ADMINISTRAR MALLAS CURRICULARES/ FRAGMENTO ESTRUCTURA PENSUM
Curso Básico	
1. El Administrador de Área ingresa los datos requeridos, Nombre, Numero Hora, Numero Créditos y obligatoria en la pantalla "Editar Unidad o Materia"	
2. El Administrador de Área elige la opción <Guardar> en la pantalla "Editar Unidad o Materia"	
3. El sistema valida que los campos requeridos tengan datos ingresados.	

4. El sistema actualiza la lista de Contenidos de la Unidad
5. El sistema guarda la Unidad
6. El caso de uso finaliza
Cursos Alternos
2A. El administrador de área desea agregar un nuevo contenido a la unidad
<ol style="list-style-type: none"> 1. El Administrador de Área elige la opción <Agregar> en la pantalla “Editar Unidad o Materia” 2. El sistema agrega el campo vacío de Nombre en la lista de Contenidos en la pantalla “Editar Unidad o Materia” 3. El Administrador de Área ingresa el Nombre del contenido en la lista de Contenidos en la pantalla editar “Editar Unidad o Materia” 4. Continúa con el paso 2 del curso básico.
2B. El administrador de área desea editar el nombre del contenido
<ol style="list-style-type: none"> 1. El Administrador de Área elige y hace clic en el Nombre del Contenido que desea editar, en la lista de Contenidos en la pantalla “Editar Unidad o Materia” 2. El Administrador de Área ingresa el Nombre del Contenido en la lista de Contenidos en la pantalla editar “Editar Unidad o Materia” 3. Continúa con el paso 2 del curso básico
2C. El administrador de área desea quitar un contenido de la unidad
<ol style="list-style-type: none"> 1. El Administrador de Área selecciona un contenido y elige la opción <Quitar> en la pantalla “Editar Unidad o Materia” 2. El sistema quita el contenido de la lista de Contenidos de la Unidad en la pantalla “Editar Unidad o Materia” 3. Continúa con el paso 2 del curso básico
2D. El administrador de área desea cancelar editar unidad o materia
<ol style="list-style-type: none"> 1. El Administrador de Área elige la opción <Cancelar> en la pantalla “Editar Unidad o Materia” 2. Continúa con el paso 6 del curso básico.

5A. EL SISTEMA NO GUARDA LA UNIDAD

1. El sistema no guarda la unidad
2. El sistema presenta los campos con error con asterisco * y con un mensaje en los campos que no pueden estar en blanco en la pantalla "Editar Unidad o Materia"
3. Continúa con el paso 1 del curso básico

DIAGRAMA DE SECUENCIA

Fragmento: Editar Unidad o Materia

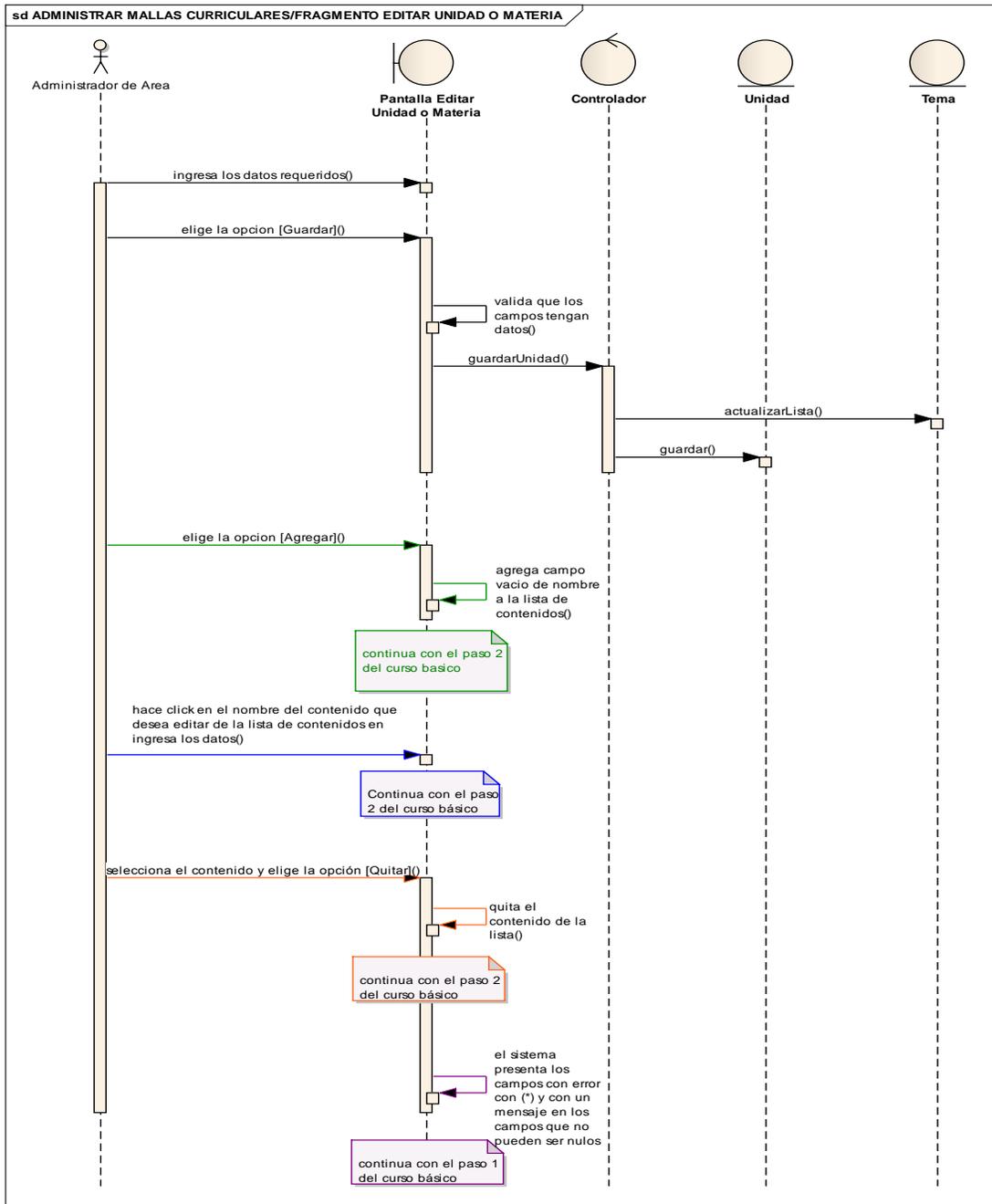


Figura 59. Diagrama de secuencia editar unidad o materia

PANTALLA ELIMINAR UNIDAD

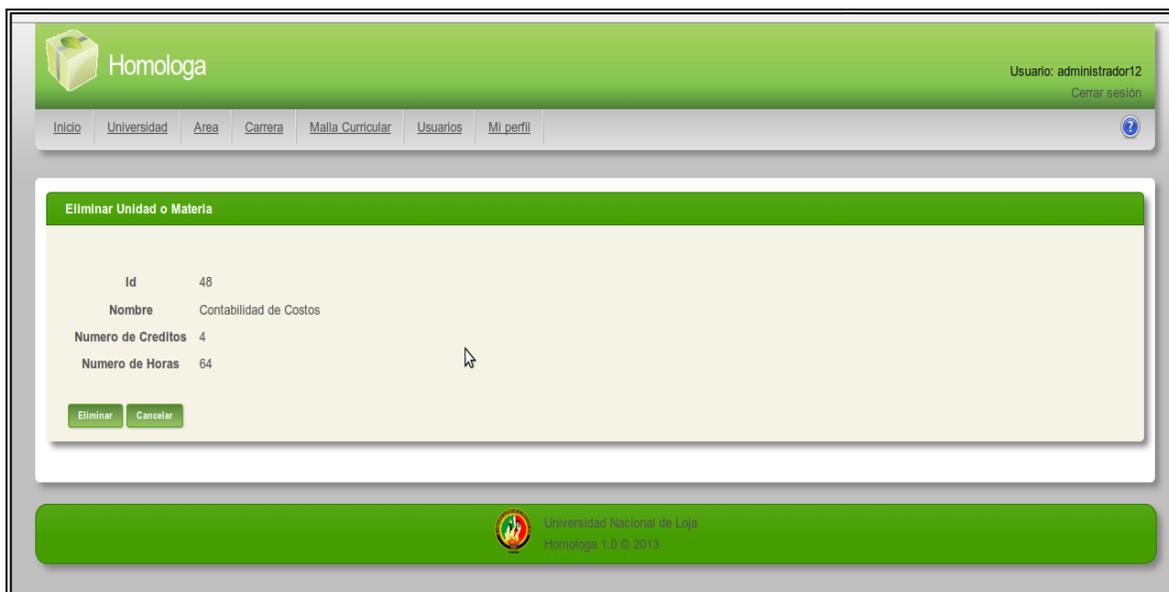


Figura 60. Pantalla eliminar unidad

FRAGMENTO: Eliminar Unidad

TABLA XXXVIII
FRAGMENTO: ELIMINAR UNIDAD

Fragmento: Eliminar Unidad	
CASO DE USO REFERIDO:	ADMINISTRAR MALLAS CURRICULARES/ FRAGMENTO ESTRUCTURA PENSUM
Curso Básico	
1. El sistema recupera la unidad seleccionada y presenta los datos en la pantalla "Eliminar Unidad o Materia"	
2. El Administrador de Área visualiza datos de la unidad y elige la opción <Eliminar> en la pantalla "Eliminar Unidad o Materia"	
3. El sistema elimina la unidad	
4. El sistema presenta la pantalla "Estructura Pensum"	

5. El caso de uso finaliza

Cursos Alternos

2A. El administrador de área desea cancelar eliminar unidad

1. El Administrador de Área elige la opción <Cancelar> en la pantalla “Eliminar Unidad o Materia”
2. Continúa con el paso 4 del curso básico.

DIAGRAMA DE SECUENCIA

Fragmento: Eliminar Unidad

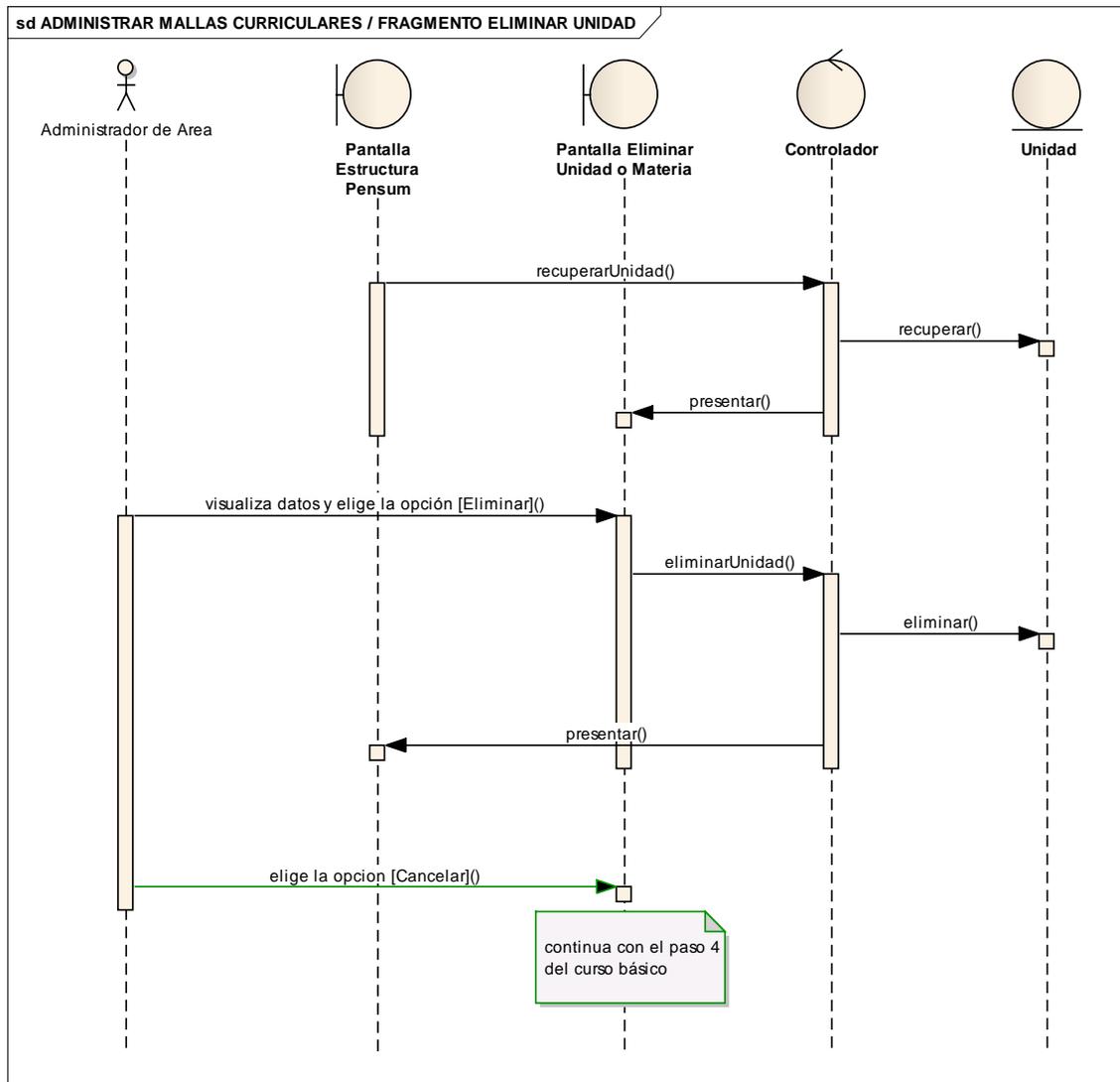


Figura 61. Diagrama de secuencia eliminar unidad

PANTALLA ADMINISTRAR CARRERAS

Homologa

Usuario: administrador12
Cerrar sesión

Inicio Universidad Area Carrera Malla Curricular Usuarios Mi perfil

Buscar Carrera

Nombre

Buscar

Lista de Carreras

Resultados Nueva Carrera

Id	Codigo	Nombre	Universidad	Acciones
8	7	Informatica	4 Universidad Nacional Tecnica Particular de Loja	Eliminar Editar
14	2	Ingenieria en Sistemas Informaticos y Computacion	4 Universidad Nacional Tecnica Particular de Loja	Eliminar Editar
12	2	Telecomunicacion	4 Universidad Nacional Tecnica Particular de Loja	Eliminar Editar

Universidad Nacional de Loja
Homologa 1.0.0 2013

Figura 62. Pantalla administrar carreras

PANTALLA EDITAR CARRERA

Homologa

Usuario: administrador12
Cerrar sesión

Inicio Universidad Area Carrera Malla Curricular Usuarios Mi perfil

Carrera

Universidad Universidad Nacional Tecnica Particular de Loja

Codigo 7

Nombre Informatica

Guardar Cancelar

Universidad Nacional de Loja
Homologa 1.0.0 2013

Figura 63. Pantalla editar carrera

CASO DE USO: Administrar Carreras

**TABLA XXXIX
CASO DE USO: ADMINISTRAR CARRERAS**

Caso de Uso:	Caso De Uso: Administrar Carreras
Precondición:	El Administrador de área debe haber ingresado al sistema
Post Condición:	Carreras quedan modificadas de alguna manera.
Referencia a Requerimientos:	
Curso Básico	
1. El administrador de Área elige la opción < Carrera> en la pantalla “Menú”	
2. El sistema carga y presenta la pantalla “Administrar Carreras”	
3. El Administrador de Área selecciona una carrera de la lista de carreras y elige la opción <Editar> en la pantalla “Administrar Carreras”	
4. El sistema recupera la carrera seleccionada y la presenta en la pantalla “Editar Carrera”	
5. El Administrador de Área selecciona la universidad e ingresa Código y Nombre de la carrera en la pantalla “Editar Carrera”	
6. El Administrador de Área elige la opción <Guardar> en la pantalla “Editar Carrera”	
7. El sistema valida que los campos requeridos no estén en blanco.	
8. El sistema guarda la carrera	
9. El caso de uso finaliza	
Cursos Alternos	
3A. El administrador de area desea crear una nueva carrera	
1. El Administrador de Área elige la opción <Nueva Carrera> en la pantalla “Administrar Carreras”	
2. El sistema crea una nueva carrera y presenta la pantalla “Editar Carrera”	
3. Continúa con el paso 5 del curso básico	

<p>3B. El administrador de area desea buscar una carrera</p> <ol style="list-style-type: none">1. El Administrador de Área ingresa el nombre de la carrera a buscar y elige la opción <Buscar> en la pantalla “Administrar Carreras”2. El sistema busca la carrera y la presenta en la pantalla “Administrar Carreras”3. Continúa con el paso 9 del curso básico
<p>3C. El administrador de area desea eliminar una unversidad</p> <ol style="list-style-type: none">1. El Administrador de Área elige la opción <Eliminar> en la pantalla “Administrar Carreras”2. El sistema llama al fragmento “Eliminar Carrera”3. Continúa con el paso 9 del curso básico
<p>6A. El administrador no desea guardar la carrera y elige la opcion cancelar</p> <ol style="list-style-type: none">1. El Administrador de Área elige la opción <Cancelar> en la pantalla “Editar Carrera”2. continua con el paso 2 del curso básico
<p>8A. El sistema no guarda la carrera.</p> <ol style="list-style-type: none">1. El sistema no guarda la carrera2. El sistema presenta los campos con error con asterisco (*) y con un mensaje en las campos que no pueden ser nulos en la pantalla “Editar Carrera”3. continua con el paso 5 del curso básico

DIAGRAMA DE SECUENCIA

Caso De Uso: Administrar Carreras

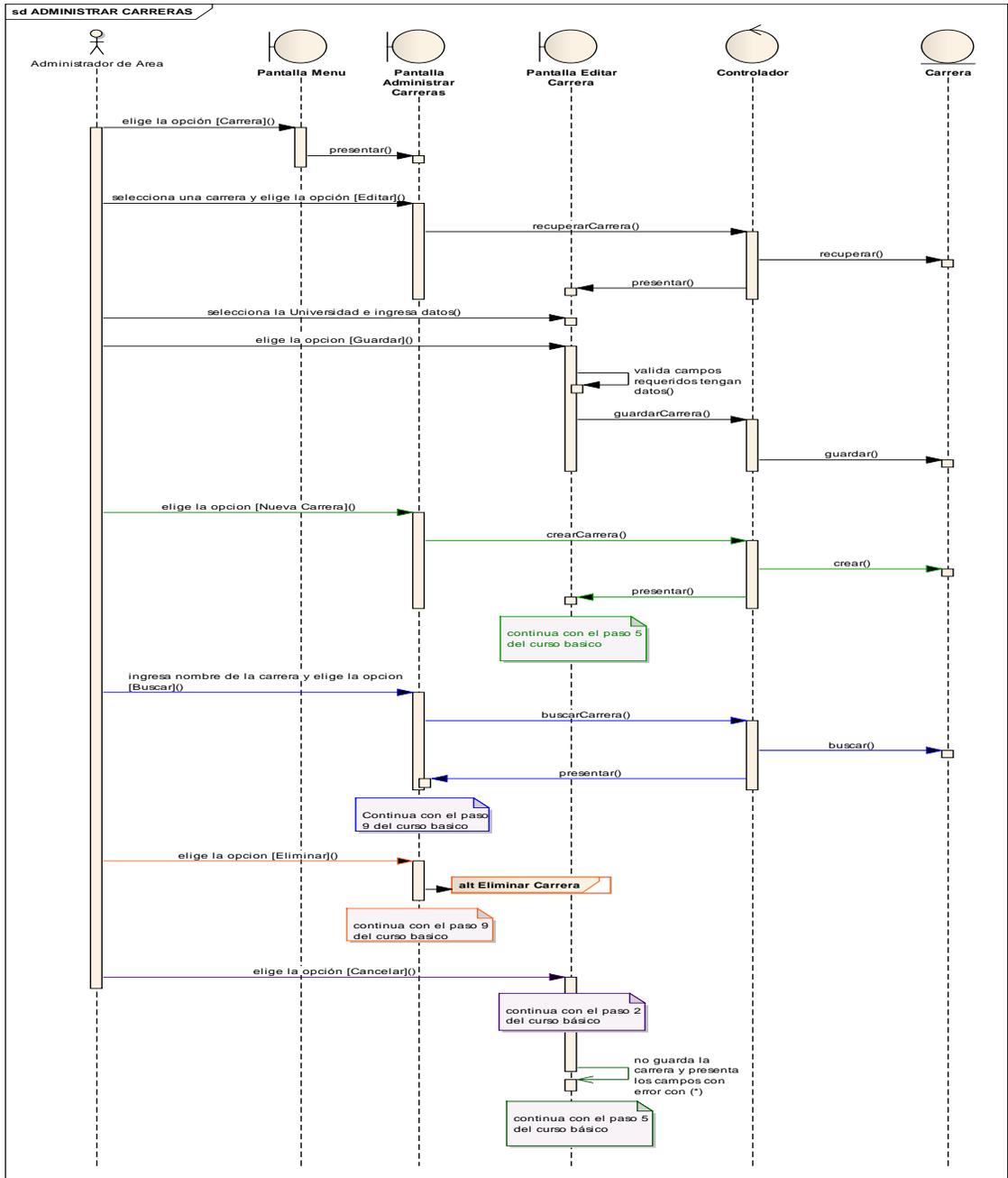


Figura 64. Diagrama de secuencia administrar carreras

PANTALLA ELIMINAR CARRERA

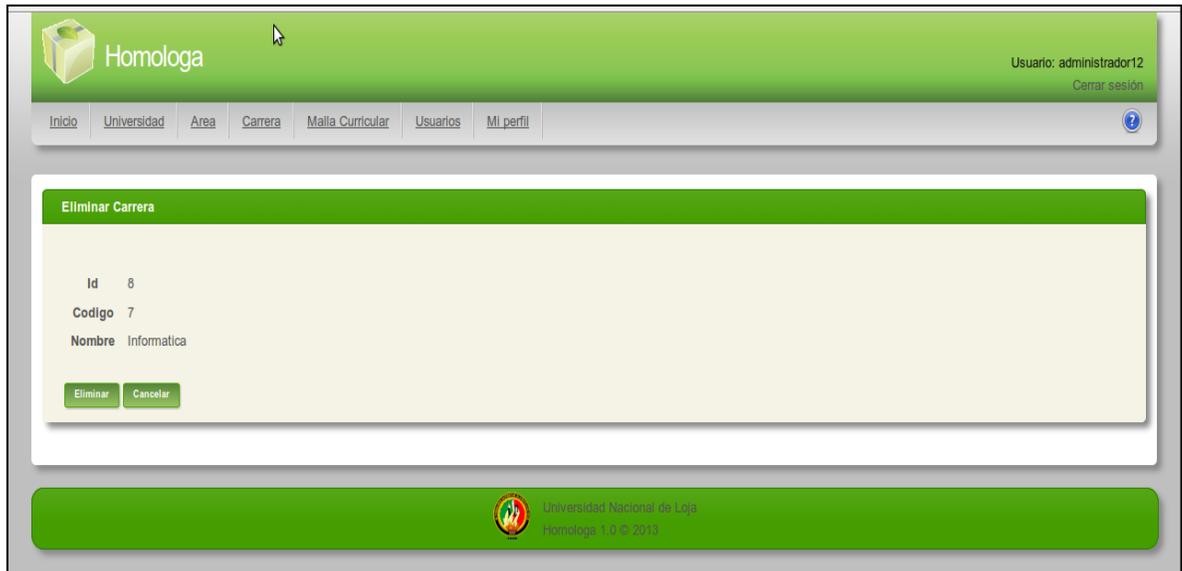


Figura 65. Pantalla eliminar carrera

FRAGMENTO: Eliminar Carrera

TABLA XL
FRAGMENTO: ELIMINAR CARRERA
FRAGMENTO: ELIMINAR CARRERA

CASO DE USO REFERIDO:	Administrar Carreras.
Curso Básico	
1. El sistema recupera los datos de la carrera y los presenta en la pantalla "Eliminar Carrera"	
2. El administrador de Área visualiza los datos de la carrera y elige la opción <Eliminar> en la pantalla "Eliminar Carrera"	

3. El sistema verifica si la carrera no posee peticiones de homologación asignadas y elimina la carrera seleccionada
4. El sistema presenta la pantalla “Administrar Carreras”
5. El caso de uso finaliza
Cursos Alternos
2A. La administrador de area desea cancelar eliminar carrera.
<ol style="list-style-type: none"> 1. El administrador de Área visualiza los datos de la universidad y elige la opción <Cancelar> en la pantalla “Eliminar Carrera” 2. Continúa con el paso 4 del curso básico.
3A. La carrera tiene petición de homologación asignada.
<ol style="list-style-type: none"> 1. El sistema presenta un mensaje (“carrera tiene peticiones de homologación asignadas, no se puede borrar”) en la pantalla “Administrar Carreras” 2. El Administrador de Área elige la opción <Cancelar> en la pantalla “Eliminar Carrera” 3. Continúa con el paso 4 del curso básico

DIAGRAMA DE SECUENCIA

Fragmento: Eliminar Carrera

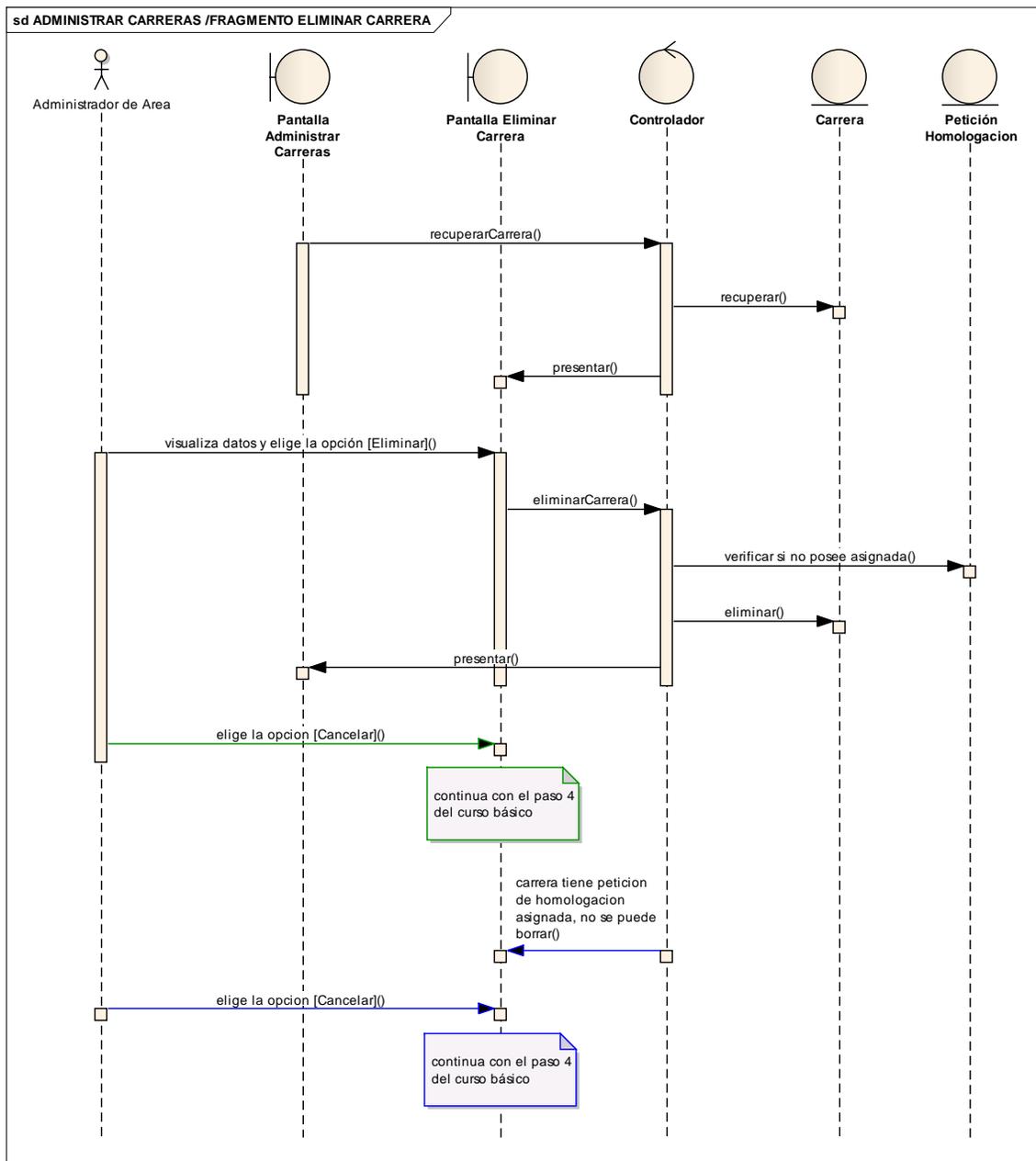


Figura 66. Diagrama de secuencia eliminar carrera

OTROS DIAGRAMAS

Aunque no sean tan comunes se decidió incluir un par de esquemas que pueden aclarar los contenidos y cómo será la interfaz de usuario. Este tipo de diagramas no forman parte de UML:

Vista de interfaz de usuario

Para facilitar la usabilidad se ha decidido mantener una estructura que sea común en las aplicaciones web de este tipo, ya que así el usuario encontrara familiar el entorno y lo utilizara de forma intuitiva.

En la siguiente figura se muestra un esbozo lógico de la interfaz de usuario Web del portal, que está dividida en: Cabecera donde ira el logo y el título, menú, sub-menú, contenido y pie de página. El menú será el eje central de navegación y se ha decidido colocarlo en la parte superior para dejar más amplitud al contenido ya que tendrá varias secciones con listas.

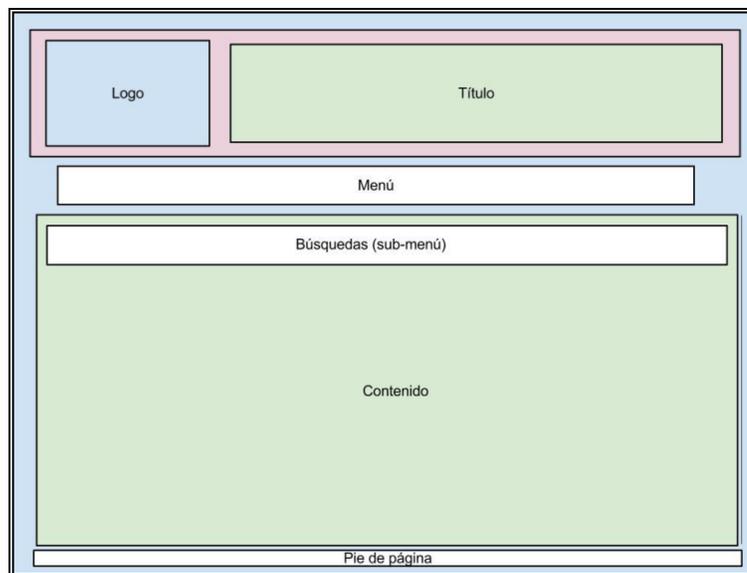


Figura 67. Esquema de la interfaz de usuario

g. Discusión

1. Desarrollo de la propuesta alternativa

- Diseñar una solución acorde a los requerimientos, y documentarla mediante las regulaciones de la Metodología ICONIX.

El presente objetivo pretende encontrar una forma adecuada de encontrar los requerimientos de la aplicación y de realizar los procesos posteriores de análisis y diseño del sistema para solucionar el tema de desarrollo de la herramienta. Al respecto existen varias metodologías que cubren el proceso de diversas formas, desde las que proponen una definición semiformal del sistema como R.U.P. con O.C.L. (Object Constraint Language) hasta las que sugieren el desarrollo del código casi directamente como Extreme Programming. Habiendo revisado varias de las metodologías se puede mencionar que la estricta formalidad y la existencia de roles específicos para cada tarea de R.U.P. es deseable cuando el tamaño y presupuesto del proyecto es lo bastante extenso como para que los roles sean cubiertos por personas específicas, en el caso del presente proyecto, existe la restricción en el número de desarrolladoras involucradas, que serían las autoras del mismo. El desarrollo en R.U.P. por lo tanto generaría un conjunto de documentación compleja y creada con el único objetivo de satisfacer los niveles de formalidad solicitados por dicha metodología más que de clarificar los procesos de análisis y diseño.

De otro lado, metodologías como Extreme Programming (XP), sugieren el uso específico de código y de ciertas técnicas de programación específicas como la programación en parejas, la integración continua y el desarrollo basado en pruebas (TDD). Sin embargo nuestro criterio como autoras, es que un conjunto de modelos específicos facilitan la comunicación entre los usuarios y los desarrolladores, es muy poco lo que un usuario final

podría comprender de un conjunto de código, en particular la discusión de un modelo conceptual y un modelo de casos de uso es bastante ilustrativo y simple de comprender para los usuarios con quienes se discuten los temas de análisis del sistema. Además y por ejemplificar, la disponibilidad de usuarios para la práctica del desarrollo en parejas era poco probable dado que no se podía contar con los usuarios a tiempo completo para cumplir con la forma de trabajo de XP.

Ante esta perspectiva, ICONIX se muestra como una alternativa que permite un desarrollo documentado como para proveer artefactos no excesivos pero suficientes para justificar adecuadamente el desarrollo. El modelo sugerido por ICONIX implica el desarrollo de un subconjunto de modelos de aquellos sugeridos por R.U.P., que permiten un desarrollo ágil del software y que a su vez evitan la carencia de documentación para el incremento de funcionalidad en mejoras o el desarrollo de nuevas características del sistema a futuro.

Esta fue la razón principal por la que en el desarrollo del proyecto se utilizó la metodología ICONIX, la cual permitió en la primera fase de análisis partir con la determinación de los requerimientos del sistema y luego diseñar una solución acorde a los mismos. La documentación generada se encuentra como parte del informe presentado.

- Construir una base de datos que permita registrar, de forma segura, ordenada y correcta los documentos que se presenten para homologación en las carreras del AEIRNNR.

El análisis de la consecución de este objetivo tiene dos aristas, en primer lugar la selección del motor de base de datos y luego la estrategia escogida para la persistencia de la información.

En cuanto a la selección del motor de base de datos, debe mencionarse que el mes de Abril de 2008, la presidencia de la República expidió el decreto 1014 acerca del uso de software libre en Ecuador, la cual incidió mucho de la decisión acerca de que motor de base de datos utilizar. Las alternativas Open Source en este ítem son claramente MySQL y PostgreSQL ya que son las más utilizadas y por ende las que mayor soporte poseen, ambas tienen muy buena calidad y son productos probados en proyectos de toda dimensión. Sin embargo, MySQL se seleccionó sobre PostgreSQL por el ecosistema casi *de facto* disponible en el ámbito de PHP llamado LAMP (Linux, Apache, MySQL, PHP), un entorno completo de software libre para el desarrollo de aplicaciones. Aun así, al usar un ORM la decisión de la base de datos a usar resulta poco trascendente ya que si se usa adecuadamente Doctrine, un cambio de motor de base de datos se limita al cambio del archivo **parameters.ini**

En cuanto a la estrategia para el manejo de persistencia, las alternativas a considerar eran usar SQL a través de consultas directas a la base de datos o alguno de los ORM's disponibles y con soporte en Symfony 2, a saber Propel y Doctrine. La alternativa más simple podía haber sido el realizar la conexión directamente a la base de datos y trabajar con SQL dado el conocimiento de dicho lenguaje ya que formó parte del currículum de la carrera, sin embargo, la búsqueda por una nueva forma de lograr los mismos resultados de gestión de persistencia a través de diferentes mecanismos, hicieron que se utilice un ORM. La selección de Doctrine fue simplemente el resultado de que Symfony 2 sugiere trabajar con ese ORM en lugar de Propel debido a una mejor integración con el framework. A la larga, la decisión, además de favorecer el aprendizaje de una nueva forma de interactuar con la base de datos a través de un ORM, resultó también en el incremento de la productividad de desarrollo, puesto que la organización de los repositorios Symfony para cada clase conceptual facilitó la programación de las consultas y su ubicación en lugares bien definidos por el framework. Los detalles sobre la

generación automática de la base de datos a partir del mapeo (DDL), la conexión a la base de datos, la generación del SQL para la manipulación de información y la creación de transacciones se facilitaron a través del uso del *EntityManager* provisto para dichas tareas.

- Construir los componentes necesarios para la administración de los documentos requeridos en el trámite de la homologación.

Puesto que la construcción de la aplicación implicó la aplicación de un proceso de desarrollo, era necesaria la selección de una herramienta de modelado. Entre las herramientas que facilitaban la tarea de construcción de los diagramas estaban Magic Draw y Enterprise Architect, ambas herramientas con muy buenas prestaciones, sin embargo la decisión de usar Enterprise Architect, vino determinada por la existencia de un plugin específico para ICONIX existente en esta herramienta para el instante en el que inició el desarrollo de la aplicación.

En lo referente a IDEs, se buscó entornos de desarrollo libres. Se revisó tanto Eclipse como Netbeans como potenciales candidatos para el desarrollo, y se descartaron otras alternativas como editores de PHP con reconocimiento sintáctico debido a que Symfony 2 incluye en la construcción de aplicaciones no solamente PHP sino un conjunto de otros archivos como plantillas twig, yaml, xml, además el proyecto utilizó JavaScript y jQuery para algunas validaciones y controles en vista. Todos los formatos descritos son bien soportados en su reconocimiento sintáctico en ambos entornos, sin embargo Netbeans trae incluido dicho soporte sin necesidad de instalar plugins adicionales. Estas razones fueron las que facilitaron la decisión de usar Netbeans 7.2.

En la construcción también fue necesario el uso de un sistema de control de versiones, el elegido entre CVS, Mercurial y Git fue SVN, debido a la gran cantidad de entornos libres en Internet para subir el proyecto y a la extensa documentación del producto disponible.

Al final, el resultado de la construcción de la aplicación se evidencia con el conjunto de funcionalidad desarrollada para facilitar el proceso de homologación gracias a las herramientas seleccionadas.

- Desarrollar un componente de consulta on-line, que permita a los estudiantes conocer el estado general y los estados transitados del trámite de homologación, minimizando el tiempo de respuesta.

Parte de la funcionalidad desarrollada en el sistema permite que cualquier usuario pueda crear una cuenta vía web, asignándole el rol de estudiante automáticamente para solicitar una homologación. Esta petición llega directamente a la secretaría del área y su trámite empieza cuando la secretaria recibe los documentos que acrediten dicha solicitud.

A partir de ese momento el solicitante puede ver el estado de su trámite a través de una máquina de estados programada en la aplicación, la cual va avanzando a medida que el despacho de la misma pasa por los diferentes responsables asignados. De esta manera el solicitante puede, sin necesidad de acercarse físicamente a las dependencias universitarias verificar cuando su petición ha sido finalmente despachada.

2. Valoración técnica económica ambiental

La aplicación ha sido posible desarrollar gracias a que la mayoría de herramientas y utilidades de software necesarios son de libre distribución, sin ninguna restricción, a continuación se detalla los recursos humanos, económicos, materiales, técnicos y tecnológicos con sus costes respectivos.

Recursos Humanos

TABLA XLI
DISCUSIÓN-VALORACIÓN TÉCNICA-RECURSOS HUMANOS

Recursos Humanos	Horas	Costo Hora(\$)	Costo Total(\$)
Janneth Quizhpe	2800	\$8.00	\$22400.00
Silvana Vélez	2800	\$8.00	\$22400.00
Director de Tesis	60	\$8.00	480

Recursos Económicos

TABLA XLII
DISCUSIÓN-VALORACIÓN TÉCNICA-RECURSOS ECONÓMICOS

Recursos Económicos	Horas	Costo Hora(\$)	Costo Total(\$)
Internet	2100	0.80	1680.00
Transporte	50	1.00	50.00
Energía eléctrica	2160	0.05 (Kw/hora)	108.00

Recursos Materiales

TABLA XLIII
DISCUSIÓN-VALORACIÓN TÉCNICA-RECURSOS MATERIALES

Recursos Materiales	Cantidad	Costo Unitario (\$)	Costo Total(\$)
Papel (resmas) INEN A4	5	4.00	20.00
Tinta de impresora (cartuchos)	4	20.00	40.00
Flash Memory	2	15.00	30.00
Anillados	6	1.50	9.00
Empastados	9	6.00	42.00
Copias	300	0.02	6.00

Recursos Técnicos y Tecnológicos

TABLA XLIV
DISCUSIÓN-VALORACIÓN TÉCNICA-RECURSOS TÉCNICOS Y TECNOLÓGICOS

Recursos Técnicos y Tecnológicos	Costo(\$)
Hosting y Dominio (1año)	60.00
2 Computador Portátil	1500.00(depreciación de la computadora, horas, total)
NetBeans IDE 7.1	0.00
PHP 5.3.2	0.00
My SQL 5.1	0.00
Enterprise Architect 3.5 Licencia: demo	0.00
Symfony2	0.00
Doctrine2	0.00
Git	0.00
Twig	0.00

Resumen del Presupuesto

TABLA XLV
DISCUSIÓN-VALORACIÓN TÉCNICA-RESUMEN DEL PRESUPUESTO

Resumen del Presupuesto	Costo Total(\$)
Recursos Humanos	45280.00
Recursos Económicos	1838.00
Recursos Materiales	138.00
Recursos Técnicos y Tecnológicos	1560.00
TOTAL	48816.00

h. Conclusiones

Al finalizar el proyecto se ha llegado a las siguientes conclusiones:

- En la construcción y diseño de un sistema, la metodología ICONIX facilita la comprensión del diseño, lo que permite un desarrollo ágil
- La base de datos resultado del desarrollo de la aplicación, permite mantener un registro de la información generada por el proceso de homologación. Para ello el motor MySQL cumple adecuadamente las expectativas del diseño. Sin embargo, el framework ORM Doctrine es totalmente agnóstico del motor de base de datos y permite generar un modelo físico de datos partiendo de las clases PHP que representan el modelo conceptual y de un conjunto de anotaciones que representan los metadatos de mapeo.
- El uso del Framework Symfony2 facilitó el desarrollo de la aplicación debido a los diversos componentes y aspectos provistos en el cómo las plantillas twig que permiten representar fácilmente una vista y separarla del código PHP, la inclusión de controladores y clases que representan el modelo.
- El proyecto generado en Symfony 2 mantiene una estructura que respeta la arquitectura MVC. Además su estructura modular permite rehusar funcionalidad como la existente en FOSUserBundle, un componente externo que realiza la gestión de seguridades de la aplicación.
- La curva de aprendizaje de Symfony 2 como framework de desarrollo es alta, sin embargo una vez comprendidos los conceptos usados resulta beneficioso debido a que obliga a mantener un orden adecuado en la construcción de los distintos componentes del proyecto.

- Una clara desventaja de Symfony 2 es la diversidad de formatos que se usan para las configuraciones (.xml, php, yaml y anotaciones). Debido a esto, resulta un tanto complejo encontrar documentación y ejemplos que permitan empezar adecuadamente el aprendizaje del framework.

i. Recomendaciones

- Es conveniente utilizar un motor de mapeo objeto relacional como Doctrine, puesto que evita la escritura de sentencias DDL para la creación del modelo físico de datos, ello ahorra tiempo de mantenimiento, puesto que los cambios se centralizan.
- En lo posible se debe utilizar un repositorio de código, puesto que esto facilita el trabajo desde cualquier lugar físico y no requiere que los desarrolladores se encuentren en la misma ubicación.
- Debe aprovecharse las ventajas de las versiones de PHP orientadas a objetos, puesto que muchas aplicaciones construidas en la actualidad continúan con prácticas de programación estructurada, para ello Symfony es una buena opción.
- Dado que el enfoque de la presente investigación se centró en el estudio de nuevos frameworks para el desarrollo de aplicaciones y de su aplicación práctica, el algoritmo de homologación desarrollado es susceptible de mejora, ante ello es posible el planteo de un nuevo tema de tesis que mejore el presente sistema a través de la construcción e integración de un algoritmo inteligente.

j. Bibliografía

SITIOS WEB:

- [1] "ORM, Object-Relational Mapping - I Parte". [Online] Disponible en:
<http://arquitectura-de-software.blogspot.com/2006/05/orm-object-relational-mapping-i-parte.html>
- [2] "Manual de Symfony2". Release 2.0.1 Traducido por Nacho Pacheco August 29, 2011, pp.5 Disponible en:
<http://books.openlibra.com/pdf/symfony2-es.pdf>
- [3] "Ajax". [Online] Disponible en:
<http://desarrollowebaws.com/herramientas/376/ajax/>
- [4] "Comparativa entre los distintos motores ECMAScript" [online] Disponible en:
<http://www.etnassoft.com/2011/05/31/comparativa-entre-los-distintos-motores-ecmascript/>
- [5] "Introducción a la Aplicaciones Web" [online], pp. 3 Disponible en:
<http://www.lsi.us.es/docencia/get.php?id=352>.
- [6] "Los Frameworks De Php Agilizan Tu Trabajo" [online] Disponible en:
<http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>
- [7] "¿Qué es Ajax? - Definición de Ajax" [online] Disponible en:
<http://www.masadelante.com/faqs/ajax>

- [8] “Conceptos básicos de ORM (Object Relational Mapping)”. [online] Disponible en:
[http://topicosjvcarlos.wikispaces.com/file/view/Conceptos+b%C3%A1sicos+de+OR
M.pdf](http://topicosjvcarlos.wikispaces.com/file/view/Conceptos+b%C3%A1sicos+de+ORM.pdf)
- [9] “Manual Twig”. Disponible en:
<http://twig.sensiolabs.org/>

K. Anexos

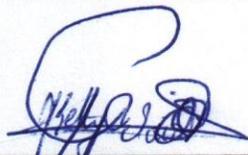
ANEXO 1. REQUERIMIENTOS DEL SISTEMA DE HOMOLOGACION

DOCUMENTO DE REQUERIMIENTOS DEL SISTEMA DE AUTOEVALUACIÓN DEL AREA DE ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES.

CODIGO	FUNCION	CATEGORIA
RQF.01	Permitirá la homologación entre carreras del AEIRNNR de la UNL	Evidente
RQF.02	Permitirá la homologación de las carreras del AEIRNNR con carreras de otras universidades	Evidente
RQF.03	Administrar los Pensum de las diferentes carreras del AEIRNNR	Evidente
RQF.04	Administrar los contenidos de las diferentes unidades de cada módulo de las carreras del AEIRNNR	Evidente
RQF.05	El sistema permitirá realizar la homologación mediante contenidos(temas)	Oculto
RQF.06	El sistema permitirá utilizar un mínimo de contenidos(temas) para realizar la Homologación	Oculto
RQF.07	Permitir que el estudiante ingrese una solicitud de homologación vía internet	Evidente
RQF.08	Permitira visualizar el estado del Proceso de Homologación (Registro de estados) a través de internet	Evidente
RQF.09	Permitirá visualizar el resultado de la homologación vía internet	Evidente

RQF.10	Permitirá Imprimir el resultado de la Homologación	Evidente
RQF.11	El sistema permitirá al usuario(Docente) asignar contenidos(temas) de las unidades de cada carrera a homologar y almacenarlos	Evidente
RQF.12	Permitirá visualizar los porcentajes de la homologación de las carreras	Evidente
RQF.13	Permitirá crear menús personalizados de acuerdo a los roles de los usuarios	Evidente
RQF.14	Permitirá llevar un registro de docentes con sus respectivas homologaciones asignadas	Evidente
RQF.15	Validará el correcto ingreso de la información de los Pensum de cada carrera	Oculto

Los Requerimientos antes expuestos son dados a conocer y aprobados por Miembros de la Comisión Técnica de Evaluación de la UNIVERSIDAD NACIONAL DE LOJA.



Ing. Kelly Palacios
COORDINADORA DE LA
CARRERA DE INGENIERIA EN SISTEMAS



ANEXO 2. Entrevista para la Determinación de Requerimientos

Departamento: Coordinación de la carrera de Ingeniería en Sistemas del Área de Energía Industria y Recursos Naturales no Renovables de la Universidad Nacional de Loja

Entrevistado: Ing. Kelly Palacios

Objetivo: Conocer los pasos que se sigue al realizar un trámite de homologación

Resumen de la Entrevista:

El trámite de homologación que se realiza en el AEIRNNR, es de forma manual, así: cualquier estudiante presenta la siguiente documentación: fotocopias de documentos personales, solicitud de homologación, mallas curriculares, pensum de estudios, planes de contenidos analíticos, etc. en Secretaría de las Coordinaciones de Carreras del Área. Seguidamente, se envía el documento al Coordinador, según corresponda para ser analizado en el Consejo Académico de Carrera, en donde se designa a un docente de la carrera para su respectivo análisis y posterior ubicación o equiparación, emitiendo así un informe al Coordinador de Carrera para la resolución pertinente; mismo que pasa a Secretaria de Coordinación de Carrera para que éste a su vez pase dicho informe a Secretaria de Dirección del Área, y así en Comisión Académica del Área el Director de la misma emita el dictamen definitivo de la ubicación y equiparación solicitada. Pudiendo de esta manera el estudiante solicitante de la homologación tener acceso a dicha resolución.

Con la información indicada anteriormente se pudo comprender los pasos que se sigue para realizar un proceso de homologación, así como los requisitos que se debe cumplir para la misma. Lo que servirá de base para determinar los requerimientos del sistema.

Este presente es dado a conocer a:



Ing. Kelly Palacios
Coordinadora de la Carrera de Ingeniería en Sistemas



ANEXO 3. Entrevista para la Determinación de Requerimientos

Departamento: Docente de la carrera de Ingeniería en Sistemas del Área de Energía Industria y Recursos Naturales no Renovables de la Universidad Nacional de Loja

Entrevistado: Ing. Edison Coronel

Objetivo: Conocer como se realiza una homologación.

Resumen de la Entrevista:

El proceso que realiza el docente para realizar una homologación, es a través de la comparación manual entre los contenidos de cada unidad de la malla curricular de la Carrera a la que va a ser admitido (Carrera Destino) con cada uno de los contenidos de la carrera de la cual proviene el solicitante (Carrera Origen).

El docente Empieza la comparación con todas las unidades que conforman el Modulo1. de la Carrera Destino con las unidades de la Carrera Origen. A través de estas comparaciones el docente va obteniendo el porcentaje de similitud de contenidos en cada unidad. Al finalizar estas comparaciones el docente obtiene un promedio del modulo. Si este promedio tiene un valor mínimo de 80% queda homologado esa modulo y sigue con el mismo proceso para el siguiente modulo

Si el promedio es menor al 80% el docente da por finalizada la homologación y emite un informe de la misma, en el formato establecido, indicando el modulo al cual ha sido homologado.

Con la información indicada anteriormente se pudo comprender el proceso que se sigue para realizar la homologación entre dos carreras. Lo que contribuye a la fase del análisis, la misma que permitirá realizar el diseño del presente proyecto.

Este presente es dado a conocer a:



Ing. Edison Coronel
Docente de la Carrera de Ingeniería en Sistemas

ANEXO 4.



**UNIVERSIDAD NACIONAL DE LOJA
ÁREA DE LA ENERGIA, LAS INDUSTRIAS Y
LOS RECURSOS NATURALES NO RENOVABLES
DIRECCIÓN**

Oficio 170-D-AEIRNNR-UNL
Loja, abril 2 de 2013

Señores:
COORDINADORES DE CARRERA
Ciudad.-

De mi consideración:

En atención a la petición de las señoritas Egresadas de la Carrera de Ingeniería en Sistemas Silvana Vélez y Janeth Quishpe, esta Dirección autoriza, para que las antes mencionadas señoritas realicen las pruebas del anteproyecto de tesis denominado **HOMOLOGA**, utilizando del Framework Symfony.

Con sentimiento de consideración.

Atentamente,
EN LOS TESOROS DE LA SABIDURIA
ESTA LA GLORIFICACION DE LA VIDA

Ing. José Francisco Ochoa Alfaro
DIRECTOR

c.c. Archivo.
JFOA/SUSANA S.-

**Ciudad Universitaria "Guillermo Falconí Espinosa", La Argelia
Telefax :2545-689, Teléfonos:2545-691,2545-688,2545-672
Casilla:Letra "S"
Portal Web:www.unl.edu.ec./areaenergía
Loja-Ecuador**

ANEXO 5.

INFORME ACADÉMICO SOBRE RECONOCIMIENTO DE ESTUDIOS

DATOS REFERENCIALES DEL PETICIONARIO
 NOMBRES Y APELLIDOS: RAMÍREZ FIALLO JOSÉ LUIS
 UNIVERSIDAD DE PROCEDENCIA: UNIVERSIDAD INTERNACIONAL DEL ECUADOR SEDE LOJA
 CARRERA DE PROCEDENCIA: INGENIERIA EN INFORMÁTICA Y MULTIMEDIA
 PERIODOS DE ESTUDIOS DE LA UNIVERSIDAD DE PROCEDENCIA: Septiembre/2011 - Enero/2012

MALLA CURRICULAR DE LA CARRERA A SER ADMITIDO	MALLA CURRICULAR DE LA CARRERA DE PROCEDENCIA	NÚMERO DE CRÉDITOS	CALIFICACIÓN	PORCENTAJE 0,80	RECONOCE		OBSERVACIONES
					SI	NO	
MODULO I							
GEOMETRIA PLANA	LENGUAJE INTEGRAL	3	87/100	10%		X	
PROCESO INVESTIGATIVO	INFORMÁTICA BÁSICA	3	83/100	10%		X	
QUÍMICA	APRENDER A APRENDER	3	91/100	10%		X	
MATEMÁTICAS	MATEMÁTICAS I	3	89/100	70%		X	
FÍSICA	FÍSICA	3	89/100	80%		X	

CONCLUSIÓN.- Luego del análisis de la documentación presentada por el peticionario y de la Malla Curricular de la carrera a ser admitido en la institución; la Carrera de Ingeniería en Sistemas mediante informe Of. N° 022 -CIS-AEIRNR-JUNL, de fecha 29 de Enero de 2013, sugiere se reconozca las asignaturas aprobadas en la Carrera de Ingeniería en Informática y Multimedia, periodo 2011-2012; y se lo ubique al primer módulo de la carrera solicitada.- Consecuentemente y en atención a lo establecido en la "NORMATIVA DE TRANSICIÓN DE CONFORMIDAD CON LA LEY ORGÁNICA DE EDUCACIÓN SUPERIOR Y SU REGLAMENTO, NORMA", norma, "TERCERO" en la que faculta a los Directores de Área asumir en forma transitoria los deberes y atribuciones del Consejo Académico del Área, previstas en el Art. 22 del Estatuto Orgánico; y 52 el Reglamento General de la Universidad; **SE RESUELVE:** convalidar (u homologar) los estudios aprobados en la Carrera de Administración de Ingeniería en Informática y Multimedia de la Universidad Internacional del Ecuador Sede Loja, según la descripción de equivalencias que antecede y se lo ubica al **PRIMER MÓDULO** de la Carrera de Ingeniería en Sistemas de esta **ÁREA**.

Loja, 29 de Enero del 2013

Ing. José Francisco Ochoa Alfaro
DIRECTOR DEL ÁREA

Dra. Enith Salcedo Paladinez,
SECRETARIA

REPORTE DE HOMOLOGACION

DATOS REFERENCIALES
 NOMBRES Y APELLIDOS: RAMIREZ FIALLO JOSE LUIS
 UNIVERSIDAD DE PROCEDENCIA: UNIVERSIDAD INTERNACIONAL DEL
 ECUADOR
 CARRERA DE PROCEDENCIA: INGENIERIA EN INFORMATICA Y MULTIMEDIA

Maña carrera a ser admitido	Maña carrera procedencia	Créditos	Calificación	Porcentaje	Se reconoce	Observaciones
Módulo: 1						
Geometria Plana	Lenguaje Integral	3	87 / 100	10 %	NO	
Proceso Investigativo	Informatica Basica	3	83 / 100	20 %	NO	
Quimica	Aprender a Aprender	3	91 / 100	10 %	NO	
Matematicas	Matematicas I	3	89 / 100	70 %	NO	
Fisica	Fisica	3	88 / 100	80 %	SI	
Proceso Investigativo	Programacion Uno	3	81 / 100	40 %	NO	

ANEXO 6.

Ing. Carlos Samaniego
Coordinador de la Carrera de Ingeniería Electromecánica

Certifico

Que las Egresadas Janneth Alexandra Quizhpe Requelme y Silvana Jimena Vélez Suquilanda, autoras de la Tesis titulada "Sistema De Apoyo al Trámite de Homologación para el A.E.I.R.N.N.R de la U.N.L utilizando el Framework Symfony" han realizado las pruebas de Validación, en el manejo del sistema "Homologa" con los distintos Roles como: Estudiante, Secretaria, Docente, Administrador en la Carrera de Ingeniería Electromecánica

Es todo cuanto puedo certificar en honor a la verdad, facultando a las interesadas hacer uso del presente en lo que estimen conveniente.

Loja, 11 de abril del 2013


Ing. Carlos Samaniego



ANEXO 7.

ENCUESTA DIRIGIDA AL PERSONAL ADMINISTRATIVO DEL AREA DE LA ENERGIA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNL.

La presente encuesta pretende realizar la comprobación y validación del Sistema de Apoyo al Trámite de Homologación. HOMOLOGA, en las diferentes funcionalidades que presta.

1. ACCESIBILIDAD

1.1 ¿Se presentó algún problema al momento de ingresar al sistema con su usuario y contraseña?

SI ()

NO (✓)

PORQUE? SE PUDO INGRESAR CORRECTAMENTE

2. FUNCIONALIDAD

2.1 ¿Tuvo algún problema al realizar alguna de las siguientes actividades?

TAREA	SI	NO
Ingresar al sistema		✓
Modificar Cuenca		✓
Crear y Modificar un Peticionario		✓
Asignar Petición de Homologación algún docente		✓
Visualizar el estado de la petición		✓

2.2 ¿Considera que el sistema le permite realizar estas actividades de manera rápida, eficiente y confiable?

SI (✓)

NO ()

3. DISEÑO Y PRESENTACION

3.1 ¿Considera que la interfaz del sistema es fácil de utilizar?

SI (✓)

NO ()

PORQUE? SE VISUALIZA RAPIDAMENTE LAS OPCIONES

QUE UNO TIENE EN EL MENÚ

GRACIAS POR SU COLABORACION

ANEXO 8.

PRUEBAS DE CAJA NEGRA: CREAR USUARIO

SIMBOLO	SIGNIFICADO
c	Código
n	Nombre
d	Descripción

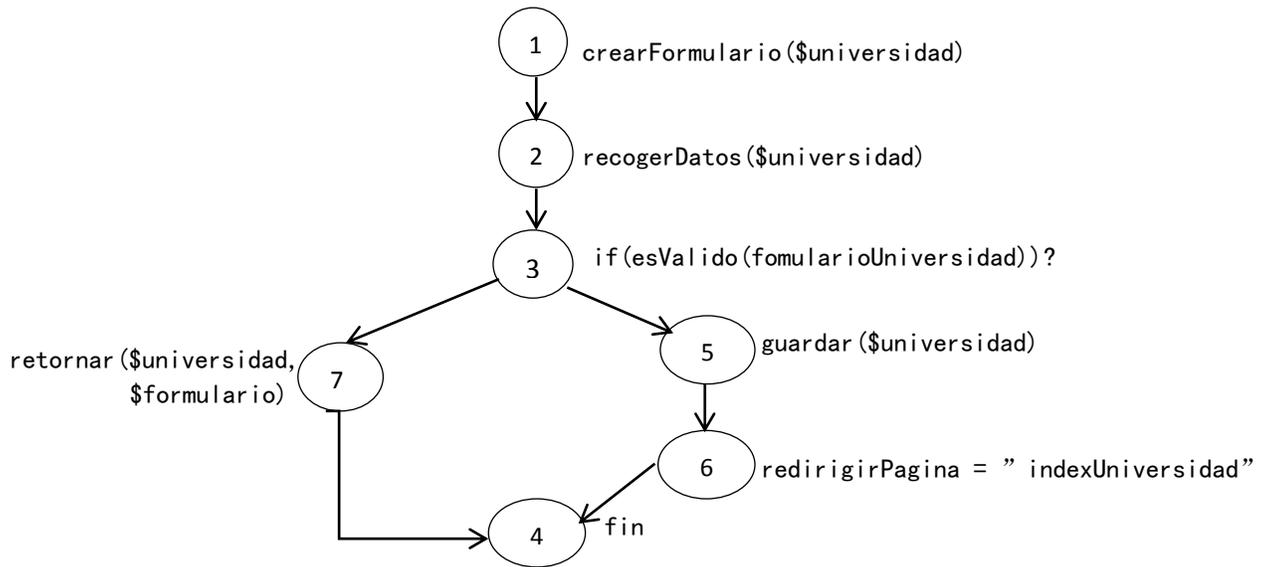
TABLA DE EQUIVALENCIA

Condición entrada	Clases validas	Clases no validas
Código	1. c=alfanumérico	2. c= vacio
Nombre	3. n= alfabético	4. n=vacio
Descripción	5. d= alfanumérico 6. d= vacio	

EJECUCION DE PRUEBAS

No. Caso	Clase de equivalencia	Código	Nombre	Descripción	Resultado
1	1, 3, 5	CUN-001	Universidad Nacional de Loja	UNL	Universidad guardada
2	1, 3, 6	CUT-002	Universidad Técnica Particular de Loja	---	Universidad guardado
3	2, 3,6	---	Universidad Técnica Equinoccial	UTE	Debe ingresar Código de la Universidad
4	1, 4,5	CUP-003	---	ESPE	Debe ingresar Nombre de la Universidad
5	2, 4, 5	---	---	ESPOCH	Debe ingresar Código de la Universidad Debe ingresar Nombre de la Universidad
6	2, 4, 6	---	---	---	Debe ingresar Código de la Universidad Debe ingresar Nombre de la Universidad

PRUEBAS DE CAJA BLANCA: CREAR UNIVERSIDAD



$$V(G) = A - N + 2$$

$$V(G) = 7 - 7 + 2 = 2$$

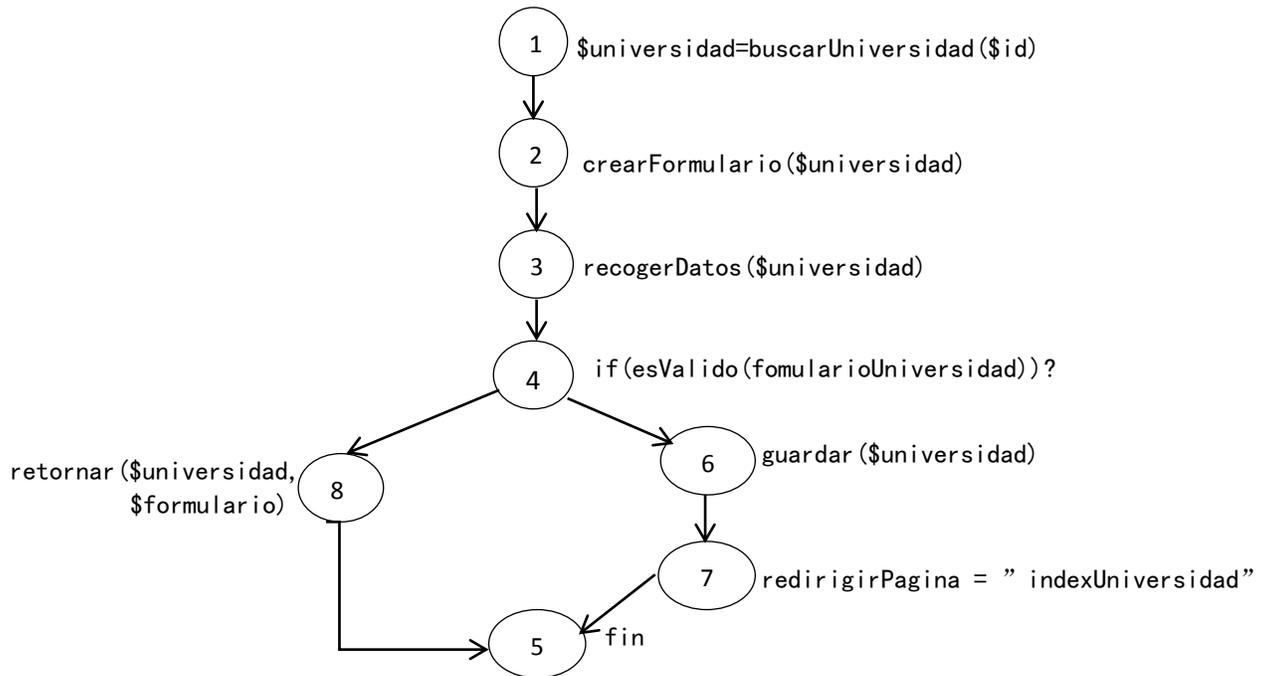
$$V(G) = P + 1$$

$$V(G) = 1 + 1 = 2$$

EJECUCIÓN DE PRUEBAS DE CAJA BLANCA GUARDAR NUEVO UNIVERSIDAD

No. Caso	Camino	Entrada			Resultado
		Código	Nombre	Descripción	
1	1,2,3,5,6,4	CUN-001	Universidad Nacional de Loja	UNL	Universidad guardada
2	1, 2, 3, 7, 4	---	---	UTE	Debe ingresar Código de la Universidad Debe ingresar Nombre de la Universidad

PRUEBAS DE CAJA BLANCA ACTUALIZAR UNIVERSIDAD



$V(G) = A - N + 2$ $V(G) = 8 - 8 + 2 = 2$ $V(G) = P + 1$ $V(G) = 1 + 1 = 2$
--

EJECUCIÓN DE PRUEBAS DE CAJA BLANCA: GUARDAR NUEVO UNIVERSIDAD

No. Caso	Camino	Entrada			Resultado
		Código	Nombre	Descripción	
1	1,2,3,4,6,7, 5	CUN-001	Universidad Nacional de Loja	UNL	Universidad guardada
2	1, 2, 3, 4, 8, 5	---	Universidad Técnica Equinoccial	UTE	Debe ingresar Código de la Universidad

PRUEBAS DE CAJA NEGRA CREAR USUARIO

SIMBOLO	SIGNIFICADO
ti	Tipo de Identificación
ci	Cedula
p	Pasaporte
a	Apellido
n	Nombre
t	Telefono
e	Email
c	Clave
rc	Repita clave
nu	Nombre de Usuario
r	Rol
h	Habilitar

TABLA DE EQUIVALENCIA

Condición entrada	Clases validas	Clases no validas
Tipo de Identificación	1. ti="cedula" 2. ti="pasaporte"	
Cedula	3. ci= numérico 4. ci= 10 dígitos validos 5. ci= vacío y t="pasaporte"	6. ci< 10 dígitos 7. ci= 10 dígitos no validos 8. ci= vacío y ti= "cedula"
Pasaporte	9. p= alfanumérico 10. 10 caracteres ≤ p ≥ 5 caracteres 11. p= vacío y ti="cedula"	12. p< 5 caracteres 13. p= vacío y ti= "pasaporte"
Apellido	14. a= alfanumérico	15. a= vacío
Nombre	16. n= alfanumérico	17. n= vacío
Telefono	18. t= numérico 19. t= vacío	
Email	20. e= alfanumérico formato email	21. e= vacío 22. e= alfanumérico diferente de formato email
Clave	23. c= alfanumérico 24. c ≥ 8 caracteres	25. c= vacío 26. c< 8 caracteres
Repita clave	27. rc=c	28. rc= vacío 29. rc ≠ c
Nombre de Usuario	30. nu= alfanumérico	31. nu= vacío

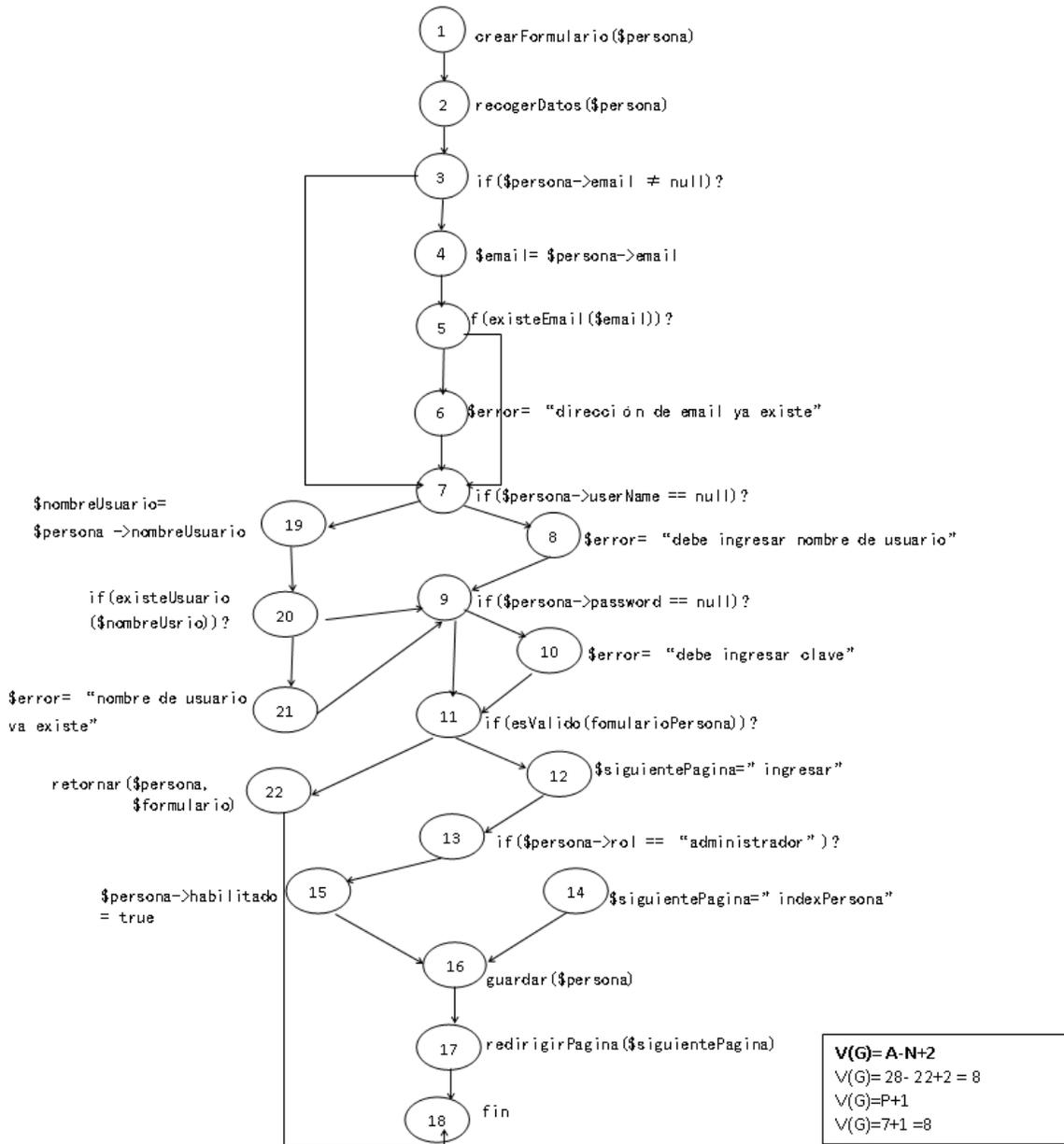
Rol	32. r= "estudiante" 33. r="secretaria" 34. r= "docente" 35. r= "administrador"	
Habilitar	36. h= true 37. h= false	

EJECUCIÓN DE PRUEBAS

No. Caso	Clase de equivalencia	Tipo de identificación	Cedula	Pasaporte	Apellido	Nombre	Telefono	Email	Clave	Repite clave	Nombre de usuario	Rol	Habilitado	Resultado
1	1,3,4,11,14,16,18,20,23,24,27,30,32,36	Cedula	1103780928	-	Buri	María	565845	bm@gmail.com	12345678	12345678	marib	Estudiante	True	Usuario guardado
2	1,3,4,11,14,16,19,20,23,24,27,30,32,36	Cedula	1103780928	-	Buri	María	-	bm@gmail.com	12345678	12345678	marib	Estudiante	True	Usuario guardado
3	1,4,8,11,14,16,19,20,23,24,27,30,32,36	Cedula	-	-	Buri	María	-	bm@gmail.com	12345678	12345678	marib	Estudiante	True	Debe ingresar número de cédula
4	2,5,9,12,14,16,19,20,23,24,27,30,33,36	Pasaporte	-	4567	Buri	María	-	bm@gmail.com	12345678	12345678	marib	Docente	True	El campo pasaporte debe tener mínimo 5 dígitos
5	1,3,4,11,14,16,19,20,25,28,31,34,37	Cedula	1103505507	-	Pérez	Paul	-	bpaz@gmail.com	-	-	-	Docente	False	Debe ingresar nombre de usuario. Debe ingresar Clave
6	2,5,9,14,16,19,	Pasaporte	-	345678	Ríos	Belén	-	rbe@g.co	-	-	buril	Secre	False	Nombre de usuario ya

	20,25,30,33,37	e						m				taria		existe. Debe ingresar Clave
7	2,5,9,14,16,19, 21,23,24,27,31 ,32,37	Pasaport e	-	5789039	Vera	Juan	-	-	12345 678	12345 678	-	Estud iante	False	Debe ingresar email Debe ingresar nombre de usuario
8	2,5,9,14,16,19, 20,25,28,30,32 ,37	Pasaport e	-	5789039	Vera	Juan	-	juanV@g.c om	-	-	vera	Estud iante	False	Debe ingresar clave

PRUEBAS DE CAJA BLANCA CREAR USUARIO

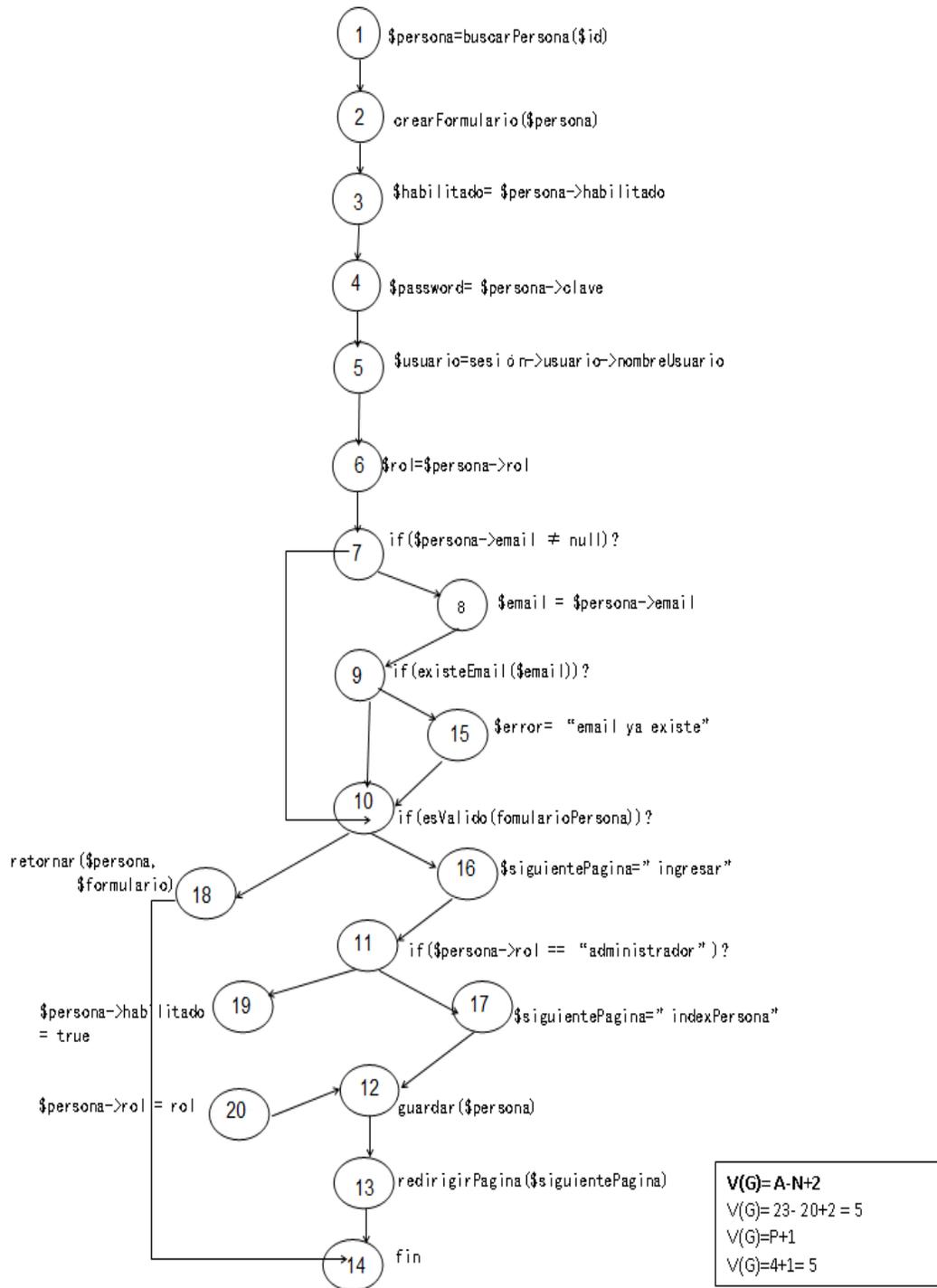


EJECUCIÓN DE PRUEBAS DE CAJA BLANCA GUARDAR NUEVO USUARIO

No. Caso	Camino	ENTRADA												Resultado
		Tipo de identificación	Cedula	Pasaporte	Apellido	Nombre	Telefono	Email	Clave	Repita clave	Nombre de usuario	Rol	Habilitado	
1	1-2-3-7-19-20-9-11-12-13-14-16-17-18	Cedula	1103780928	-	Buri	María	565845	bm@gmail.com	12345678	12345678	marib	Administrador	True	Usuario guardado
2	1-2-3-7-19-20-9-11-12-13-15-16-17-18	Cedula	1103735237	-	Buri	Paz	-	bpaz@gmail.com	12345678	12345678	burip	Estudiante	True	Usuario guardado
3	1-2-3-4-5-6-7-8-9-10-11-22-18	Cedula	1103505507	-	Pérez	Paul	-	bpaz@gmail.com	-	-	-	Docente	False	Debe ingresar nombre de usuario. Debe ingresar Clave Dirección de email ya existe
4	1-2-3-4-5-7-19-20-21-9-10-11-22-18	Pasaporte	-	345678	Ríos	Belén	-	rbe@g.com	-	-	burip	Secretaria	False	Nombre de usuario ya existe. Debe ingresar Clave
5	1-2-3-7-8-9-11-22-18	Pasaporte	-	5789039	Vera	Juan	-	-	12345678	12345678	-	Estudiante	False	Debe ingresar email Debe ingresar nombre de usuario

6	1-2-3-4-5-7-19-20-9-10-11-22-18	Pasaporte	-	5789039	Vera	Juan		juanV@g.com	-	-	veraJ	Estudiante	False	Debe ingresar clave
7	1-2-3-4-5-7-8-9-11-22-18	Pasaporte	-	489208	Quiroz	Raúl	-	quir@g.com	12345678	12345678	-	Estudiante	False	Debe ingresar nombre de usuario
8	1-2-3-4-5-7-19-20-21-9-11-22-18	Pasaporte	-	58902'	Vélez	Betty	-	veBett@gmail.com	12345678	12345678	veraJ	Estudiante	False	Nombre de usuario ya existe

PRUEBAS DE CAJA BLANCA ACTUALIZAR USUARIO



EJECUCIÓN DE PRUEBAS DE CAJA BLANCA GUARDAR NUEVO USUARIO

No. Caso	Camino	ENTRADA												Resultado
		Tipo de identificación	Cedula	Pasaporte	Apellido	Nombre	Telefono	Email	Clave	Repita clave	Nombre de usuario	Rol	Habilitado	
1	1-2-3-4-5-6-7-8-9-11-12-13-14-15-16-17	Cedula	1103780928	-	Buri	María	565845	bm@gmail.com	12345678	12345678	marib	Administrador	True	Usuario guardado
2	1-2-3-4-5-6-7-8-9-10-11-12-13-18-19-15-16-17	Cedula	1103735237	-	Buri	Paz	-	bpaz@gmail.com	12345678	12345678	burip	Estudiante	True	Usuario guardado
3	1-2-3-4-5-6-7-11-20-17	Cedula	1103505507	-	Pérez	Paul	-	-	12345678	12345678	perezpaul	Docente	False	Debe ingresar email
4	1-2-3-4-5-6-7-8-9-11-20-17	Pasaporte	-	345678	Ríos		-	rbe@g.com	12345678	12345678	burip	Secretaria	False	Debe ingresar nombre
5	1-2-3-4-5-6-7-8-9-10-11-20-17	Pasaporte	-	5789039	Vera	Juan	-	rbe@g.com	12345678	12345678	juanV	Estudiante	False	Email ya existe

PRUEBAS DE CAJA NEGRA CREAR PETICIÓN

SIMBOLO	SIGNIFICADO
ti	Tipo de Identificación
ci	Cedula
p	Pasaporte
a	Apellido
n	Nombre
t	Telefono
e	Email
uo	Universidad Origen
nuo	Nombre universidad origen
co	Carrera Origen
nco	Nombre carrera origen
cd	Carrera Destino

TABLA DE EQUIVALENCIA

Condición entrada	Clases validas	Clases no validas
Tipo de Identificación	1. ti="cedula" 2. ti="pasaporte"	
Cedula	3. ci= numérico 4. ci= 10 dígitos validos 5. ci= vacío y t="pasaporte"	6. ci< 10 dígitos 7. ci= 10 dígitos no validos 8. ci= vacío y ti= "cedula"

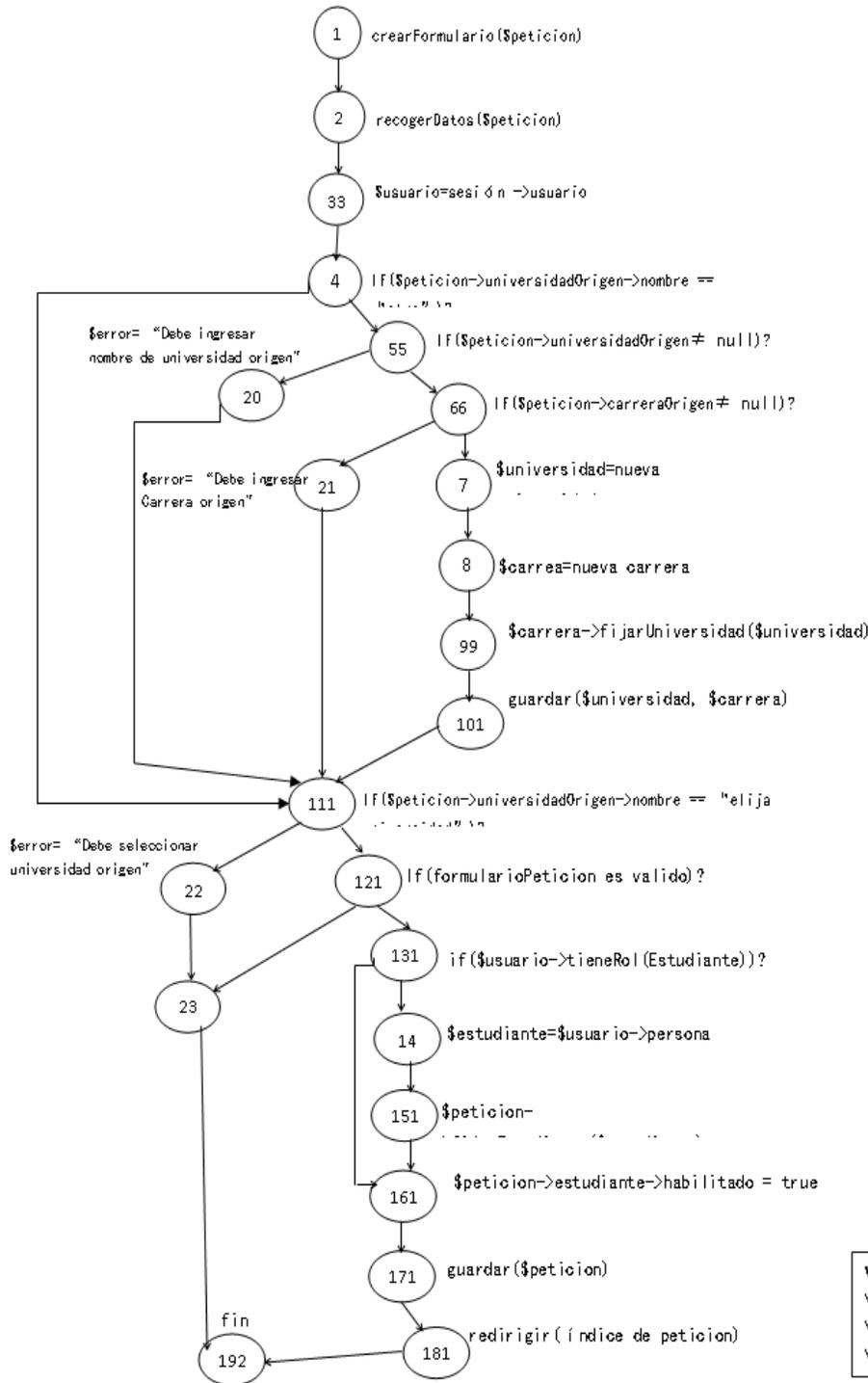
Pasaporte	9. p= alfanumérico 10. 10 caracteres ≤ p ≥ 5 caracteres 11. p= vacío y ti="cedula"	12. p< 5 caracteres 13. p= vacío y ti= "pasaporte"
Apellido	14. a= alfanumérico	15. a= vacío
Nombre	16. n= alfanumérico	17. n= vacío
Telefono	18. t= numérico 19. t= vacío	
Email	20. e= alfanumérico formato email	21. e= vacío 22. e= alfanumérico diferente de formato email
Universidad Origen	23. uo= alfanumérico 24. uo= "otra"	25. uo= "elija universidad"
Nombre universidad origen	26. nuo=alfanumérico	27. nuo= vacio
Carrera Origen	28. co= alfanumérico 29. co= "otra"	30. co= "elija carrera"
Nombre carrera origen	31. nco=alfanumérico	32. co= vacio
Carrera Destino	33. cd= alfanumérica	

EJECUCIÓN DE PRUEBAS

	Clase de equivalencia	Tipo de identificación	Cedula	Pasaporte	Apellido	Nombre	Telefono	Email	Universidad Origen	Nombre Universidad Origen	Carrera Origen	Nombre Carrera Origen	Carrera Destino	Resultado
1	1,3,4,14,16,18,20,23,26,28,31,33	Cedula	1103780928	-	Parra	Julio	565845	bm@gmail.com	Unl	Unl	Electrónica	Electrónica	Sistemas	Petición guardada
2	1,3,4,14,16,19,20,24,26,29,31,33	Cedula	1103780928	-	Parra	Julio	-	bm@gmail.com	Otra	Uda	Otra	Derecho	Sistemas	Petición guardada
3	1,8,11,14,16,19,20,24,27,30,32,33	Cedula	-	-	Parra	Julio	-	bm@gmail.com	Otra	-	Elija carrera	-	Sistemas	Debe ingresar número de cédula. Debe. Ingresar nombre Universidad Origen.
4	2,5,9,10,14,16,19,20,24,26,29,32,33	Pasaporte	-	874567	Buri	María	-	bm@gmail.com	Otra	Uda	Otra	-	Sistemas	Debe ingresar la carrera origen.
5	1,3,4,14,16,19,20,25,27,32,33	Cedula	1103505507	-	Pérez	Paul	-	bpaz@gmail.com	Elija universidad	-	-	-	Sistemas	Debe seleccionar universidad origen.

6	2,5,9,10,14, 16,19,21,24,2 7, 29,32,33	Pasaporte	-	5789039	Vera	Juan	-	-	Otra	-	Otra	-	Sistema s	Debe ingresar nombre de universidad origen Debe ingresar nombre de carrera origen.
---	--	-----------	---	---------	------	------	---	---	------	---	------	---	--------------	---

PRUEBAS DE CAJA BLANCA CREAR PETICIÓN



$V(G) = A - N + 2$
 $V(G) = 28 - 23 + 2 = 7$
 $V(G) = P + 1$
 $V(G) = 6 + 1 = 7$

EJECUCIÓN DE PRUEBAS DE CAJA BLANCA GUARDAR NUEVO PETICIÓN

No. Caso	Camino	ENTRADA												Resultado
		Tipo de identificación	Cedula	Pasaporte	Apellido	Nombre	Telefono	Email	Universidad Origen	Nombre Universidad Origen	Carrera Origen	Nombre Carrera Origen	Carrera Destino	
1	1-2-3-4-11-12-13-14-15-16-17-18-19	Cedula	1103780928	-	Pérez	Paul	565845	bm@gmail.com	Unl	Unl	Electrónica	Electrónica	Sistemas	Petición guardada
2	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19	Cedula	1103735237	-	Pérez	Paul	-	bpaz@gmail.com	Otra	Uda	Otra	Derecho	Sistemas	Petición guardada
3	1-2-3-4-5-6-7-8-9-10-11-12-13-16-17-18-19	Cedula	1103505507	-	Solano	Víctor	-	soVi@g.com	Otra	Ute	Otra	Informática	Sistemas	Petición guardada
4	1-2-3-4-5-20-11-12-23-19	Pasaporte	-	345678	Ruiz	Olga	-	rbe@g.com	Otra	-	Elija Universidad	-	Sistemas	Debe ingresar nombre de universidad origen

5	1-2-3-4-5-6-21-11-12-23-19	Pasaporte	-	5789039	Ruiz	Olga	-	rbe@g.com	Otra	Ute	Otra	-	Sistemas	Debe ingresar nombre de carrera origen
6	1-2-3-4-11-22-23-19	Pasaporte	-	587698	Ruiz	Olga	-	rbe@g.com	Elija Universidad	-	Elija carrera	-	Sistemas.	Debe seleccionar universidad origen
7	1-2-3-4-11-12-23-19	Pasaporte	-	579681	Ruiz	Olga	-	-	Elija Universidad	-	-	-	Sistemas	Debe ingresar email.