



Universidad
Nacional
de Loja

Universidad Nacional de Loja

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

Maestría en Ingeniería en Software

**Desarrollo de un Sistema de Gestión de Aprendizaje como
herramienta que permita organizar, centralizar contenidos
académicos y facilitar la exposición de recursos formativos**

Trabajo de Titulación previo a la
obtención del título de Magíster en
Ingeniería en Software

AUTOR:

Edison Enrique Ramírez Coronel

DIRECTOR:

Ing. Edwin René Guamán Quinche, Mg Sc.

Loja - Ecuador
2023

Certificación

Loja, 24 de abril de 2023

Ing. René Guamán Quinche, Mg.Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo proceso de la elaboración del Trabajo de Titulación denominado: **Desarrollo de un Sistema de Gestión de Aprendizaje como herramienta que permita organizar, centralizar contenidos académicos y facilitar la exposición de recursos formativos**, previo a la obtención del título de **Magíster en Ingeniería en Software**, de autoría del estudiante **Edison Enrique Ramirez Coronel**, con cédula de identidad Nro. **1103796973**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja para el efecto, autorizo la presentación para la respectiva sustentación y defensa.

Ing. René Guamán Quinche, Mg.Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

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

Firma:

Cédula de Identidad: 1103796973

Fecha: 03/05/2023

Correo electrónico: edison.ramirez@unl.edu.ec

Teléfono: +593 994244041

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

Yo, **Edison Enrique Ramirez Coronel**, declaro ser autor del Trabajo de Titulación denominado: **Desarrollo de un Sistema de Gestión de Aprendizaje como herramienta que permita organizar, centralizar contenidos académicos y facilitar la exposición de recursos formativos** como requisito para optar el título de **Magíster en Ingeniería en Software**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

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

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

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

Firma:

Autor: Edison Enrique Ramírez Coronel

Cédula de identidad: 1103796973

Dirección: Antonia Salinas S2-106 y Diego de Salazar esquina.

Correo electrónico: edison.ramirez@unl.edu.ec

Teléfono: +593 994244041

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. René Guamán Quinche Mg. Sc.

Dedicatoria

Primeramente, dedico este trabajo a Dios, quien me ha dado su gracia y sabiduría para trabajar diariamente en pos de mis metas.

Quiero dedicar el presente Trabajo de Titulación a mis hijos y mi esposa que son parte fundamental en mi vida. Gracias por su apoyo incondicional, por su paciencia y por motivarme siempre a seguir adelante en este camino.

A mis padres, quienes han sido mi mayor inspiración y motivación, gracias por creer en mí y por darme las herramientas necesarias para alcanzar mis metas. A Gustavo y Beatriz, que me han convencido que todos podemos alcanzar logros con un simple cambio de mentalidad.

A mis profesores y compañeros de universidad, gracias por compartir conmigo sus conocimientos y por ser una fuente de inspiración y motivación en mi formación académica.

Finalmente, quiero dedicar este Trabajo de Titulación a todas aquellas personas que, como yo, luchan cada día por alcanzar sus metas y sueños. Espero que mi ejemplo les inspire a perseverar y a nunca rendirse, y que les demuestre que, con esfuerzo y dedicación, todo es posible.

Con gratitud

Edison Enrique Ramírez Coronel

Agradecimiento

Quiero dirigirme a la Universidad Nacional de Loja para expresar mi más sincero agradecimiento por la oportunidad que me brindaron de llevar a cabo mi Trabajo de Titulación en esta prestigiosa institución.

Durante el proceso académico, tuve la oportunidad de aprender de excelentes profesionales en la Maestría en Ingeniería de Software, quienes me brindaron su experiencia y conocimientos para ayudarme a lograr los objetivos de mi Trabajo de Titulación.

También quisiera agradecer a mi director de Trabajo de Titulación, quien me brindó su guía y apoyo durante todo el proceso de investigación y escritura, y me motivó a dar lo mejor de mí en todo momento.

No puedo dejar de mencionar al personal administrativo, quienes siempre estuvieron dispuestos a ayudarme y responder a mis preguntas y solicitudes.

Este Trabajo de Titulación ha sido una experiencia muy enriquecedora en mi formación profesional y personal, y estoy agradecido por haber tenido la oportunidad de llevarlo a cabo en esta institución de renombre.

Nuevamente, muchas gracias por todo lo que hicieron por mí y por mi Trabajo de Titulación.

Edison Enrique Ramírez Coronel

Índice de contenidos

Portada	i
Certificación	ii
Autoría	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de contenidos	vii
Índice de tablas	x
Índice de figuras	xi
Índice de anexos	xii
1. Título	1
2. Resumen	2
3. Introducción	4
4. Marco teórico	6
4.1 Evolución del elearning	6
4.2 Características de los entornos virtuales de enseñanza-aprendizaje	6
4.3 Teorías del aprendizaje	7
4.3.1 Conductismo	8
4.3.2 Constructivismo	8
4.3.3 Cognoscitivismo	8
4.3.4 Aprendizaje social	8
4.3.5 Socioconstructivismo	9
4.4 Diseño Instruccional	9
4.4.1 El principio de demostración	10
4.4.2 El principio de aplicación	11
4.4.3 El principio de integración	11
4.4.4 Gestión de la información organizacional del curso a través de un LMS	11
4.5 Tecnología educativa:	12
4.5.1 Programación, repetición y práctica	12
4.5.2 Entrenamiento basado en computadora (CBT) con multimedia	12
4.5.3 Entrenamiento basado en Internet (IBT)	12
4.5.4 E-learning	13
4.5.5 M-learning	13
4.5.6 Mix-Learning	13
4.5.7 Software Social	13
4.6 Objetos de Aprendizaje	13

4.6.1	¿Por qué utilizar objetos de aprendizaje?	14
4.7	Arquitectura de Ambientes Virtuales de Aprendizaje	14
5.	Metodología	16
5.1	Área de estudio	16
5.2	Procedimiento	17
5.2.1	Método Aplicado.	17
5.2.2	Método de Investigación.....	17
5.2.3	Método de desarrollo en cascada para la construcción del Sistema de Gestión de Aprendizaje.....	17
5.2.4	Integración de ICONIX dentro del proceso de desarrollo del SGA	18
5.2.5	Uso del Patrón Modelo Vista Controlador (MVC).....	22
6.	Resultados.....	24
6.1	Objetivo 1: Identificar los requerimientos y módulos que contempla el sistema de gestión de aprendizaje a través de la revisión de literatura y de soluciones establecidas en el mercado.	24
6.1.1	Lista de requerimientos funcionales principales.....	24
6.1.2	Lista de requerimientos no funcionales	26
6.1.3	Modelo del dominio.....	27
6.1.4	Diccionario de datos	29
6.1.5	Modelo del caso de uso.....	30
6.1.6	Diagramas de caso de uso.....	33
6.1.7	Descripción de Casos de Uso	34
6.1.8	Administrar un Curso	37
6.1.9	Modelado de la interacción.....	41
6.2	Objetivo 2: Aplicar el patrón de diseño de software Modelo Vista Controlador, haciendo uso del Framework Laravel 9 para crear una solución robusta y eficiente que permita a futuro lograr mantenibilidad en la solución.	48
6.2.1	El Modelo	48
6.2.2	La Vista.....	49
6.2.3	El Controlador	49
6.2.4	Responsabilidades.....	49
6.2.5	Arquitectura de una aplicación en Laravel	52
6.2.6	Detalle de resultados.....	54
6.3	Objetivo 3: Validar y verificar el funcionamiento de la solución de software mediante pruebas de software en un ambiente Cloud.	56
6.3.1	Contexto de las pruebas a ejecutar.....	56
6.3.2	Comunicación de las Pruebas	58
6.3.3	Registro de Riesgos	58

6.3.4	Estrategia de pruebas	60
6.3.5	Actividades y Estimados de Prueba.....	65
6.3.6	Personal.....	65
6.3.7	Necesidades de Entrenamiento	66
6.3.8	Cronograma	66
6.3.9	Informe de pruebas de usabilidad	67
7.	Discusión.....	68
7.1	Impacto de la educación virtual	68
7.2	Implementación de un Sistema de Gestión de Aprendizaje como herramienta que permita organizar, centralizar contenidos académicos y facilitar la exposición de recursos formativos	69
7.3	Uso de la metodología de desarrollo ágil ICONIX	70
7.4	Uso del diseño arquitectónico Modelo-Vista-Controlador.....	71
8.	Conclusiones.....	73
9.	Recomendaciones.....	75
10.	Bibliografía	77
11.	Anexos.....	79

Índice de tablas:

TABLA I. REQUERIMIENTOS FUNCIONALES ESTUDIANTE	24
TABLA II. REQUERIMIENTOS FUNCIONALES INSTRUCTOR	25
TABLA III. REQUERIMIENTOS FUNCIONALES ADMINISTRADOR	26
TABLA IV. REQUERIMIENTOS NO FUNCIONALES	26
TABLA V. DICCIONARIO DE DATOS	29
TABLA VI. CASOS DE USO DE TODOS LOS USUARIOS	31
TABLA VII. CASOS DE USO DEL ESTUDIANTE	31
TABLA VIII. CASOS DE USO DEL INSTRUCTOR	32
TABLA IX. CASOS DE USO DEL ADMINISTRADOR DEL SISTEMA	32
TABLA X. DESCRIPCIÓN DEL CASO DE USO VISUALIZAR CURSOS	35
TABLA XI. DESCRIPCIÓN DEL CASO DE USO ADMINISTRAR UN CURSO	38
TABLA XII. DESCRIPCIÓN UC REALIZAR ENROLAMIENTO EN CURSO	40
TABLA XIII. COMUNICACIÓN DE LAS PRUEBAS	58
TABLA XIV. REGISTRO DE RIESGOS	59
TABLA XV. DETALLE DE MÉTRICAS INTERNAS	61
TABLA XVI. DETALLE DE MÉTRICAS EXTERNAS	62
TABLA XVII. CARACTERÍSTICAS DE CONFIABILIDAD	63
TABLA XVIII. CARACTERÍSTICAS DE USABILIDAD	63
TABLA XIX. DETALLE DE ROLES, ACTIVIDADES Y RESPONSABILIDADES ..	66
TABLA XX. CRONOGRAMA GENERAL DE PRUEBAS	67

Índice de figuras:

Fig. 1. Los principios del TBI	10
Fig. 2. Aspectos, Arquitectura de ambientes Virtuales de Aprendizaje	15
Fig. 3. Las 5 etapas principales del desarrollo en cascada	18
Fig. 4. Descripción del proceso ICONIX	23
Fig. 5. Esquema de capas del patrón modelo vista controlador	19
Fig. 6. Descripción del MVC según Joseph Spinelli	23
Fig. 7. Diagrama que muestra el modelo de dominio inicial	23
Fig. 8. Diagrama que muestra el modelo de dominio en la seguridad del sistema	28
Fig. 9. Diagrama que detalla los principales paquetes que compone el SGA	29
Fig. 10. Diagrama que detalla la administración de cursos por rol del sistema	33
Fig. 11. Prototipo que muestra la lista de cursos en el SGA	34
Fig. 12. Diagrama de robustes, caso de uso: Visualizar curso	35
Fig. 13. Prototipo que muestra el caso de uso: Administrar curso	37
Fig. 14. Prototipo que muestra el caso de uso: Administrar secciones	37
Fig. 15. Diagrama de robustes, caso de uso: Administrar curso	39
Fig. 16. Prototipo que muestra la información del curso y enrolamiento	40
Fig. 17. Diagrama de secuencia sobre el caso de uso: Buscar curso	42
Fig. 18. Diagrama de secuencia sobre el caso de uso: Enrolar curso	43
Fig. 19. Diagrama de secuencia, caso de uso: Administrar curso	44
Fig. 20. Diagrama de paquetes del módulo Learning	45
Fig. 21. Diagrama de clases del módulo de Learning	46
Fig. 22. Diagrama de componentes del módulo de Learning	47
Fig. 23. Diagrama de despliegue del sistema	47
Fig. 24. Diseño arquitectónico: Modelo-Vista-Controlador	48
Fig. 25. Estructura de funcionamiento del Modelo-Vista-Controlador	49
Fig. 26. Diagrama de secuencia del diseño arquitectónico: MVC	51
Fig. 27. Elementos básicos que compone un proyecto en Laravel	53
Fig. 28. Iteración de los elementos de un proyecto en Laravel	53
Fig. 29. Diagrama que detallan todos los modelos creados en el proyecto	54
Fig. 30. Diagrama que detallan todas las vistas creadas en el proyecto	55
Fig. 31. Diagrama que detallan todos los controladores creados en el proyecto	55
Fig. 33. Diagrama de pruebas	55

Índice de anexos:

Anexo 1. GitHub Framework Laravel	79
Anexo 2. Sitio web con la documentación oficial del Framework Laravel	79
Anexo 3. Diagrama físico de base de datos	79
Anexo 4. Informe de pruebas de usabilidad	83
Anexo 5. Certificación de traducción del resumen.....	95

1 Título

Desarrollo de un Sistema de Gestión de Aprendizaje como herramienta que permita organizar, centralizar contenidos académicos y facilitar la exposición de recursos formativos.

2 Resumen

En este mundo cambiante cada vez existen más estudiantes que no pueden asistir a clases por cuestiones de trabajo, responsabilidades familiares, la distancia o los gastos que conlleva la educación tradicional, frente a esto los sistemas de gestión del aprendizaje son cada vez más populares en el ámbito educativo debido a su amplia gama de funcionalidades y ventajas. Estos sistemas combinan elementos pedagógicos y modelos de enseñanza para proporcionar a los estudiantes una herramienta de aprendizaje moderna y efectiva.

Frente a esto, el presente trabajo de titulación propone el desarrollo de un Sistema de Gestión de Aprendizaje como herramienta que permitió organizar, centralizar contenidos académicos para facilitar la exposición de recursos formativos.

En este proceso de desarrollo de software se utilizó la metodología ICONIX por ser iterativo e incremental enfocándose en la construcción de prototipos funcionales y la validación temprana de requisitos, dividiendo el proceso en cuatro fases: Requisitos, Diseño, Implementación y Pruebas.

Como resultado final se elaboró un sistema que permitió realizar la centralización y la gestión del proceso de enseñanza-aprendizaje permitiendo administrar todos los recursos educativos en un solo lugar, facilitando su gestión y organización.

Esta herramienta permitió realizar un seguimiento más preciso y continuo del progreso de los estudiantes, lo que facilita la identificación temprana de necesidades y la implementación de estrategias de mejora.

La validación del avance en la capacitación de recursos formativos, así como la evaluación de los diferentes componentes fue importante al momento de reconocer las fortalezas y debilidades de los participantes, esto se logró gracias al monitoreo y estudio de métricas relacionados con el progreso de cada participante.

Palabras clave: *gestión de conocimiento, plataforma educativa, plataforma en línea, LMS, SGA, aprendizaje, aula virtual.*

2.1. Abstract

In this changing world, there are more and more students who cannot attend classes due to work, family responsibilities, distance or the expenses that traditional education entails. Against this, learning management systems are increasingly popular in the educational field due to its wide range of functionalities and advantages. These systems combine pedagogical elements and teaching models to provide students with a modern and effective learning tool.

Faced with this, the current degree work proposes the development of a Learning Management System as a tool that allowed organizing and centralizing academic content to facilitate the exposure of training resources.

In this software development process, the ICONIX methodology was used because it is iterative and incremental, focusing on the construction of functional prototypes and the early validation of requirements, dividing the process into four phases: Requirements, Design, Implementation and Testing.

As a final result, a system was developed that allowed the centralization and management of the teaching-learning process, allowing the administration of all educational resources in one place, facilitating their management and organization.

This tool allowed more precise and continuous monitoring of student progress, which facilitates the early identification of needs and the implementation of improvement strategies.

The validation of the progress in the training of teaching resources, as well as the evaluation of the different components was important when recognizing the strengths and weaknesses of the participants, this was achieved thanks to the monitoring and study of metrics related to the progress of each participant.

Keywords: *knowledge management, educational platform, online platform, LMS, learning, virtual classroom.*

3 Introducción

La construcción de un Sistema de Gestión de Aprendizaje (SGA) es un proceso clave en el ámbito educativo actual. Los SGA son plataformas en línea diseñadas para mejorar el proceso de enseñanza y aprendizaje mediante el uso de tecnologías de la información y la comunicación.

Según los estudios realizados por Pérez, Fernández: Los entornos virtuales de aprendizaje son una herramienta eficaz para potenciar los cambios necesarios y apoyar el desarrollo de nuevos modelos como el presencial o el modelo semipresencial con un amplio soporte de las tecnologías [1].

Varios autores han estudiado las ventajas relacionadas con esta metodología, destacando la gran rentabilidad –consecuencia del ahorro de gastos de desplazamiento y de espacio, ya que ofrece oportunidades de aprendizaje para un gran número de estudiantes sin necesidad de muchos espacios [2]. Además, con esta metodología, el estudiante consigue un aumento de la satisfacción personal y una disminución del estrés, ya que puede elegir su propio ritmo de aprendizaje y, de esta manera, incrementar su autonomía [3].

Este tipo de educación ofrece a los estudiantes un recurso fácilmente accesible desde cualquier lugar o ubicación [4] y brinda oportunidades a personas con discapacidad [5]. Sin embargo, existen estudios que identifican algunos inconvenientes relacionados con la educación online, como la dificultad de comprensión de la información más técnica. Por ello, establecen que esta modalidad de enseñanza no puede utilizarse de manera efectiva en áreas científicas que requieren experiencias prácticas [6].

Un SGA permite a los estudiantes y profesores administrar y acceder a los recursos educativos de manera centralizada, realizar un seguimiento del progreso académico, colaborar en línea y comunicarse de manera efectiva. El objetivo principal de este trabajo de titulación es diseñar y construir un SGA que sea fácil de usar y esté diseñado para adaptarse a las necesidades específicas de un entorno educativo determinado. Además, se busca garantizar que el SGA sea escalable y capaz de soportar un número creciente de usuarios y recursos educativos a medida que se amplía el alcance del proyecto. En este informe se detalló el proceso de diseño, desarrollo e implementación del SGA, así como los desafíos y soluciones encontrados durante el proceso. Además, se presentarán los resultados obtenidos y la eficacia del SGA en la mejora del rendimiento académico y la satisfacción de los usuarios.

Para lograr los objetivos del trabajo de titulación, se llevará a cabo un proceso de diseño cuidadoso que tenga en cuenta las necesidades específicas de los usuarios y las características del entorno educativo en el que se implementará el SGA. Se identificarán los requisitos funcionales y no funcionales del sistema y se desarrollará un plan de desarrollo detallado que tenga en cuenta la arquitectura del sistema, la selección de tecnologías y herramientas, y el ciclo de vida del software.

Durante la implementación del SGA, se pondrá especial atención en la seguridad y la privacidad de los datos, ya que se manejarán información personal y académica confidencial. Se establecerán medidas de seguridad adecuadas para garantizar que la información sea protegida y se realizarán pruebas de seguridad regulares para garantizar la integridad del sistema.

La evolución del e-learning está sujeta a las transformaciones tecnológicas y a los cambios relacionados con los espacios y tiempos para la formación y el aprendizaje. Inicialmente, el e-learning estaba vinculado al uso de los ordenadores personales y actualmente está sostenido por los sistemas distribuidos que han favorecido las redes de aprendizaje y el acceso a las plataformas desde cualquier dispositivo. Así, el uso de dispositivos móviles se ha integrado en los diseños educativos y las actividades de aprendizaje en línea [7].

Una vez implementado, el SGA será sometido a pruebas rigurosas y se llevarán a cabo ajustes y mejoras para garantizar que funcione de manera óptima y cumpla con las expectativas de los usuarios. Se establecerán procedimientos para la capacitación de los usuarios y se proporcionará soporte técnico continuo para garantizar la máxima satisfacción de los usuarios.

4 Marco teórico

Un sistema de gestión de aprendizaje (LMS) es una plataforma tecnológica que permite la administración y el seguimiento de la enseñanza y el aprendizaje en un entorno virtual. Un LMS brinda a los estudiantes acceso a recursos de aprendizaje, tareas y evaluaciones, y a los instructores les proporciona herramientas para crear y administrar contenido educativo permitiendo realizar un seguimiento analítico a los programas de capacitación.

El Sistema de Gestión de Aprendizaje se ha convertido en una herramienta increíblemente poderosa para las organizaciones que buscan mejorar el rendimiento y la retención de su fuerza de trabajo. La mayoría de los sistemas de gestión de aprendizaje son soluciones de software basados en la nube que las empresas usan como su herramienta fundamental para la gestión de sus programas de capacitación. Así como los equipos de ventas usan y dependen de softwares CRM, o los equipos de recursos humanos dependen de software HRIS, el software LMS es usualmente la tecnología base usada por los departamentos de capacitación y desarrollo de las empresas [8].

4.1 Evolución del elearning

La evolución del e-learning está sujeta a las transformaciones tecnológicas y a los cambios relacionados con los espacios y tiempos para la formación y el aprendizaje. Inicialmente, el e-learning estaba vinculado al uso de los ordenadores personales y actualmente está sostenido por los sistemas distribuidos que han favorecido las redes de aprendizaje y el acceso a las plataformas desde cualquier dispositivo. Así, el uso de dispositivos móviles se ha integrado en los diseños educativos y las actividades de aprendizaje en línea [7].

4.2 Características de los entornos virtuales de enseñanza-aprendizaje

El avance acelerado de la Tecnología de la Información y Comunicación (TIC) han generado múltiples beneficios a la colectividad, en especial en el ámbito educativo vinculado con los intereses y necesidades del educando a la vez dejando atrás el viejo modelo educativo donde el docente era un mero transmisor y expositor de información.

Por eso, los ambientes de aprendizaje no se circunscriben al espacio escolar más bien se trata de aquellos espacios en donde se crean las condiciones para que el participante se apropie de nuevos conocimientos, experiencias y elementos que le generen procesos de análisis, reflexión y apropiación. Cabe puntualizar que, una de las bondades de las TIC está relacionada con la

conectividad que facilita acceder a la información de forma rápida y hacer uso de cualquier recurso. Por lo demás, economiza esfuerzo y tiempo a través del uso de internet [9].

Desde el surgimiento de los entornos virtuales de enseñanza se ha buscado la manera de vincular en el mismo todos los aspectos informáticos incorporando la multimedia como imagen, sonido, videos, entre otros. De la misma manera, tomar en consideración los cinco componentes principales que le conforman:

- El espacio.
- Los estudiantes.
- Tutores.
- Los contenidos educativos.
- Los medios.

Por ello, la planeación de la enseñanza en línea presenta ciertas falencias por parte del profesorado puesto que algunos continúan con la obsesión por la transmisión de contenidos, tendencia al uso de metodologías de naturaleza conductista, preferencia a la evaluación sumativa olvidándose en muchos casos del análisis de los procesos de construcción del conocimiento; excesivo descuido en el diseño de estrategias instructivas [9].

4.3 Teorías del aprendizaje

Dentro del diseño y desarrollo de un LMS, es fundamental considerar las teorías del aprendizaje para crear un entorno educativo efectivo.

Las teorías del aprendizaje son una construcción que explica y profetiza el cómo aprende el ser humano basándose en la concepción de diversos teóricos. Así de una manera general las teorías contribuyen al conocimiento y desde diferentes enfoques explican el cómo se da el proceso de aprendizaje en los seres humanos. Son aquellas que realizan la representación de un proceso que permitirá a una persona aprender algo.

Estas ayudarán a entender, anticipar y regular la conducta a través del diseño de maniobras que facilitarán el acceso al conocimiento. De este modo interpretará los acontecimientos de aprendizaje y sugerirá soluciones a inconvenientes que pueden surgir en el proceso de enseñanza- aprendizaje.

4.3.1 Conductismo

Es una corriente de la psicología que se centra en el comportamiento humano y animal, deja de lado todo lo intrínseco para concentrarse en las conductas observables y medibles. Esta conducta está determinada por refuerzos y castigos.

Se considera como fundador de esta teoría a **J.B Watson**, el nombró a la conducta su objeto de estudio, en ese momento la conducta que se estudiaba debía de ser cien por ciento observable y operacionalizable. Los fines que tenía el conductismo era la investigación y el análisis de las relaciones que rigen los sucesos ambientales y las conductas de los organismos en base a estímulo-respuesta.

4.3.2 Constructivismo

Se puede entender como aquel que permite que el alumno construya su propio aprendizaje, las ideas principales de esta teoría son: el alumno es responsable de su propio conocimiento, construye su conocimiento por sí mismo, relaciona la información nueva con conocimientos previos, establece relaciones entre elementos, da significado a la información que recibe, necesita un apoyo pudiendo ser el profesor, pares o padres y el profesor se convierte en el orientador. Los principales estudiosos del constructivismo son: Piaget. y Ausubel.

4.3.3 Cognoscitivismo

Los aportes de esta teoría se enmarcan en considerar al ser humano como ente pensante quien transforma el pensamiento como resultado de su ambiente interno y externo.

De acuerdo con esta teoría la concepción de la enseñanza puede reducirse en los siguientes puntos: aprender y solucionar problemas, aprendizajes significativos con sentido, desarrollar habilidades intelectuales y estratégicas, proceso sociocultural, conocimiento previo y metas de aprendizaje.

4.3.4 Aprendizaje social

Como principio de esta teoría se tiene que el aprendizaje directo no es el principal mecanismo de enseñanza, sino que el elemento social de la base de un aprendizaje nuevo en los individuos, esta teoría resulta útil para explicar cómo las personas aprenden y desarrollan nuevas formas de conducirse mediante la observación a otros individuos.

4.3.5 Socioconstructivismo

Este paradigma fue desarrollado por L.S Vigotsky en la década de 1920, se basa en el alumno como ser social, donde la cultura modifica o crea todo su aprendizaje, si el conocimiento de una persona se construye socialmente de acuerdo a esta teoría será entonces necesario que la planeación se diseñe con actividades que permitan precisamente esa interacción social y que la relación no se centre en alumno-maestro, sino que se lleve a éste a su comunidad y se promueva la participación activa con sus pares [10].

4.4 Diseño Instruccional

El diseño instruccional (DI) se considera parte central de todo proceso de aprendizaje porque representa la planificación detallada de las actividades educativas sin importar la modalidad de enseñanza. El DI es un proceso sistemático, planificado y estructurado, que se apoya en una orientación psicopedagógica del aprendizaje para producir con calidad y pertinencia, una amplia variedad de materiales educativos (unidades didácticas, actividades de aprendizaje, autoevaluaciones, etc.) adecuados a las necesidades de aprendizaje de los estudiantes, y a las modalidades educativas alternas, mediadas por las Tecnologías de la Información y Comunicación (TIC). El DI que se apoya en las TIC, deja de ser lineal porque ofrece múltiples perspectivas de creación y de uso, es dialéctico y holístico, lo que desemboca en una diversidad de interacciones, que deben ser integradas. Así, el aprendizaje es el resultado de los esfuerzos mancomunados de grupos de personas que procuran resolver un problema. Por lo tanto, es necesario formular diseños instruccionales que permitan el acceso a la información de manera compartida, a través de la facilitación de debates generadores de conocimientos, dentro de grupos de discusión.

El DI guiado por el aprendizaje basado en proyectos o tareas (TBL) es un enfoque dinámico para llevar a cabo procesos de enseñanza-aprendizaje, en los cuales, los estudiantes exploran los problemas del mundo real y sus desafíos. El TBL es un tipo de aprendizaje activo y colaborativo que inspira, motiva y compromete a los estudiantes para obtener un conocimiento más profundo de los temas que se están estudiando. Hace hincapié en la comunicación, alentando las habilidades de pensamiento de orden superior (análisis, síntesis, evaluación y creación) y le da valor a la investigación y a la resolución de problemas, apoyándose en la tecnología para complementar el trabajo realizado en el salón de clases. Entre los objetivos del TBL se encuentran la motivación para el aprendizaje autodirigido, el desarrollo de estructuras

de conocimiento flexibles y de la experiencia de adaptación y la enseñanza de la resolución de problemas de manera efectiva.

Un proyecto se define como un trabajo educativo más o menos prolongado con fuerte participación de los estudiantes en su planteamiento, su diseño y su seguimiento que propicia la indagación en una labor autopropulsada conducente a resultados propios.

Las ideas para los proyectos no deben surgir de una imposición, son los estudiantes los que deben encontrar el interés y, a partir de ahí, trabajar la temática. Los posibles proyectos pueden ser científicos, tecnológicos o de investigación ciudadana o proyectos ciudadanos.

De una revisión crítica y sistemática sobre modelos e investigación en DI, David Merrill (2009) identificó y resumió un conjunto de cinco principios que apoyan el TBL y que se encuentran interrelacionados (ver Fig. 1): Tareas, Activación, Demostración, Aplicación e Integración [11].



Fig. 1. Describe los principios del TBL

4.4.1 El principio de demostración

El aprendizaje se promueve cuando los estudiantes observan una demostración de las habilidades que serán aprendidas y que son consistentes con el tipo de contenido que se está enseñando y no solo con la transmisión de información. El aprendizaje por demostración es eficaz cuando se guía a los estudiantes para que relacionen la información o la estructura de organización de la información en casos concretos. Este aprendizaje se ve reforzado cuando

los estudiantes observan los medios que son relevantes para el contenido y discuten y colaboran entre sí.

4.4.2 El principio de aplicación

Este principio se observa cuando los estudiantes aplican conocimientos o habilidades recién adquiridas en las tareas que son consistentes con el tipo de contenido que se enseña y es más efectivo cuando los estudiantes reciben retroalimentación intrínseca o correctiva. También se enriquece cuando los estudiantes son tutorados y cuando este coaching se retira gradualmente conforme se avanza en tareas subsecuentes. El principio de aplicación se ve reforzado cuando los estudiantes trabajan en equipos.

4.4.3 El principio de integración

El aprendizaje se promueve cuando los estudiantes integran sus nuevos conocimientos y destrezas en su mundo, en su quehacer cotidiano.

El aprendizaje se facilita cuando el estudiante puede demostrar, discutir su nuevo conocimiento o habilidad y cuando puede crear, inventar o explorar nuevas formas de utilización.

Dentro del ciclo completo de instrucción se puede identificar otro ciclo más sutil con hábitos que se interrelaciona con los principios básicos y que son: estructura-guía-coaching-reflexión.

En la fase de activación, la instrucción debe proporcionar a los alumnos una estructura de organización sobre lo que ya saben. Esta estructura facilita la adquisición de los nuevos conocimientos durante las fases restantes del ciclo de instrucción. En la fase de demostración no sólo se orienta a relacionar la información anterior, sino también se relaciona el nuevo material con la estructura desarrollada en la fase de activación. En la fase de aplicación, el coaching ayuda a los estudiantes a utilizar esta estructura para facilitar el uso de las habilidades recién adquiridas en la realización de las nuevas tareas.

En la fase de integración, se debe promover la reflexión para que los alumnos resuman lo que han aprendido y vuelvan a examinar cómo el conocimiento nuevo se relaciona con lo que previamente conocían, utilizando las estructuras cognoscitivas desarrolladas [11].

4.4.4 Gestión de la información organizacional del curso a través de un LMS

La gestión del curso que se lleva a cabo por medio de un LMS se puede considerar como un entorno de enseñanza organizacional (OTE). Cada uno de los temas del curso pretende lograr

un avance gradual en el objetivo de construir un sistema de información para cada proyecto social escogido de manera personal por cada alumno, cuya finalidad es la mejora de una problemática de la comunidad donde se vive. De esta manera, se busca que el alumno esté interesado y se mantenga motivado para lograr este objetivo durante todo el curso y se mantenga la creencia de que pueda continuarlo, aun después de haber terminado el curso.

El LMS apoya la práctica docente y el aprendizaje. Consisten de un conjunto de herramientas para subir contenidos, comunicarse, colaborar y obtener retroalimentación.

Incorpora también instrumentos analíticos del aprendizaje como el procesamiento de datos, el análisis de estadísticas, la generación de informes e información sobre las interacciones del alumno y de su progreso en el entorno [11].

4.5 Tecnología educativa:

La Tecnología Educativa se refiere al uso de herramientas y recursos tecnológicos para mejorar el proceso de enseñanza y aprendizaje en el ámbito educativo.

La aplicación de las tecnologías en la vida diaria ha tenido una constante evolución, puesto que el surgimiento de estas ha sido en respuesta a las necesidades del hombre, las cuales han ido cambiando a través del tiempo.

4.5.1 Programación, repetición y práctica

Esta primera etapa se sitúa a finales de los años 70 y principios de los 80. Aquí el uso de las tecnologías se veía marcado por la utilización de programas sencillos destinados a desarrollar determinadas habilidades en los estudiantes, mediante la técnica de repetición y práctica.

4.5.2 Entrenamiento basado en computadora (CBT) con multimedia

Con la llegada de los equipos multimedia y las unidades de CD-ROM a finales de los 80 principios de los 90, se vio la posibilidad de conglomerar varios medios como fueron las imágenes y el sonido en uno solo. Estas características fueron añadidas al material didáctico enriqueciendo la utilización de la computadora en el aula.

4.5.3 Entrenamiento basado en Internet (IBT)

Ante la necesidad de tener la información actualizada en todo momento y gracias a las posibilidades comunicativas que brinda el Internet; a principios de los 90 se llevó el ámbito

educativo a esta plataforma. En un principio las limitantes del medio sólo permitían el manejo de texto y algunas imágenes, pero con el desarrollo de la world wide web, otra vez se hizo posible la presencia de la multimedia.

4.5.4 E-learning

A finales de los 90 principios del 2000 con el desarrollo de IBT surge el e-learning, el cual de acuerdo con Morten Fíate Paulsen (2003) se enfoca más en el contenido didáctico en sí y está orientado a una interactividad entre los estudiantes con los contenidos, dotándolos de retroalimentación inmediata y un sinfín de recursos multimediales e hipertextuales.

4.5.5 M-learning

Como una nueva etapa del e-learning se encuentran diversas variantes que se apoyan en la tecnología, un ejemplo es el M-learning (Mobile learning), el cual ofrece un sistema educativo que se imparte a través de dispositivos móviles como los teléfonos celulares y los asistentes personales.

4.5.6 Mix-Learning

La etapa posterior al e-learning es la aplicación de una mezcla de sus herramientas con sistemas educativos tradicionales. La finalidad es dirigir más específicamente los contenidos a los estudiantes. Es así que el Blend Learning, Mix Learning o Hybrid Learning se presenta como la combinación efectiva de los diferentes modelos de reparto, modelos de enseñanza y modelos de aprendizaje.

4.5.7 Software Social

Teemu Leinonen habla de una quinta etapa, la del software social con contenido gratuito y abierto. Este tipo de servicio y aplicaciones vuelven a la idea de compartir información a través de Internet y a la colaboración conjunta, para mejorar los recursos educativos [12].

4.6 Objetos de Aprendizaje

Se puede creer que el objeto de aprendizaje (OA) es una entidad extensa, pero en realidad, debe desarrollar solamente un objetivo, debe contener actividades y una evaluación o comprobación final. Lo anterior puede ser variado según las necesidades específicas. Se trata de un material concreto, con contenidos y actividades de evaluación. Esto tiene el propósito de ampliar las

posibilidades de adaptar el objeto para varios cursos y contextos, a diferencia de un material que sea más extenso y abarque muchos contenidos.

El objeto de aprendizaje (OA) se puede visualizar, de manera sencilla, como un estándar con el cual se elabora material digital para ser utilizado en un curso en línea. Unificar el formato y la forma de los contenidos obedece a varios propósitos, entre ellos, tener a mano un banco de recursos que pueden ser utilizados, mejorados y adaptados a un determinado curso que contenga el tema tratado en un OA. El punto medular es optimizar la elaboración de material digital, pero, desde luego, es deseable que el formato del objeto permite la adaptación a los espacios virtuales.

4.6.1 ¿Por qué utilizar objetos de aprendizaje?

En primera instancia, el uso de los OA debe responder a una necesidad y a un deseo de mejorar las prácticas de la elaboración de materiales para la virtualidad.

De esa forma, se pueden aprovechar producciones previas y que no se tengan que diseñar materiales que ya están elaborados.

La enseñanza en la virtualidad experimenta problemáticas como la “ausencia de unas metodologías técnicas, documentales y psicopedagógicas comunes y aceptadas que garanticen los objetivos, interoperabilidad, durabilidad y reutilización de los materiales curriculares basados en las redes. Tales problemáticas se originan por el modo en que se presentan los contenidos de un curso. Por tanto, un OA elaborado adecuadamente, puede facilitar, paulatinamente, mejoras en la educación virtual.

La enseñanza en la virtualidad supone el uso de material digital y en Internet existen muchos recursos. Pero su uso muchas veces está limitado por su formato de soporte: precisamente, “el formato en el que se encuentran publicados; puesto que la obsolescencia de la tecnología con que fueron creados, aunada a la falta de modularidad y la ausencia de etiquetas para saber qué son y a quién pertenecen, dificultan su reutilización en un entorno puramente digital” [13].

4.7 Arquitectura de Ambientes Virtuales de Aprendizaje

Entre los aspectos que caracterizan una arquitectura de ambientes virtuales de aprendizaje se destacan 3 (ver Fig. 2): Herramientas de comunicación, Contenidos de aprendizaje y Administración de cursos, usuarios y contraseñas.

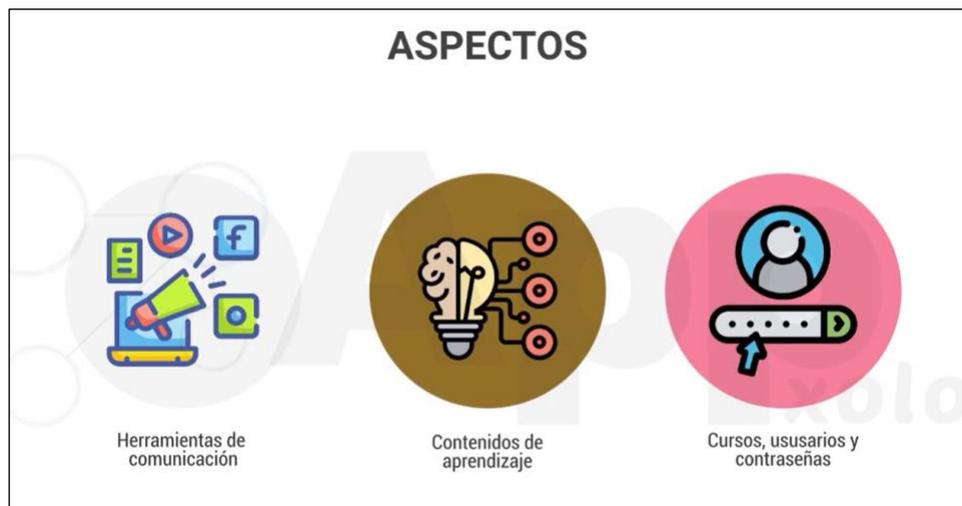


Fig. 2. Describe los aspectos, Arquitectura de ambientes Virtuales de Aprendizaje.

5 Metodología

5.1 Área de estudio

La evaluación de la metodología se lleva a cabo en una empresa que presta servicios tecnológicos, dentro del área de desarrollo y pruebas de software. Para los fines consiguientes se crearon cursos con el método tradicional sin el uso de las tecnologías de la información, el internet, o el modelo pedagógico tradicional. Posteriormente a esta fase se cambiará el rol del instructor y se aplicará la funcionalidad de un SGA con los mismos grupos para generar una comparación entre los resultados obtenidos.

Una vez que se han identificado los componentes de un SGA, se puede iniciar la fase de desarrollo, realizando un análisis de la metodología aplicable en dicho proceso.

Para el cumplimiento de este objetivo se utilizará la METODOLOGÍA DE DESARROLLO EN CASCADA: (denominado así por la posición de las fases en el desarrollo de esta, que parecen caer en cascada “por gravedad” hacia las siguientes fases), es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase. Este modelo fue el primero en originarse y es la base de todos los demás modelos de ciclo de vida.

La versión original fue propuesta por Winston W. Royce en 1970 y posteriormente revisada por Barry Boehm en 1980 e Ian Sommerville en 1985 [14].

Se identificaron los componentes a desarrollar e implementar.

- Componente de Seguridad (Usuarios, roles, funcionalidades y logs).
- Componente de Configuración (Parámetros y Catálogos).
- Componente de Gestión de cursos (Crear, Leer, Actualizar y Eliminar).
- Componente de Gestión de enrolamientos.
- Componente de Notas.
- Componente de Tareas.
- Componente de Notificaciones.
- Componente de Mensajería.

De los componentes detallados se los prioriza de acuerdo a la arquitectura del sistema y se les asignó un tiempo para su desarrollo por cada componente, con lo cual se respetó lo indicado en la metodología. Al finalizar el desarrollo de cada componente se realizaron revisiones comprobatorias para determinar los posibles errores y soluciones que se pueden ejecutar antes de pasar al desarrollo del siguiente componente.

5.2 Procedimiento

5.2.1 Método Aplicado.

Para poder ejecutar este proyecto de desarrollo se tomó en cuenta dos metodologías la primera de investigación y la segunda la de desarrollo, las mismas se detallan a continuación.

5.2.2 Método de Investigación

El método elegido para realizar este trabajo de titulación es el deductivo, ya que es una forma de razonamiento lógico que se utiliza para llegar a una conclusión a partir de una serie de premisas o afirmaciones que se consideran verdaderas o aceptadas. Este método se basa en la lógica formal y consiste en aplicar reglas de inferencia para obtener una conclusión que se deduce necesariamente de las premisas.

En el método deductivo, se parte de una proposición general o premisa mayor, que se asume como verdadera, y se le añade una premisa menor, que se deriva de la observación o experiencia. A partir de estas dos premisas, se deduce una conclusión lógica que es necesariamente verdadera, siempre y cuando las premisas sean verdaderas.

5.2.3 Método de desarrollo en cascada para la construcción del Sistema de Gestión de Aprendizaje.

Para la implementación del proyecto utilizando el método de desarrollo en cascada se dividieron en 5 etapas. Esto principalmente se debe a que algunas de las etapas se fusionan en una sola, quedando de la siguiente manera:

1. Análisis.
2. Diseño.
3. Implementación.
4. Verificación.
5. Mantenimiento.

Se puede adaptar estas fases según lo requiera el proyecto de software que estás desarrollando a través del ciclo de vida en cascada, siempre y cuando se tomen en cuenta todos los puntos importantes dentro de las 5 etapas principales [15].

En la Fig. 3 se describen las etapas de este proceso de desarrollo.

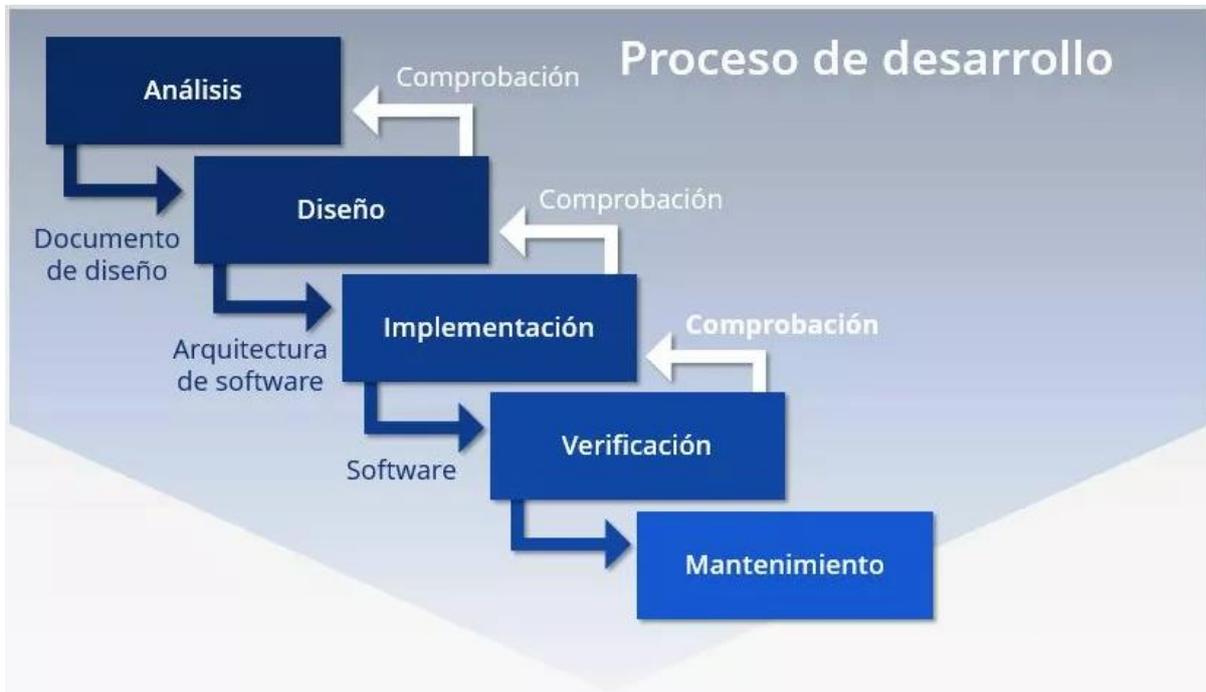


Fig. 3. Descripción de las 5 etapas principales dentro de la metodología de desarrollo en cascada.

5.2.4 Integración de ICONIX dentro del proceso de desarrollo del SGA

Aunque la metodología de desarrollo en cascada y la metodología ICONIX difieren en su enfoque general, se integró ICONIX en el proceso de desarrollo en cascada utilizando técnicas de modelado para producir modelos y prototipos detallados de cada fase del ciclo de vida del software. De esta manera, detectamos problemas de forma temprana en el proceso y se aplicaron cambios en el diseño antes de pasar a la siguiente fase.

Para lograr un correcto y óptimo desarrollo de un sistema de software, se aplicó un proceso de desarrollo de software que permitió abordar de manera organizada las diferentes etapas del proyecto. Por lo cual, la metodología ICONIX es adoptada para la ejecución del proyecto ya que es una metodología de desarrollo de software iterativa e incremental que se enfoca en la construcción de prototipos funcionales y la validación temprana de requisitos. La metodología ICONIX se basa en el modelo de desarrollo de software orientado a objetos (OO) y se utilizó

técnicas como el diseño de diagramas de clases y la programación orientada a objetos (OOP) para diseñar y construir el SGA. La Fig. 4 muestra una descripción detallada del proceso ICONIX.

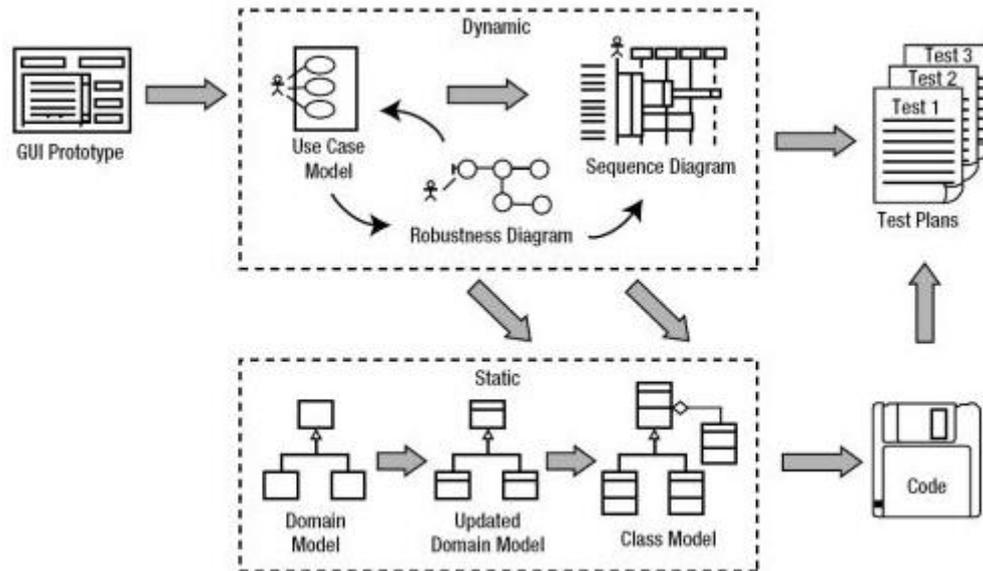


Fig. 4. Descripción del proceso ICONIX.

El enfoque de la metodología ICONIX permitió dividir el proceso de desarrollo en cuatro fases principales: Requisitos, Diseño, Implementación y Pruebas. En cada fase se definió un conjunto de objetivos y entregables específicos que se completaron antes de avanzar a la siguiente fase:

- **Fase de Requisitos:** En esta fase, el equipo de desarrollo trabajó con los usuarios y otros interesados para comprender los requisitos del sistema. Los objetivos de esta fase fueron definir los requisitos, crear modelos de casos de uso y realizar una validación temprana de los requisitos.
- **Fase de Diseño:** En esta fase, el equipo de desarrollo creó modelos de diseño y prototipos funcionales del sistema. Los objetivos de esta fase fueron crear diagramas de clases y de secuencia, diseñar la arquitectura del sistema y crear prototipos funcionales.
- **Fase de Implementación:** En esta fase, el equipo de desarrollo implementó el sistema a partir del diseño creado en la fase anterior. Los objetivos de esta fase fueron escribir y probar el código, y crear documentación.
- **Fase de Pruebas:** En esta fase, el equipo de desarrollo realizó validaciones al sistema completo para garantizar que cumpla con los requisitos del usuario y que funcione de

manera efectiva. Los objetivos de esta fase fueron realizar pruebas de unidad, pruebas de integración y pruebas de aceptación.

5.2.4.1 Fase de requisitos

La fase de requisitos en la metodología ICONIX tiene como objetivo comprender y definir los requisitos del sistema a desarrollar de manera clara y precisa. Esta fase es crucial porque establece la base para el resto del proceso de desarrollo de software.

La primera tarea en esta fase fue identificar los interesados en el proyecto, como los usuarios finales, el equipo de desarrollo, los patrocinadores del proyecto, los dueños del negocio, entre otros, y definir sus necesidades y expectativas. Esto se puede lograr a través de entrevistas, encuestas y otros métodos de investigación.

Luego se procede a la elaboración de un documento de requisitos que incluye una descripción detallada de las funcionalidades y características que el sistema debe tener. Este documento debe ser claro, preciso y comprensible para todos los interesados en el proyecto. Para ello, se utilizaron herramientas como:

- Historias de usuario
- Casos de uso
- Prototipos
- Diagramas de flujo
- Diagramas de componentes

Una vez que se tiene los documentos de requisitos, es necesario validarlos con los interesados en el proyecto para asegurarse de que se han comprendido correctamente y que se han identificado todas las necesidades del sistema. La validación se puede realizar a través de reuniones de revisión o presentaciones del documento de requisitos.

5.2.4.2 Fase de diseño

En esta fase el objetivo fue transformar los requisitos identificados en la fase anterior en una arquitectura detallada y en un diseño de componentes que cumpla con los requisitos funcionales y no funcionales del sistema.

Durante la fase de diseño, se definió la estructura de clases y objetos, las interfaces de usuario, los diagramas de secuencia y colaboración, y los artefactos necesarios para describir la arquitectura del sistema. Además, se identificaron los patrones de diseño apropiado para el sistema (Modelo Vista Controlador) y definir el modelo de datos y la estrategia de persistencia.

El objetivo principal de la fase de diseño fue el crear una especificación detallada y completa del sistema que se utilizó como guía en la implementación. Esta especificación incluyó la suficiente información para permitir la construcción del sistema sin ambigüedades, sin suposiciones incorrectas y sin información faltante.

Además, el validar que la arquitectura propuesta y el diseño de componentes cumplan con los requisitos y objetivos del sistema. La validación se puede llevar a cabo a través de revisiones técnicas, pruebas de concepto y simulaciones.

5.2.4.3 Fase de implementación

En la implementación se llevó a cabo la construcción del software basado en las especificaciones y diseños definidos en las fases anteriores del proceso.

En esta fase, se desarrollan los componentes del software utilizando las herramientas y tecnologías apropiadas (Framework Laravel como base), y se lleva a cabo la integración de los diferentes módulos y componentes para formar una aplicación completa. Se llevan a cabo también pruebas de integración y pruebas unitarias para asegurar que los componentes individuales funcionen correctamente y se integren adecuadamente.

Se incluye también la configuración y el despliegue de la aplicación en un entorno de producción, lo que implica la instalación de software y la configuración de servidores y otros recursos necesarios para su correcto funcionamiento.

Durante esta fase es importante mantener una comunicación cercana con el equipo de desarrollo para garantizar que se cumplan los requisitos y las expectativas del cliente y que el software se construya de manera eficiente y efectiva. Se llevarán a cabo pruebas rigurosas para asegurar de que se aborden cualquier problema o error que se encuentre antes de pasar a la fase de pruebas finales de certificación y entrega al cliente.

5.2.4.4 Fase de pruebas

La fase de pruebas en la metodología ICONIX es crucial para garantizar la calidad del software desarrollado. El objetivo principal de esta fase es validar y verificar que el sistema cumpla con todos los requisitos y especificaciones establecidos en las fases anteriores.

Para lograr este objetivo, la fase de pruebas se divide en diferentes etapas, cada una con su propio objetivo específico:

- **Pruebas de unidad:** en esta etapa se prueban las unidades de código individuales para asegurarse de que funcionen correctamente. El objetivo es encontrar y corregir errores de programación y asegurarse de que cada unidad de código trabaje como se espera.
- **Pruebas de integración:** en esta etapa, se prueban las diferentes unidades de código juntas para asegurarse de que funcionen correctamente como un sistema integrado. El objetivo es encontrar errores de integración entre las unidades de código y corregirlos.
- **Pruebas de sistema:** en esta etapa, se prueba el sistema completo para asegurarse de que cumple con todos los requisitos y especificaciones. El objetivo es encontrar errores en la funcionalidad del sistema y corregirlos.
- **Pruebas de aceptación:** en esta etapa, se prueba el sistema completo en un entorno de producción simulado para asegurarse de que cumple con los requisitos del usuario y que el usuario final puede utilizarlo de manera efectiva. El objetivo es encontrar errores de usabilidad y corregirlos.

5.2.5 Uso del Patrón Modelo Vista Controlador (MVC)

El uso del patrón Modelo Vista Controlador (MVC) en la implementación fue importante porque proporciona una estructura organizada y separada de las responsabilidades en la aplicación, lo que facilita el mantenimiento y la evolución de la misma (ver Fig. 5).

El patrón Modelo Vista Controlador, separa la parte gráfica de una aplicación, de los procesos lógicos y de los datos de la misma [16].

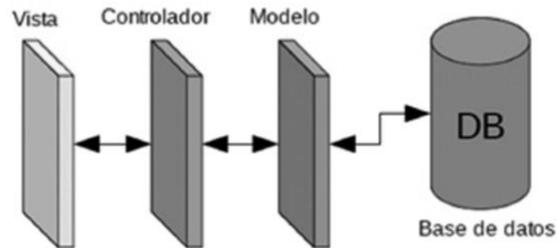


Fig. 5. Esquema de capas del patrón modelo vista controlador.

5.2.5.1 La vista (view)

Permitió la organización de la interfaz gráfica, es con lo que el usuario de la aplicación interactúa y es donde se reciben las “órdenes” y se presentan los resultados de los procesos al usuario.

5.2.5.2 El controlador (controller)

Actuó como intermediario entre el modelo y la vista, procesando las solicitudes del usuario y actualizando el modelo y la vista en consecuencia.

5.2.5.3 El modelo (model)

Es la representación específica de la información con la que se está operando en el instante de la ejecución de la aplicación, representó variables, objetos, datos en general que se están modificando de acuerdo a lo que el usuario solicite [16].

La Fig. 6 destaca la interacción de los diferentes componentes de la arquitectura modelo vista controlador.

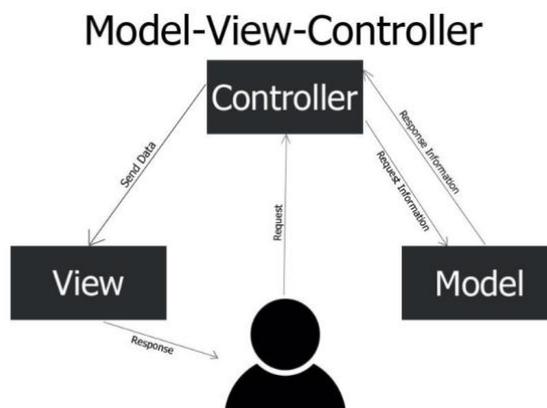


Fig. 6. Descripción del MVC según Joseph Spinelli.

6 Resultados

En esta sección, se detallan los resultados obtenidos en la construcción del Sistema de Gestión de Aprendizaje, que tiene como objetivo principal el centralizar contenidos académicos y facilitar la exposición de recursos formativos, se detalla el proceso de diseño y desarrollo del sistema, así como los desafíos enfrentados durante su implementación.

6.1 Objetivo 1: Identificar los requerimientos y módulos que contempla el sistema de gestión de aprendizaje a través de la revisión de literatura y de soluciones establecidas en el mercado.

6.1.1 Lista de requerimientos funcionales principales

Para identificar los requisitos de software se realizó un análisis de mercado de las herramientas implementadas como: Udemy, Crehana y Platzi, producto de esta actividad se identificaron los requisitos detallados a continuación. En la Tabla I se describen los requerimientos funcionales del estudiante.

El sistema debe permitir al estudiante:

TABLA I
REQUERIMIENTOS FUNCIONALES ESTUDIANTE

Código	Descripción	Categoría	Prioridad
RF001	Crear un registro al sitio ingresando: <ul style="list-style-type: none">● Nombres,● Apellidos,● Tipo de identificación,● Número de identificación,● Correo electrónico,● Número de celular	Visible	Alta
RF002	El registro de cliente debe ser único mediante número de identificación	Oculto	Alta
RF003	Iniciar sesión dentro del sitio mediante el ingreso de: <ul style="list-style-type: none">● Usuario (correo electrónico)● Clave de acceso	Visible	Alta
RF004	Visualizar los cursos a los que está enrolado el estudiante.	Visible	Alta
RF005	Ingresar dentro de un curso previamente enrolado para poder iniciar o continuar su capacitación.	Visible	Alta
RF006	Visualizar el resumen generado por participación dentro de un curso enrolado.	Visible	Alta
RF007	Buscar cursos disponibles que se encuentran en el sitio a través del:	Visible	Alta

	<ul style="list-style-type: none"> ● Nombre ● Descripción 		
RF008	<p>Enrolarse a un curso determinado de acuerdo a:</p> <ul style="list-style-type: none"> ● Nombre del curso <p>Registro del estudiante al curso se encuentre en estado</p> <ul style="list-style-type: none"> ○ ENROLADO, ○ PENDIENTE APROBAR 	Visible	Alta
RF009	<p>Eliminar Enrolamiento de un curso,</p> <ul style="list-style-type: none"> ● Curso enrolado previamente ● Registro del estudiante a un curso se encuentre en estado ○ ENROLADO, ○ PENDIENTE APROBAR 	Visible	Alta
RF010	Listar los cursos enrolados por el estudiante	Visible	Media

En la Tabla II se describen los requerimientos funcionales del instructor.
El sistema debe permitir al **instructor**:

TABLA II
REQUERIMIENTOS FUNCIONALES INSTRUCTOR

Código	Descripción	Categoría	Prioridad
RF011	<p>Crear un nuevo curso</p> <ul style="list-style-type: none"> ● Nombre ● Resumen 	Visible	Alta
RF012	<p>Actualizar la información de un curso registrado previamente.</p> <ul style="list-style-type: none"> ● Título ● Idioma ● Palabras claves ● Información ● Requisitos ● Secciones <ul style="list-style-type: none"> ● Lecciones ● Recursos: Imágenes/Imagen portada ● Tráiler (opcional) 	Visible	Alta
RF013	Publicar un curso creado y actualizado previamente	Visible	Alta
RF014	Deshabilitar/Habilitar un curso registrado previamente.	Visible	Media
RF015	Eliminar un curso registrado previamente.	Visible	Alta
RF016	Responder preguntas registradas a un curso.	Visible	Alta
RF017	Responder a inquietudes registradas por estudiantes dentro de una sección.	Visible	Alta

RF018	Deshabilitar el ingreso de un estudiante a un curso registrado previamente.	Visible	Media
RF019	Eliminar el enrolamiento de un estudiante registrado dentro de un curso.	Visible	Alta
RF020	Registrar un anuncio dentro de un curso	Visible	Media

En la Tabla III se describen los requerimientos funcionales del administrador del sistema. El sistema debe permitir al **administrador del sistema (ROOT)**:

TABLA III
REQUERIMIENTOS FUNCIONALES ADMINISTRADOR

Código	Descripción	Categoría	Prioridad
RF021	Iniciar sesión con sus credenciales de cuenta: <ul style="list-style-type: none"> ● Usuario (correo electrónico) ● Clave 	Visible	Alta
RF022	Actualizar (agregar o quitar) roles usuario de sistema	Visible	Alta
RF023	Actualizar (agregar o quitar) funcionalidades del sistema		
RF024	Actualizar (agregar o quitar) usuarios de sistema		
RF025	Desactivar cuentas de usuarios.	Inferido	Alta
RF026	Buscar usuarios por <ul style="list-style-type: none"> ● Su número de identificación, ● Nombres y apellidos. 	Inferido	Media
RF027	Crear cuentas de usuario a sujetos ya existentes	Inferido	Media

6.1.2 Lista de requerimientos no funcionales

En la Tabla IV se describen los requerimientos no funcionales.

TABLA IV
REQUERIMIENTOS NO FUNCIONALES

Código	Categoría	Descripción	Prioridad
RNF01	Usabilidad	El tiempo de inducción del software para un usuario deberá ser menor a 4 horas.	Alta
RNF02		La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tablets y teléfonos inteligentes.	Alta
RNF03		El sistema debe poseer interfaces gráficas amigables y bien formadas.	Alta
RNF04		La aplicación debe proporcionar mensajes de aviso y de inconvenientes que sean informativos y orientados al usuario final.	Alta

RNF05	Eficiencia	El tiempo de respuesta para solicitudes debe ser menor a 9 segundos.	Alta
RNF06		El software debe ser capaz de operar adecuadamente con un máximo de 500 usuarios con sesiones concurrentes.	Alta
RNF07	Seguridad	Los permisos de acceso al sistema podrán ser cambiados solamente por el usuario que tenga el rol de administrador.	Alta
RNF08		Todas las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema deben ser a través del protocolo HTTPS .	Alta
RNF09		Las claves de usuario deben estar encriptadas utilizando algoritmo RSA dentro del motor de base de datos.	Alta
RNF10	Mantenibilidad	Debe ser de fácil mantenimiento y corrección de bugs	Baja
RNF11	Disponibilidad	El sistema debe tener una disponibilidad al menos de un 99% en un año	Baja
RNF12		El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.	Media
RNF13	Portabilidad	La nueva aplicación debe manejar fuentes del alfabeto en inglés y español.	Baja
RNF14		La interfaz de usuario será implementada para navegadores web.	Alta

6.1.3 Modelo del dominio

Finalizada la actividad de identificación de requisitos, se realizó un análisis/revisión gramatical a la tabla de requisitos funcionales (ver Tabla 3) para identificar sustantivos o clases sustantivas. La figura 7, detalla el modelo de dominio inicial por paquete

Modelo del dominio inicial

La Fig. 7 describe el diagrama del modelo de dominio inicial referente al componente de cursos.

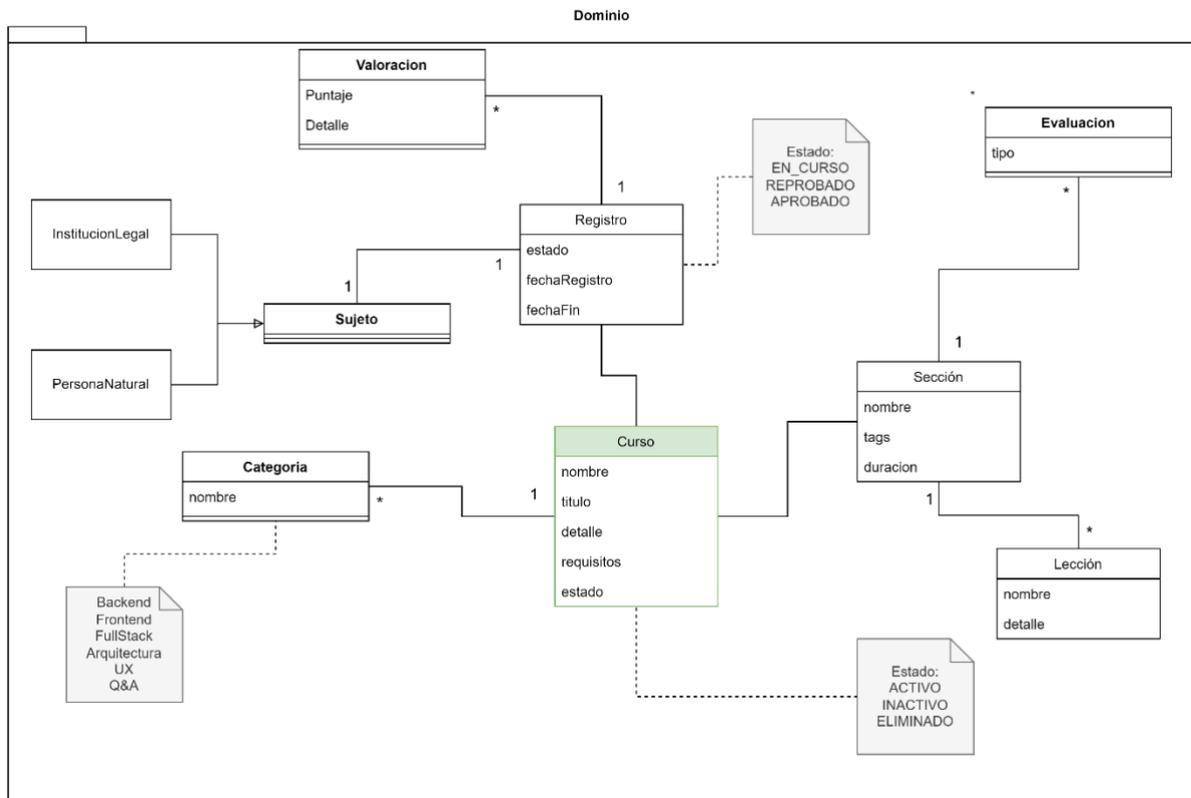


Fig. 7. Diagrama que muestra el modelo de dominio inicial.

La Fig. 8 muestra el modelo de dominio con respecto a la seguridad de la aplicación

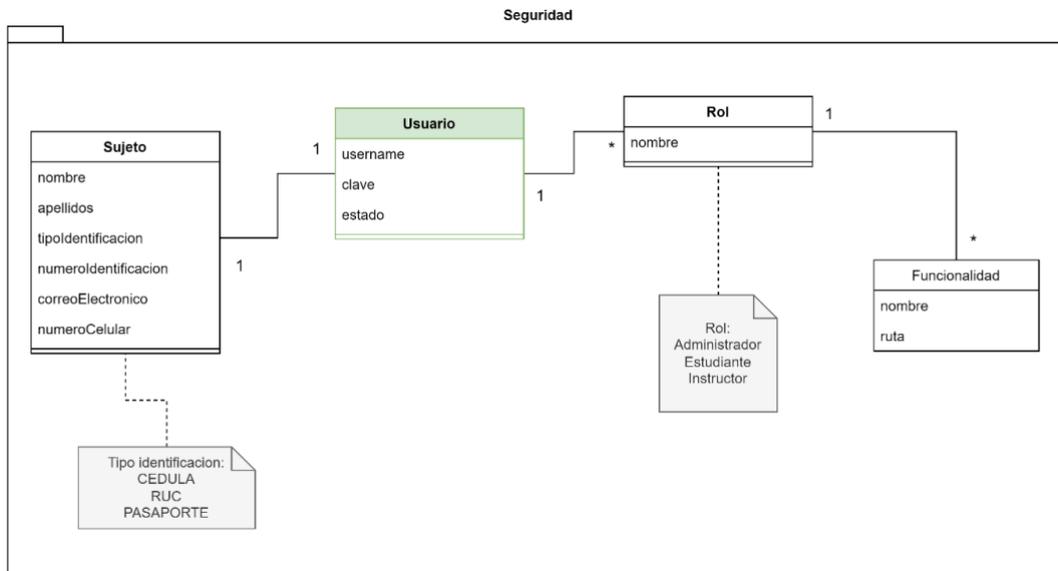


Fig. 8. Diagrama que muestra el modelo de dominio aplicado en la seguridad del sistema.

6.1.4 Diccionario de datos

A continuación, se detalla el diccionario de datos (ver Tabla V).

TABLA V
DICCIONARIO DE DATOS

Término	Descripción
Rol	Se refiere a un conjunto de permisos y responsabilidades asignados a un usuario o grupo de usuarios en un sistema. Un rol define lo que un usuario puede o no puede hacer dentro de un sistema y ayuda a limitar el acceso a las funciones y datos del sistema en función de las necesidades y responsabilidades de ese usuario o grupo.
Funcionalidad	Se refiere a una capacidad o acción que un sistema es capaz de realizar. Las funcionalidades pueden ser tan simples como la capacidad de mostrar una lista de elementos o tan complejas como la capacidad de realizar un procesamiento de datos complejo. En un sistema informático, las funcionalidades se dividen generalmente en módulos o componentes, cada uno con su conjunto específico de

	funcionalidades. Estos módulos se combinan para proporcionar una funcionalidad completa del sistema
Usuario	Es una persona, cuenta o entidad que utiliza un sistema para realizar tareas y acceder a recursos. Los usuarios interactúan con el sistema mediante una interfaz de usuario, que puede ser una aplicación de escritorio, una aplicación web, una aplicación móvil u otro tipo de interfaz.
Estudiante	Persona que usa los servicios del Sistema de Gestión de Aprendizaje.
Curso	Corresponde a un componente (material educativo) que puede ser ofertado dentro del sitio web para su enrolamiento.
Sección	Corresponde al componente del cual está conformado un curso, permite el registro de texto, imágenes o videos.
Sujeto	Puede ser definido como una persona natural o una institución legal, actor que interactúa con el sistema, ya sea para ingresar, procesar o recibir información. Un sujeto puede acceder al sistema a través de dispositivos como una computadora, un teléfono móvil o una tableta y realizar diversas acciones según los permisos y el rol asignado en el sistema. Algunas de las acciones comunes que un sujeto puede realizar en un sistema informático incluyen la inserción de datos, la ejecución de funcionalidades, la modificación de configuraciones, la consulta de información y la realización de transacciones.

6.1.5 Modelo del caso de uso

En este proceso se identificó los actores, módulos del sistema y componentes realizando un análisis de las funcionalidades en cada requisito para determinar así los diferentes casos de uso.

Sistema:

Desarrollo de un Sistema de Gestión de Aprendizaje como herramienta que permita organizar, centralizar contenidos académicos y facilitar la exposición de recursos formativos.

Módulo:

- Cursos

Identificación de actores:

- Instructor
- Estudiante
- Administrador del sistema

Identificación de casos de uso (Funcionalidad)

En la Tabla VI se detallan los casos de uso que aplican a todos los usuarios independiente de su rol.

Actor: Todos los Usuarios (Estudiante, Instructor, Administrador del Sistema)

TABLA VI
CASOS DE USO DE TODOS LOS USUARIOS

Funcionalidad (Casos de Uso)	Código	Requerimientos Cubiertos
Iniciar Sesión (Ingresar al sistema)	CU-001	RF033
Cerrar Sesión (Salir del sistema)	CU-002	RF035
Registrar su cuenta de usuario	CU-003	RF034
Actualizar clave	CU-004	RF036
Actualizar su perfil de usuario	CU-005	RF037
Registrar su información personal	CU-006	RF011

En la Tabla VII se detallan los casos de uso que aplican a los usuarios cuyo rol sea el de estudiante.

Actor: Estudiante

TABLA VII
CASOS DE USO DEL ESTUDIANTE

Funcionalidad (Casos de Uso)	Código	Requerimientos Cubiertos
Visualizar cursos disponibles	CU-007	RF004
Visualizar cursos enrolados	CU-008	RF010
Realizar enrolamiento en curso	CU-009	RF008
Eliminar enrolamiento	CU-010	RF009

Ingresar a curso para iniciar aprendizaje	CU-011	RF014
Generar resumen aprendizaje de curso	CU-012	RF006
Buscar cursos	CU-013	RF007

En la Tabla VIII se detallan los casos de uso que aplican a los usuarios cuyo rol sea el de instructor.

Actor: Instructor

TABLA VIII
CASOS DE USO DEL INSTRUCTOR

Funcionalidad (Casos de Uso)	Código	Requerimientos Cubiertos
Crear curso	CU-014	RF011
Editar/Actualizar curso	CU-015	RF012
Publicar curso	CU-016	RF013
Habilitar/Deshabilitar curso	CU-017	RF014
Eliminar curso	CU-018	RF015
Responder preguntas	CU-019	RF016, RF017
Deshabilitar ingreso de estudiante a curso	CU-020	RF018
Eliminar enrolamiento	CU-021	RF019
Registrar anuncio en curso	CU-022	RF020

En la Tabla IX se detallan los casos de uso que aplican a los usuarios cuyo rol sea el de administrador del sistema.

Actor: Administrador del sistema

TABLA IX
CASOS DE USO DEL ADMINISTRADOR DEL SISTEMA

Funcionalidad (Casos de Uso)	Código	Requerimientos Cubiertos
Administrar Cuentas de usuario	CU-023	RF001, RF002, RF003

6.1.6 Diagramas de caso de uso

Diagrama de paquetes

En la Fig. 9 se detallan todos los paquetes clasificados por módulos del sistema.

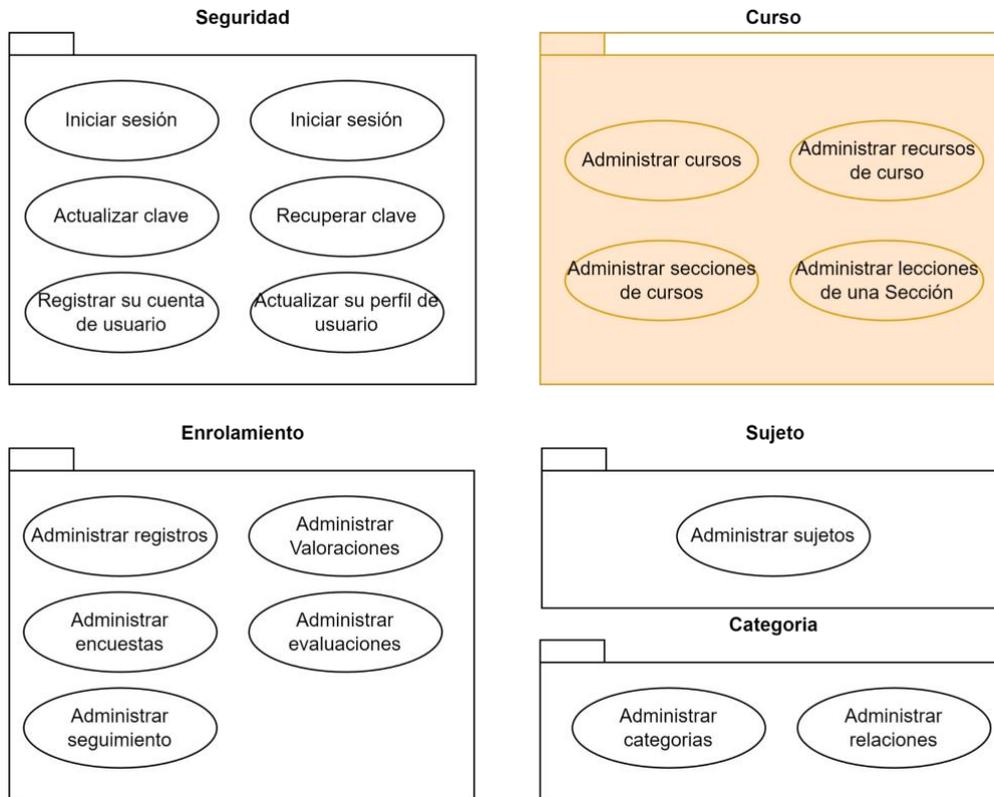


Fig. 9. Diagrama que detalla los principales paquetes que componen el SGA.

Diagramas de caso de Uso del Sistema

En la Fig. 10 se detallan los casos de uso que forman parte del componente cursos, como estudiante: visualizar cursos registrados, enrolar curso, como instructor: Administrar cursos, Gestionar enrolamientos, Gestionar inquietudes.

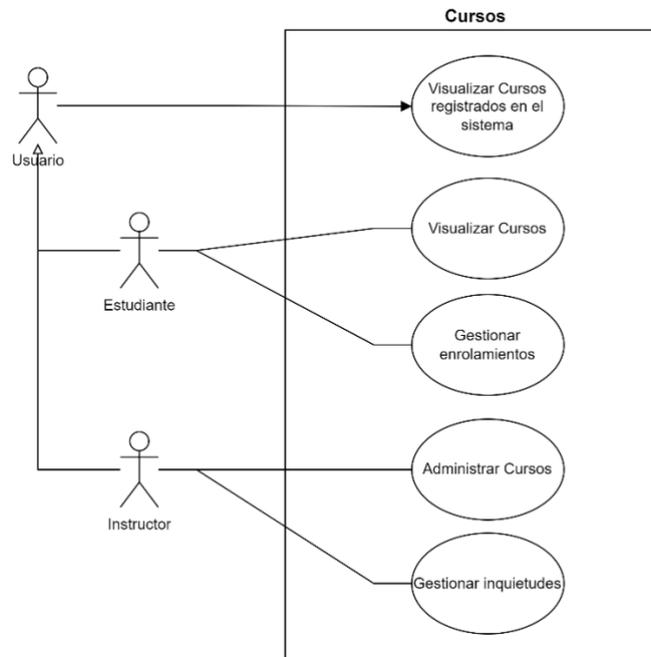


Fig. 10. Diagrama que detalla la administración de cursos por rol del sistema.

6.1.7 Descripción de Casos de Uso

En esta sección se detallan casos de uso seleccionados por su relevancia en el sistema.

Visualizar cursos

La Fig. 11 detalla el prototipo referente al caso de uso: Visualizar curso, que fue usado para la determinación y validación del requerimiento. Además, la descripción se encuentra descrita en la Tabla X.

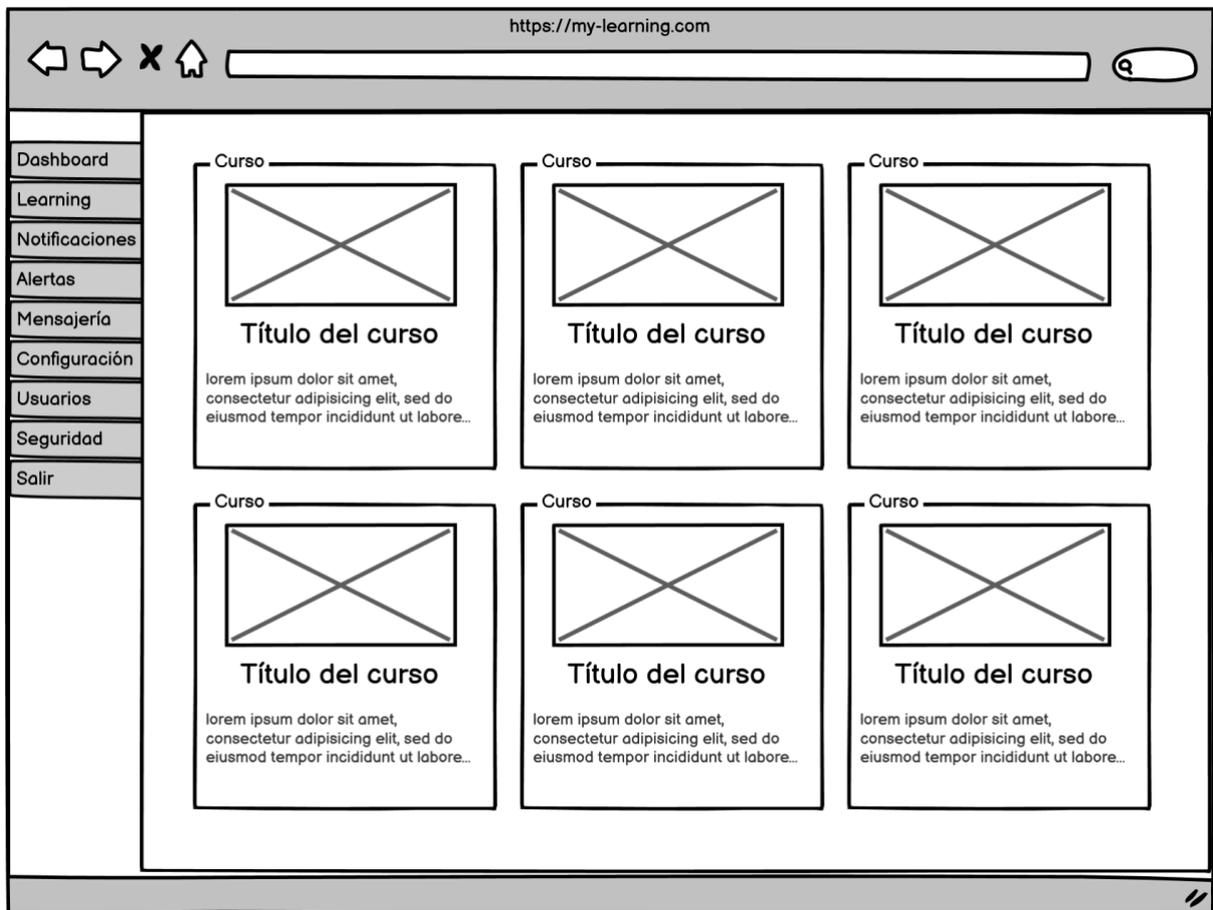


Fig. 11. Prototipo que muestra la lista de cursos en el SGA.

TABLA X
DESCRIPCIÓN DEL CASO DE USO VISUALIZAR CURSOS

Caso de Uso:	VISUALIZAR CURSOS	Actor(es):	Estudiante
Código:	CU-007	Pantalla(s):	P001
Ref. Req:	RF004		
Resumen / Descripción:	El sistema deberá permitir visualizar el catálogo de los cursos disponibles en el sistema.		
Objetivos:	<ul style="list-style-type: none"> ● Visualizar los cursos ofertados en el sistema 		
Precondiciones:	<ul style="list-style-type: none"> ● El usuario (GENERAL) debe estar dentro del sistema. ● El sistema debe haber mostrado la página de inicio. 		
Postcondiciones:	<ul style="list-style-type: none"> ● El sistema mostrará la acción realizada por el usuario. 		
CURSO NORMAL DE EVENTOS			
Visualizar Cursos			
<ol style="list-style-type: none"> 1. El usuario ingresa a la página Inicio. 2. El sistema muestra el menú. 3. El usuario selecciona la opción LEARNING - CURSOS DISPONIBLES. 4. El sistema muestra en la pantalla con la lista de cursos registrados. 5. El sistema muestra el listado de 10 cursos por pantalla y los muestra en la tabla de resultados. 6. El usuario selecciona la tarjeta del Curso. 			

CURSO ALTERNO DE EVENTOS

4. Buscar por criterio

- 4.1. El usuario ingresa el nombre, tags referentes a un tema.
- 4.2. El usuario presiona el botón buscar.
- 4.3. El sistema busca los cursos disponibles acorde a los criterios ingresados.
- 4.4. El sistema muestra el listado de cursos disponibles acordes a los criterios aplicados.

Diagrama de Robustes

Este tipo de diagrama permitió identificar el comportamiento de un sistema en términos de su capacidad para manejar situaciones excepcionales o inesperadas en la funcionalidad de: Visualizar cursos (ver Fig. 12).

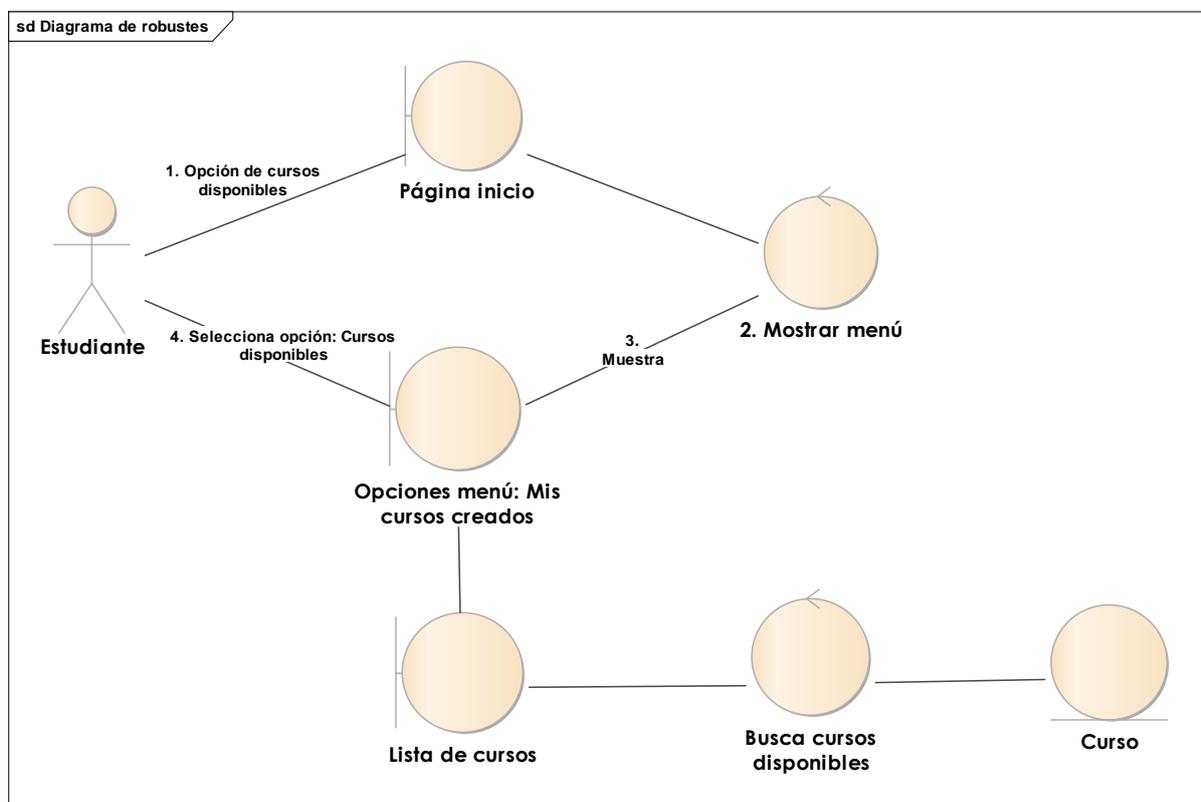


Fig. 12. Diagrama de robustes que describe el caso de uso: Visualizar curso.

6.1.8 Administrar un Curso

El diseño del prototipo permite ordenar las ideas, explorar diferentes caminos de concepto o diseño, y detectar posibles problemas o carencias antes de empezar la fase de implementación, en Fig. 13 se detalla la Administración de un curso y en Fig. 14 la Administración de secciones de un curso.

https://my-learning.com

General | Información | Requisitos | Recursos | Secciones

Dashboard

Learning

Notificaciones

Alertas

Mensajería

Configuración

Usuarios

Seguridad

Salir

Nombre

Titulo

URL video trailer

Idioma

Tags

Resumen

Actualizar

Fig. 13. Prototipo que muestra el caso de uso: Administrar curso.

https://my-learning.com

Secciones

Nombre

Resumen

Registrar

Nombre	Clases	Descripción	v
Sección 1	40	Peldi	⊙
Sección 2	38		☑
Marco Botton			
Tuttofare			
Mariah MacIachlan	41	Patata	☐
Better Half			
Valerie Liberty		Val	☑
Head Chef			

Fig. 14. Prototipo que muestra el caso de uso: Administrar secciones.

La descripción del caso de uso: CU-010 Administrar un curso se detalla en la Tabla XI.

TABLA XI
DESCRIPCIÓN DEL CASO DE USO ADMINISTRAR UN CURSO

Caso de Uso:	ADMINISTRAR UN CURSO	Actor(es):	Cajero
Código	CU-010	Pantalla(s):	P002, P003
Ref. Req:	RF026, RF027		
Resumen / Descripción:	El sistema deberá permitir la administración (Crear, Actualizar, Eliminar, Buscar) y gestión de cursos.		
Objetivos:	Administrar el CRUD (Create, Read, Update, Delete) de los menús en el sistema.		
Precondiciones:	El usuario debe haber iniciado sesión.		
Postcondiciones:	El sistema mostrará la acción realizada por el usuario.		
CURSO NORMAL DE EVENTOS			
1. Buscar cursos			
1.1. El instructor ingresa a la página Inicio.			
1.2. El sistema muestra el menú de navegación.			
1.3. El instructor selecciona la opción LEARNING - MIS CURSOS CREADOS .			
1.4. El sistema muestra en la pantalla con la lista de cursos registrados por el instructor.			
1.5. El sistema muestra el listado de 10 cursos por pantalla y los muestra en la tabla de resultados.			
1.6. El instructor realiza la búsqueda del curso, utilizando el control de paginación.			
1.7. El instructor selecciona el curso a administrar.			
CURSO ALTERNO DE EVENTOS			
2. Nuevo curso			
2.1. El usuario presiona: Crear curso.			
2.2. El usuario ingresa el nombre y el resumen.			
2.3. El usuario presiona el botón Registrar .			
2.4. El sistema valida y registra la información ingresada en el sistema.			
2.5. El sistema muestra la pantalla: Editar la información del Curso .			
3. Editar Curso			
3.1. El instructor presiona el botón EDITAR un curso ubicado en la tabla de resultados de la pantalla Gestión de cursos .			
3.2. En la pantalla Editar Curso , el instructor procede a actualizar la información del curso.			
3.3. El instructor, ingresa la información faltante dentro de la pantalla Editar Curso			
3.4. El sistema presenta un mensaje de confirmación.			
3.5. El flujo continúa en el paso 2.5 del curso alterno de eventos			
4. Eliminar curso			
4.1. El instructor, presiona el botón Eliminar en la pantalla Gestión de cursos .			
4.2. El sistema muestra mensaje de confirmación de eliminación.			

- 4.3. El instructor presiona el botón **Sí** del mensaje de confirmación.
 - 4.4. El sistema elimina el curso seleccionado.
 - 4.5. El sistema remueve el curso de la tabla de resultado de la pantalla **Gestión de cursos**.
- 5. Datos no válidos.**
- 5.1. El sistema muestra un mensaje de error indicando que los datos no son válidos.
 - 5.2. El flujo continúa en el **paso 2.4** de la sección **Nuevo Curso**.
- 6. El curso no se puede eliminar.**
- 6.1. El sistema muestra un mensaje de error indicando que el curso no se puede eliminar, debido a que ya ha sido enrolado por un estudiante.

Diagrama de Robustez

El objetivo del diagrama de robustez es añadir nuevas relaciones a los diagramas de clase, en Fig. 15 se detalla el diagrama del caso de uso: Administrar curso.

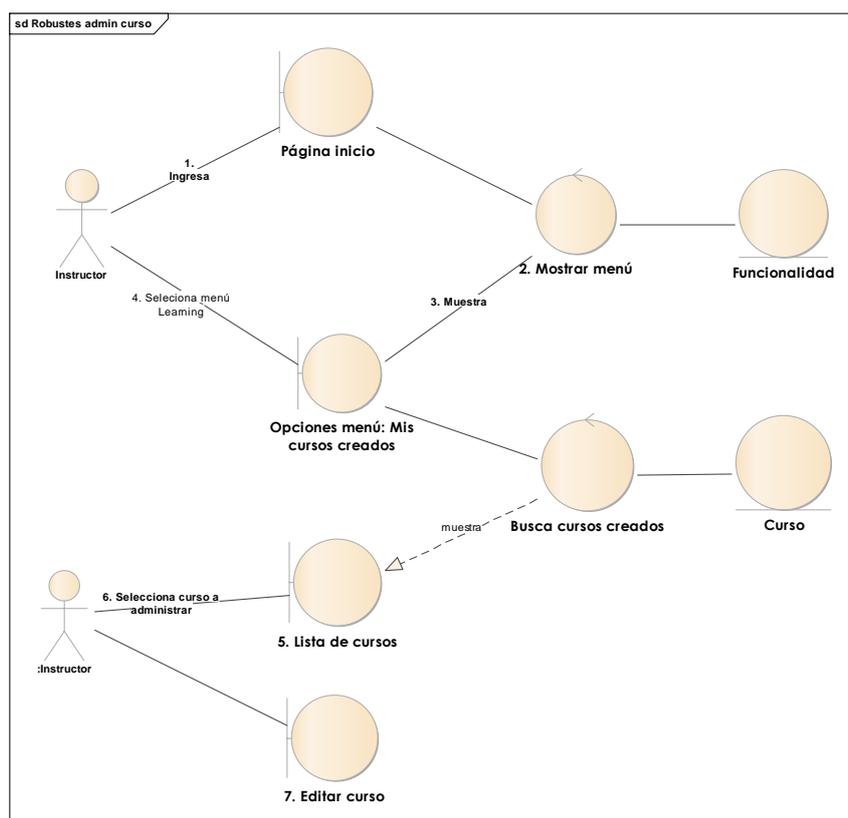


Fig. 15. Diagrama de robustez, caso de uso: Administrar curso.

Realizar enrolamiento en curso

En Fig. 16 se diseña el prototipo referente al detalle del curso, muestra la información sobre un curso.

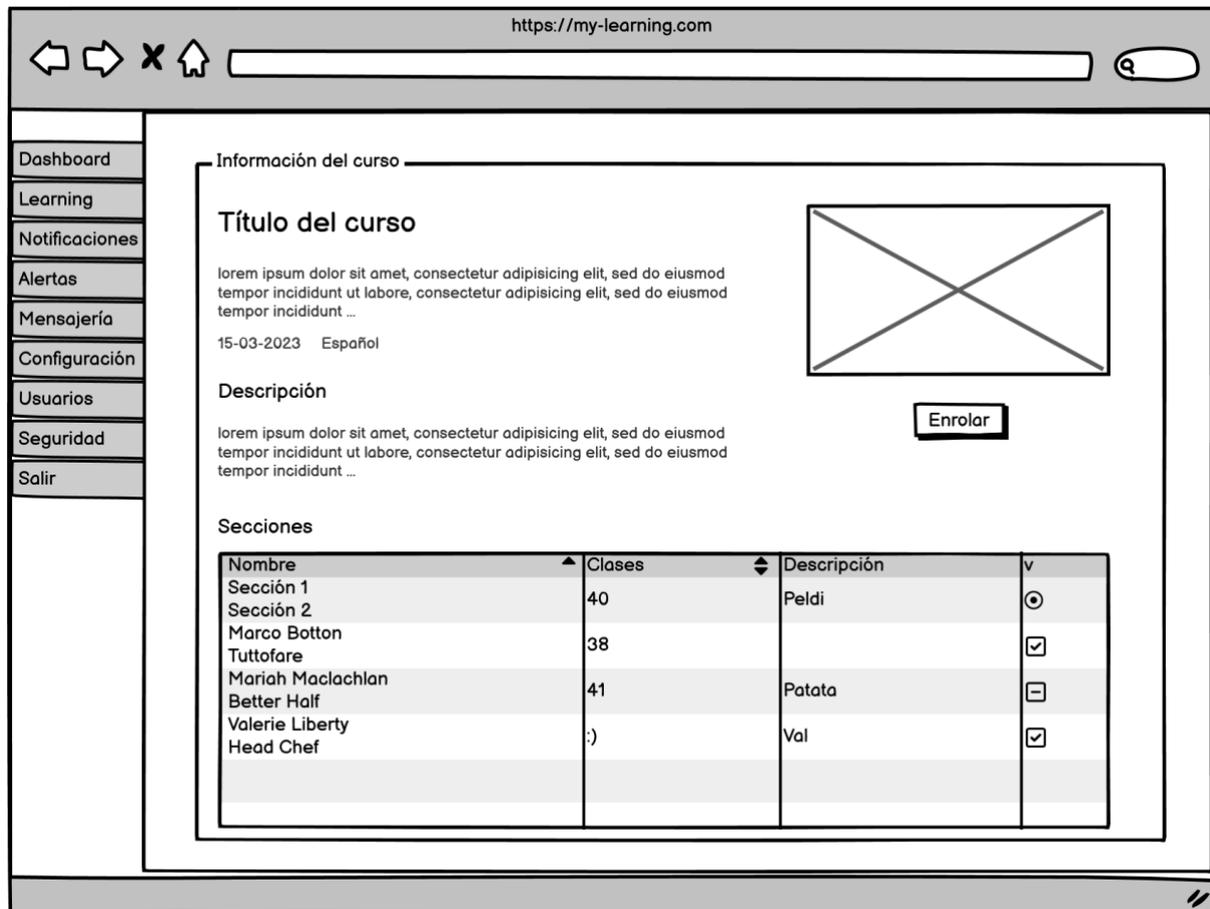


Fig. 16. Prototipo que muestra la información del curso y enrolamiento.

La descripción del caso de uso CU-009 se describe en la Tabla XII.

TABLA XII
DESCRIPCIÓN UC REALIZAR ENROLAMIENTO EN CURSO

Caso de Uso:	REALIZAR ENROLAMIENTO EN CURSO	Actor(es):	Estudiante
Código	CU-009	Pantalla(s)	P004
Ref. Req:	RF008		
Resumen / Descripción:	El sistema deberá permitir realizar el enrolamiento a un curso.		
Objetivos:	<ul style="list-style-type: none"> ● Visualizar los cursos disponibles en el sistema 		
Precondiciones:	<ul style="list-style-type: none"> ● El usuario (GENERAL) debe estar dentro del sistema. ● El sistema debe haber mostrado la página de inicio. 		

Postcondiciones:

- El sistema mostrará la acción realizada por el usuario.

CURSO NORMAL DE EVENTOS**Visualizar Cursos**

7. El usuario ingresa a la página Inicio.
8. El sistema muestra el menú.
9. El usuario selecciona la opción LEARNING - CURSOS DISPONIBLES.
10. El sistema muestra en la pantalla con la lista de cursos registrados.
11. El sistema muestra el listado de 10 cursos por pantalla y los muestra en la tabla de resultados.
12. El usuario selecciona la tarjeta del **Curso**.
13. El usuario revisa información referente al curso.
14. El usuario presiona el botón: Enrolar.
15. El sistema registra petición en el sistema.

CURSO ALTERNO DE EVENTOS**5. Ingresar al curso enrolado**

- 5.1. El usuario selecciona la tarjeta del curso.
- 5.2. El usuario presiona el botón: Ir al curso.
- 5.3. El sistema muestra el contenido del curso.

6.1.9 Modelado de la interacción**6.1.9.1 Estilos arquitectónicos estructurales**

- Orientado a Objetos
- Capas:
 - Patrón Diseño: Facade
 - Patrón Data Access Object (DAO)

6.1.9.2 Estilos arquitectónicos de despliegue

- Monolito → Servicios (Backend)
- Monolito → Vista (FrontEnd)

6.1.9.3 Diagramas de secuencia**Diagrama de Secuencia: Buscar Curso**

El diagrama de secuencia (ver Fig. 17) permite visualizar claramente la interacción entre diferentes objetos en un sistema en el caso de uso: Buscar Curso.

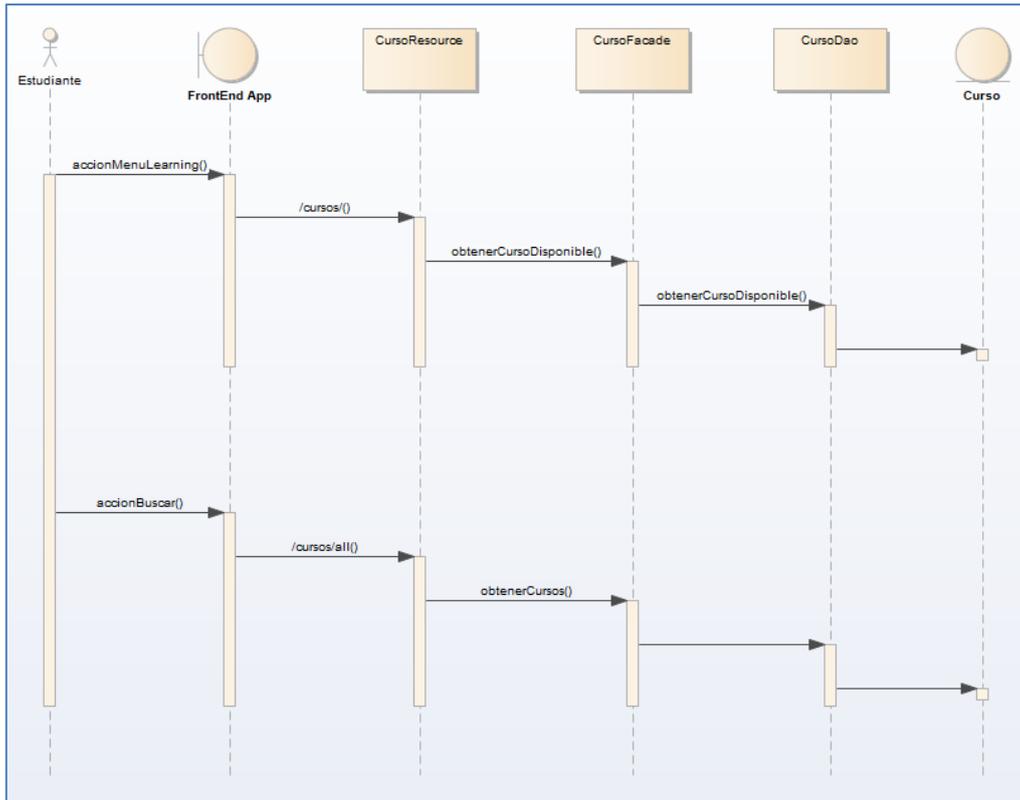


Fig. 17. Diagrama de secuencia sobre el caso de uso: Buscar curso.

Diagrama de Secuencia: Enrolar a un Curso

Mediante el diagrama de secuencia se modela la lógica de una operación, una función o un procedimiento complejo, en Fig. 18 se detalla el diagrama sobre el proceso de enrolar un estudiante a un curso.

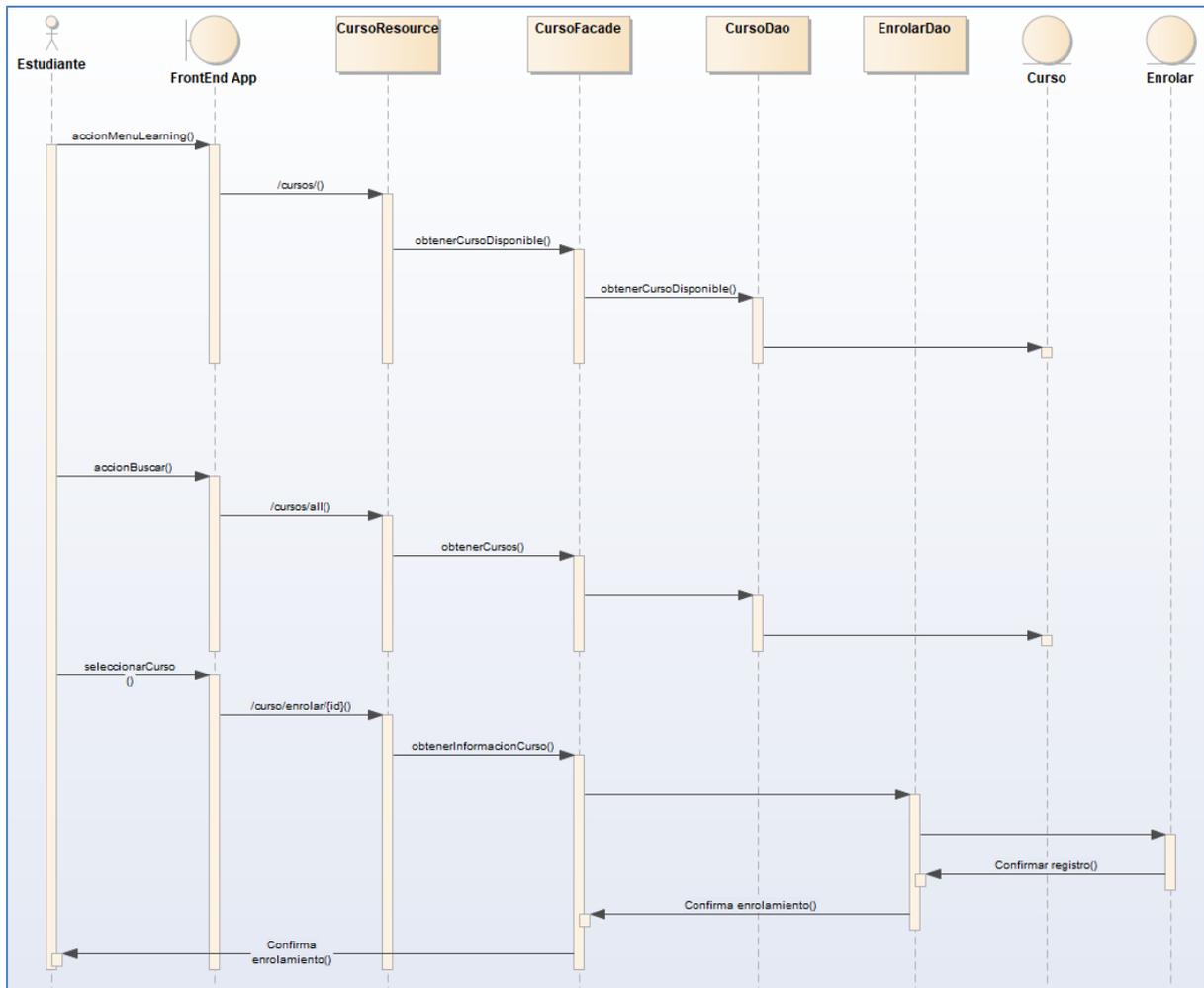


Fig. 18. Diagrama de secuencia sobre el caso de uso: Enrolar curso.

Diagrama de secuencia: Administrar curso

Para visualizar la interacción entre los diferentes objetos del sistema y la secuencia de eventos que ocurren al momento de administrar un curso en Fig. 19 se describe su proceso.

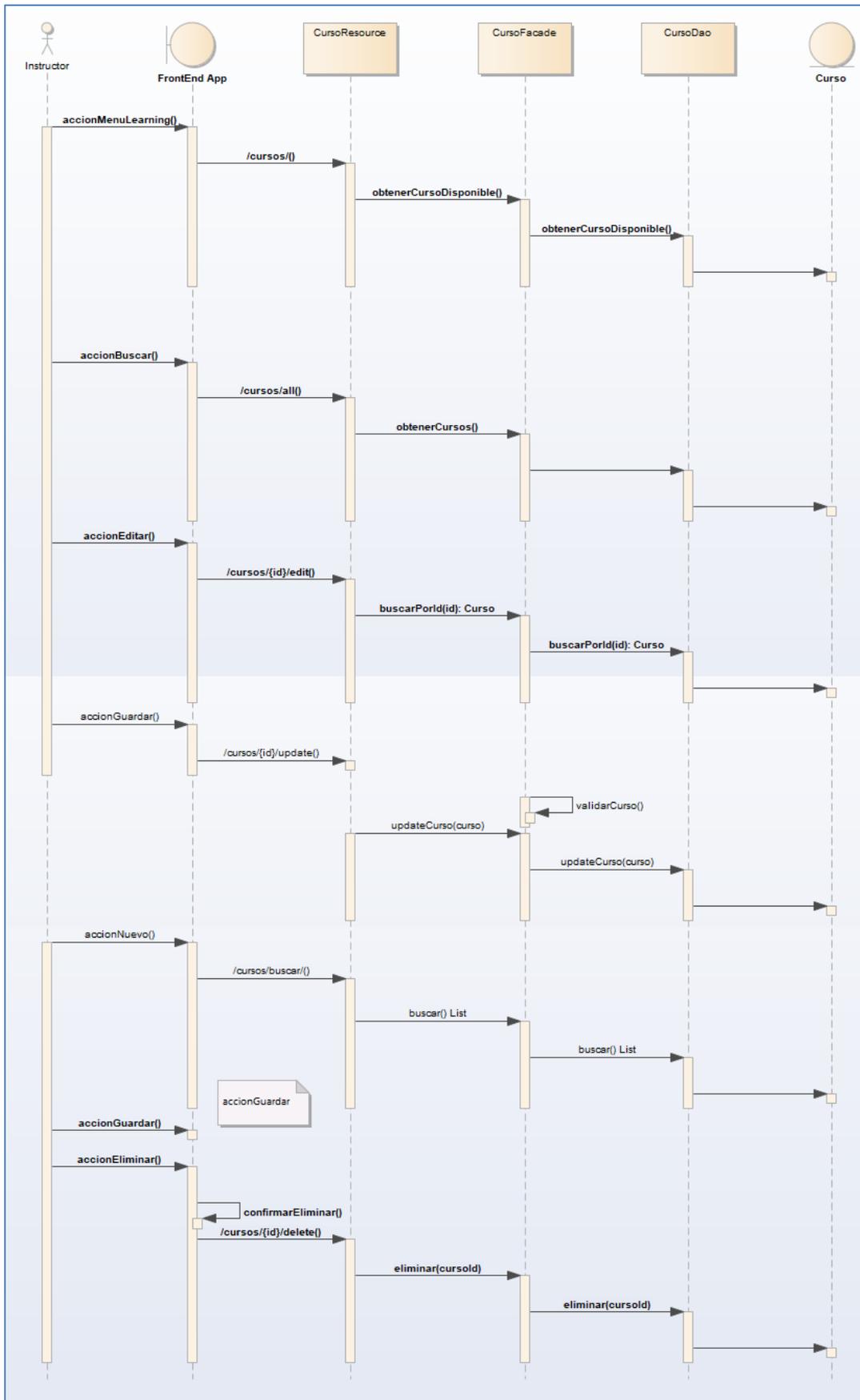


Fig. 19. Diagrama de secuencia sobre el caso de uso: Administrar curso.

6.1.9.4 Modelo estático final

Diagrama de paquetes

Este tipo de diagrama permitió organizar elementos del modelo en grupos lógicos que pueden representar paquetes de código fuente, bibliotecas de clases, componentes del sistema o cualquier otro conjunto de elementos relacionados (ver Fig. 20).

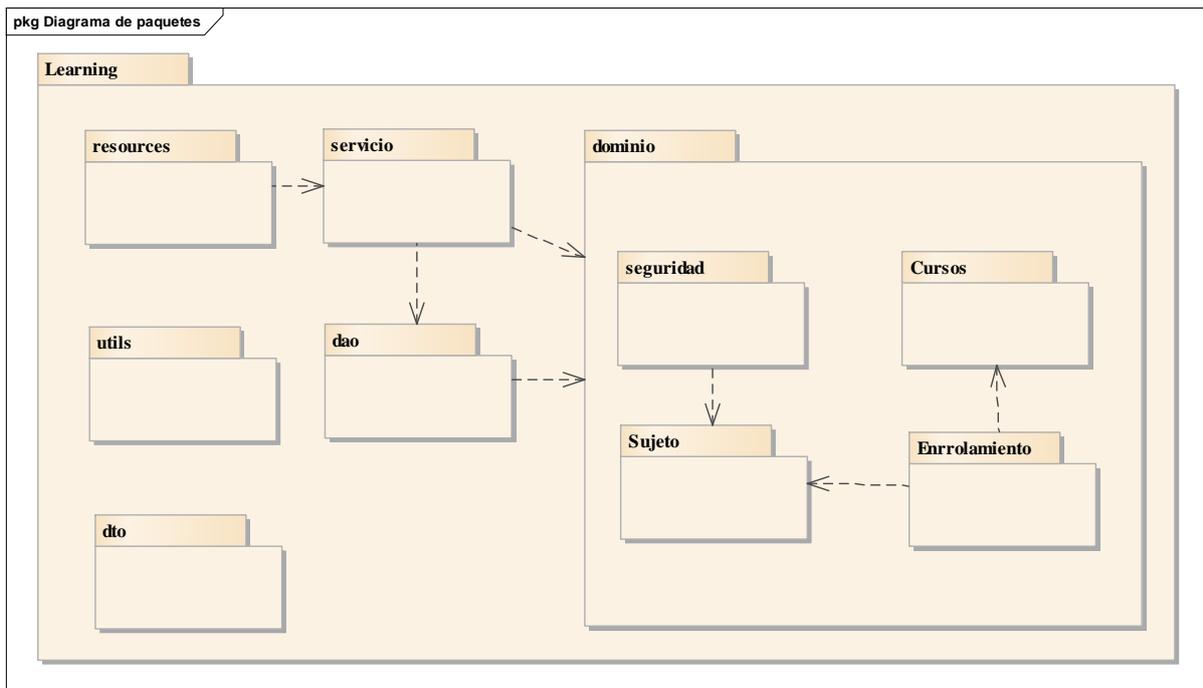


Fig. 20. Diagrama de paquetes del módulo Learning.

Diagrama de clases

Este tipo de diagrama permitió ilustrar las clases, interfaces y sus relaciones, propiedades y métodos. Las clases son representaciones abstractas de entidades del mundo real y se utilizan para representar objetos en el software (ver Fig. 21).

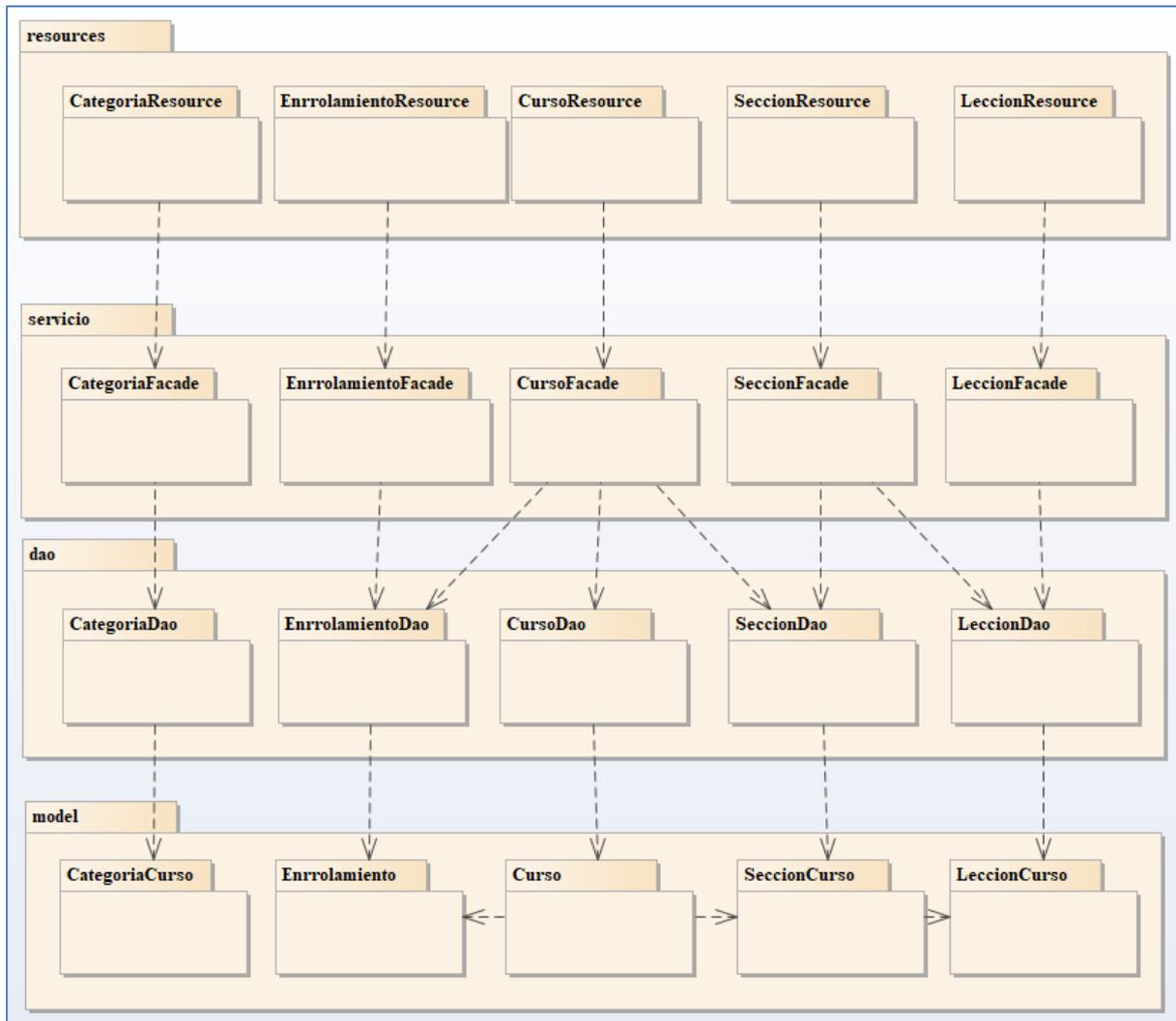


Fig. 21. Diagrama de clases del módulo de Learning.

6.1.9.5 Modelado de la implementación

Diagrama de componentes

Este tipo de diagrama permitió comprender cómo se relacionan los diferentes componentes del sistema y cómo se comunican entre sí. En Fig. 22 se detallan los componentes y su relación.

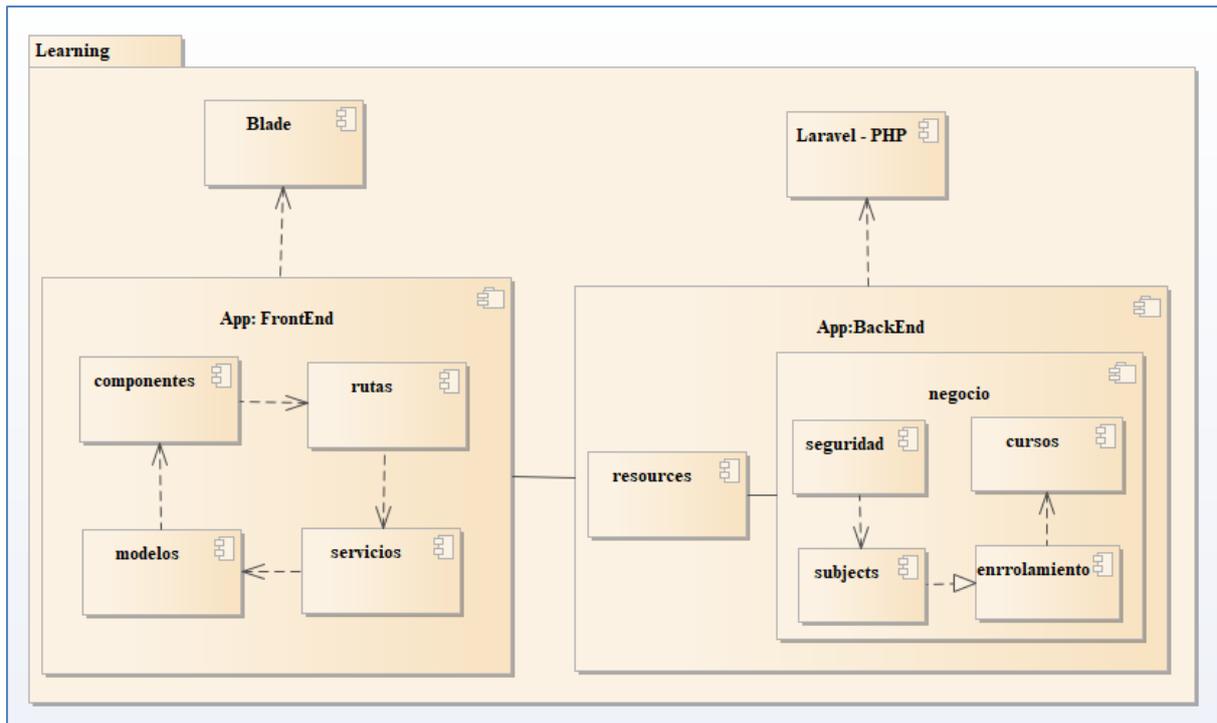


Fig. 22. Diagrama de componentes del módulo de Learning.

Diagrama de despliegue

Este tipo de diagrama permitió modelar la arquitectura física del sistema. Este diagrama representa cómo se despliegan los componentes en los nodos de hardware y cómo se comunican entre sí (ver Fig. 23).

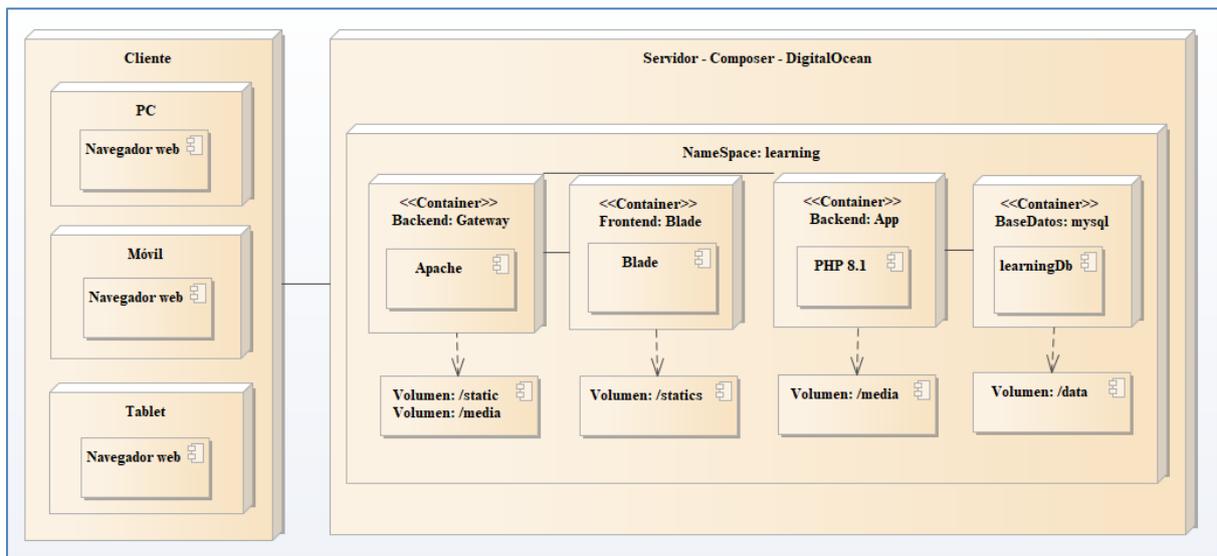


Fig. 23. Diagrama de despliegue del sistema.

6.2 Objetivo 2: Aplicar el patrón de diseño de software Modelo Vista Controlador, haciendo uso del Framework Laravel 9 para crear una solución robusta y eficiente que permita a futuro lograr mantenibilidad en la solución.

Para aplicar de forma correcta el patrón de diseño de software Modelo Vista Controlador se realizó un análisis detallado de los diferentes componentes e iteraciones que forman parte de este patrón dentro del Framework Laravel 9 (ver Fig. 24).

El patrón de diseño arquitectónico Modelo-Vista-Controlador (MVC) surge con el objetivo de mejorar y reducir el esfuerzo de programación dentro de la implementación de sistemas informáticos, a partir de estandarizar el diseño de las aplicaciones.

MVC surge como un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo (ver Fig. 25).

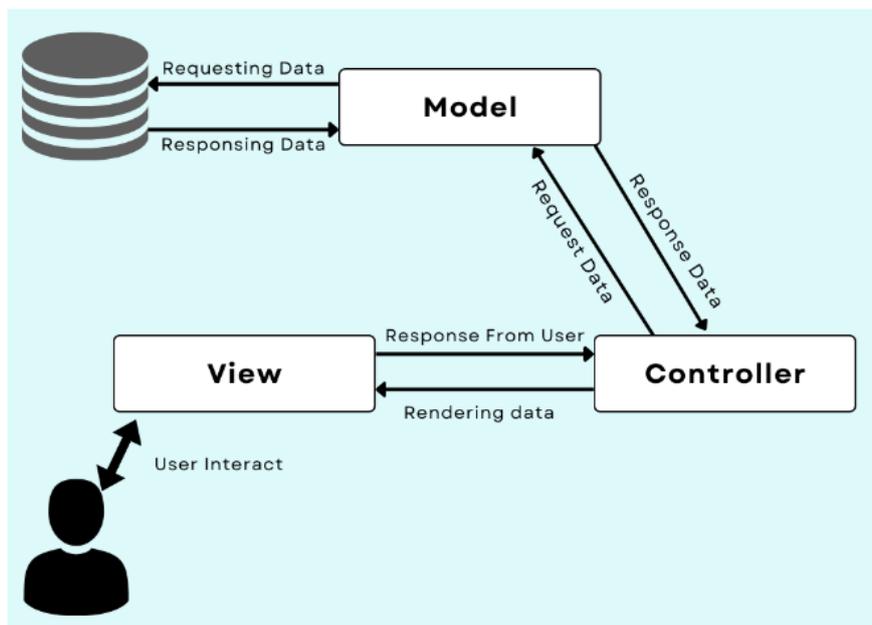


Fig. 24. Diseño arquitectónico: Modelo-Vista-Controlador.

6.2.1 El Modelo

Representa la lógica de negocio y los datos de la aplicación. Es el componente encargado de interactuar con la base de datos y manejar la lógica de la aplicación. El modelo puede contener clases, funciones y estructuras de datos necesarios para representar los objetos de la aplicación y realizar las operaciones CRUD (Create, Read, Update, Delete) en la base de datos.

6.2.2 La Vista

Es el componente encargado de mostrar la información al usuario. Se trata de la interfaz de usuario que el usuario final utiliza para interactuar con la aplicación. La vista puede ser una página web, una aplicación móvil, una interfaz gráfica de usuario (GUI), etc.

6.2.3 El Controlador

Es el componente que actúa como intermediario entre el modelo y la vista. Recibe las solicitudes del usuario a través de la vista, procesa la información necesaria utilizando el modelo y actualiza la vista con la información relevante. El controlador es el encargado de coordinar la interacción entre el modelo y la vista.

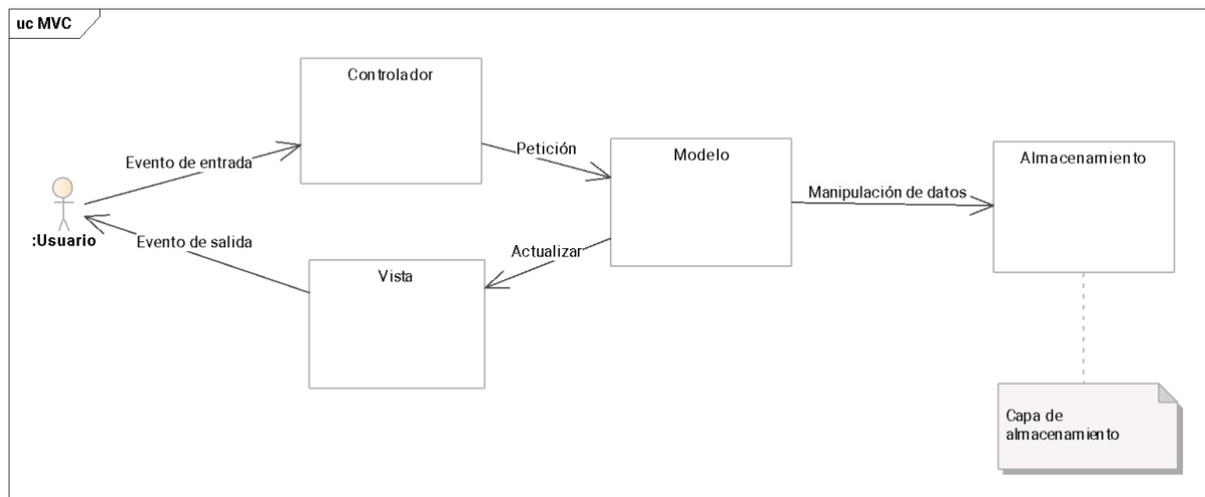


Fig. 25. Estructura de funcionamiento del Modelo-Vista-Controlador.

6.2.4 Responsabilidades

Modelo:

Los modelos están asociados a los recursos de la aplicación y trabajan a menudo con registros de bases de datos. Son clases que representan entidades en la aplicación, por ejemplo, usuarios, artículos de noticias o eventos.

Laravel incluye Eloquent, que es un eficiente mapeador objeto-relacional (ORM) y el desarrollador puede definir entidades, asignarlas a tablas de base de datos relacionadas y usar métodos PHP con ellas en lugar de sentencias SQL sin procesar. Podemos realizar con el ORM

consultas eficientes a la base de datos, sin escribir ninguna instrucción SQL. Eloquent opera como una capa-modelo en las aplicaciones [17].

El Modelo es responsable de realizar operaciones de Crear, Leer, Actualizar y Borrar (CRUD) en fuentes de datos; y cualquier otro procesamiento relacionado con el negocio - basado en las peticiones de la Vista/Controlador - y producir los resultados de vuelta a la Vista/Controlador [18]. En el Anexo 3 se detalla el modelo físico de la base de datos que se usa como insumo dentro del Modelo.

Controlador:

La responsabilidad principal del controlador es actuar como intermediario entre la vista y el modelo. El controlador recibe las solicitudes del usuario desde la vista y coordina la interacción con el modelo para obtener los datos necesarios. Luego, el controlador actualiza la vista con la información relevante.

El controlador es responsable de procesar la entrada del usuario y tomar decisiones sobre qué acciones deben ser tomadas en respuesta a esa entrada. Por ejemplo, si el usuario hace clic en un botón, el controlador es el encargado de interpretar ese clic y decidir qué acción se debe llevar a cabo en respuesta a él.

Además, el controlador también es responsable de aplicar la lógica de control de la aplicación. Esto incluye la gestión de errores, la validación de datos y la realización de cálculos y transformaciones de datos adicionales.

El Controlador es un módulo activo, que media entre la Vista y el Modelo pasivos, realizando el manejo de eventos [18].

Vista:

La responsabilidad principal de la vista es mostrar los datos al usuario y permitirle interactuar con ellos. La vista es la parte de la aplicación que se encarga de la presentación visual de la información y la interfaz de usuario.

La vista debe ser independiente del modelo y del controlador, lo que significa que no debe tener conocimiento de la forma en que los datos son manejados por el modelo ni de cómo son

procesados por el controlador. En lugar de eso, la vista simplemente solicita los datos que necesita al controlador y los muestra al usuario de la manera más adecuada.

Además, la vista también debe ser capaz de capturar la entrada del usuario y enviarla al controlador para que sea procesada. Esto incluye la captura de eventos clics, pulsaciones de teclas, selecciones de menú y cualquier otra acción que el usuario pueda realizar.

La vista también puede aplicar cierta lógica de presentación, como el formateo de datos y la manipulación de la interfaz de usuario en respuesta a ciertas acciones del usuario. Por ejemplo, una vista puede mostrar un gráfico que cambia en tiempo real en respuesta a los datos proporcionados por el controlador, como se puede ver en Fig. 26.

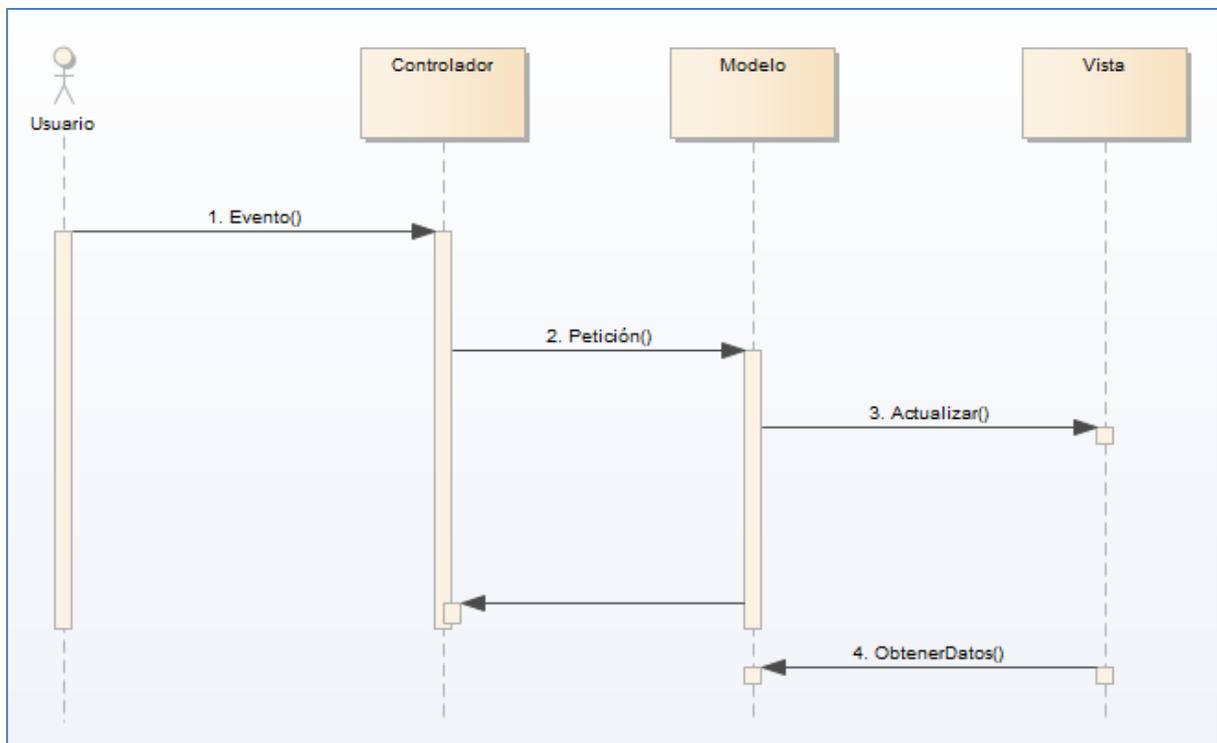


Fig. 26. Diagrama de secuencia del diseño arquitectónico: MVC.

Pasos:

1. El usuario genera un evento.
2. El Controlador recibe el evento y lo traduce en una petición hacia el Modelo.
3. El modelo de ser necesario puede llamar a la vista para su actualización.
4. Para cumplir con la actualización la Vista puede solicitar datos al Modelo.
5. El Controlador recibe el control.

6.2.5 Arquitectura de una aplicación en Laravel

Laravel es uno de los frameworks más populares de código abierto para el desarrollo de aplicaciones web usando PHP. Fue creado por Taylor Otwell en 2011 y ha ganado mucha popularidad desde entonces debido a su simplicidad, elegancia y facilidad de uso.

Laravel se basa en el patrón de arquitectura MVC (Modelo Vista Controlador) y utiliza una variedad de herramientas y bibliotecas de terceros para proporcionar una experiencia de desarrollo fluida y eficiente.

Características más destacadas:

- Sistema de enrutamiento
- Migraciones de base de datos que permiten a los desarrolladores definir la estructura de la base de datos y realizar cambios de forma programática
- Integración con el motor de plantillas Blade para crear vistas
- Sistema de autenticación de usuarios completo y flexible
- Capacidades avanzadas de pruebas unitarias y de integración
- Cuenta con una amplia comunidad de desarrolladores y una documentación detallada, lo que lo convierte en una opción para la ejecución de proyectos de desarrollo web de todos los tamaños y niveles de complejidad.

La arquitectura de Laravel (ver Fig. 27) es un flujo de comunicación entre el Foundation del framework, los Services Providers, una estructura de Controllers con Middlewares y una capa de servicios que se comunica con el acceso a datos del ORM y al final, con la base de datos.

En el Anexo 1 se adjunta el link donde podemos acceder al repositorio oficial del GitHub, así mismo, en el Anexo 2 se adjuntó el link donde podemos acceder a la documentación oficial de Laravel convirtiéndose en una fuente confiable.

Elementos de la arquitectura básica de una app en Laravel

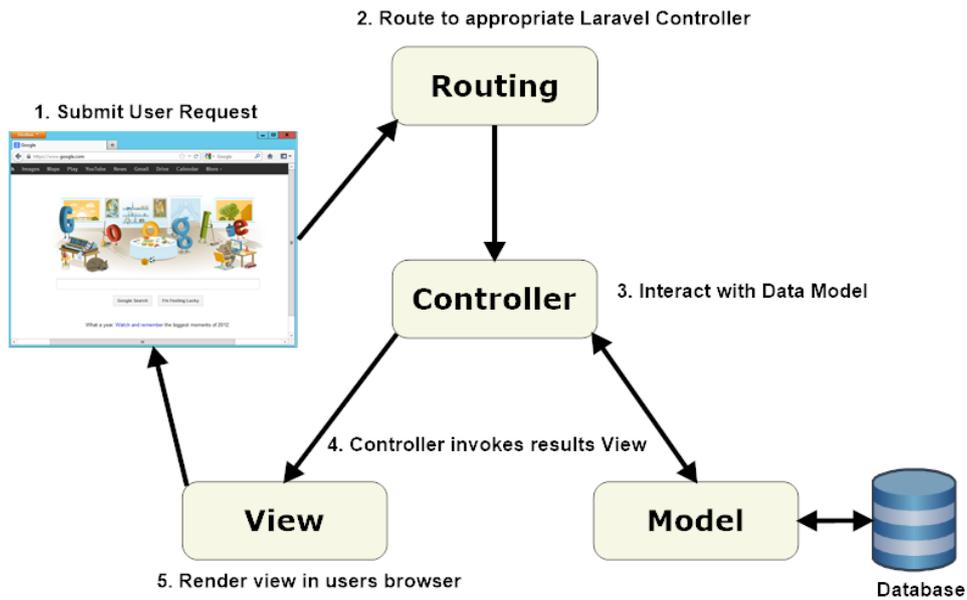


Fig. 27. Elementos básicos que componen un proyecto en Laravel.

Interacción entre los elementos principales en Laravel

Los elementos principales en Laravel interactúan entre sí para crear una estructura sólida y coherente para el desarrollo de aplicaciones web como se puede apreciar en Fig. 28.

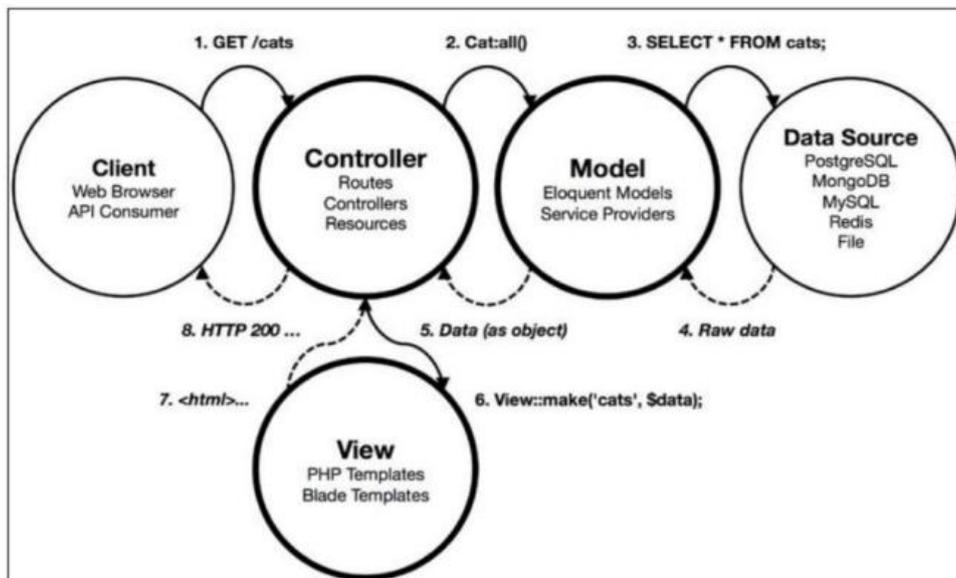


Fig. 28. Iteración de los elementos de un proyecto en Laravel.

6.2.6 Detalle de resultados

Luego de realizar el análisis de los diferentes componentes que forman parte el Sistema de gestión Académica tenemos como resultado los siguientes componentes creados dentro del framework Laravel:

Modelo

Inventario de componentes dentro del modelo (ver Fig. 29).

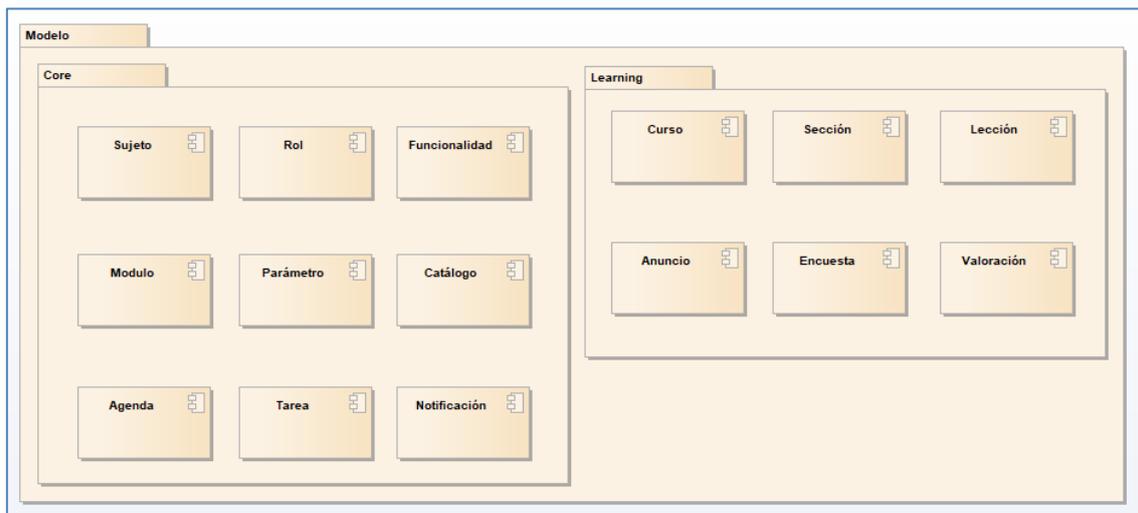


Fig. 29. Diagrama que detallan todos los modelos creados en el proyecto.

Vista

Inventario de componentes dentro de la vista (ver Fig. 30).

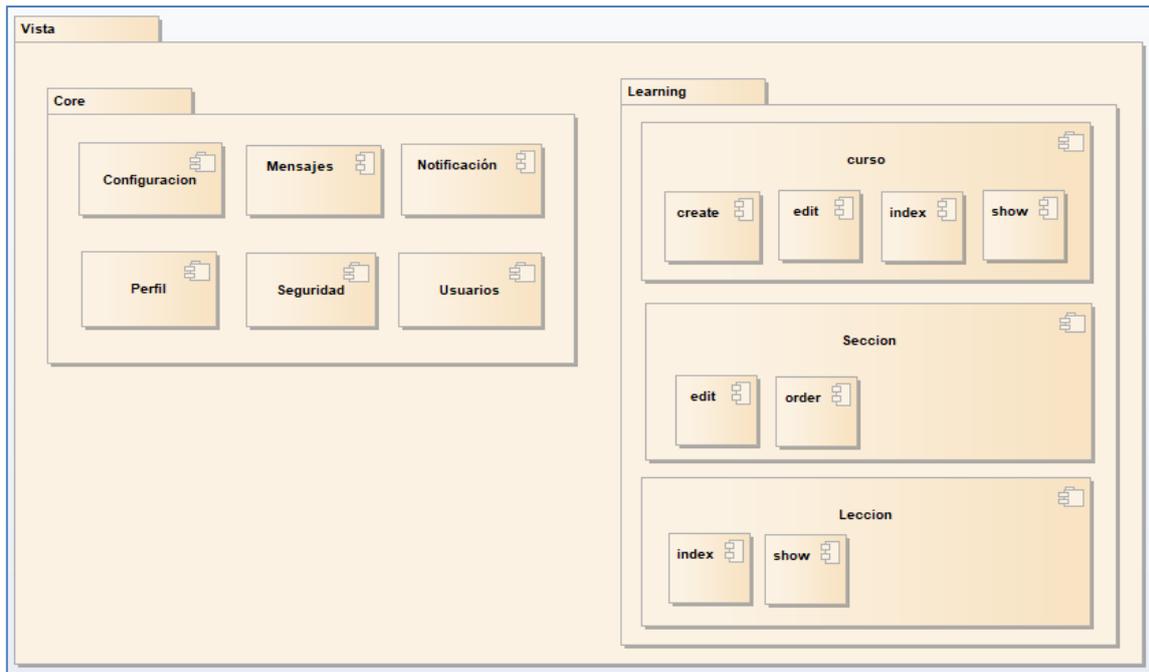


Fig. 30. Diagrama que detallan todas las vistas creadas en el proyecto.

Controlador

Inventario de componentes dentro de la vista (ver Fig. 31).

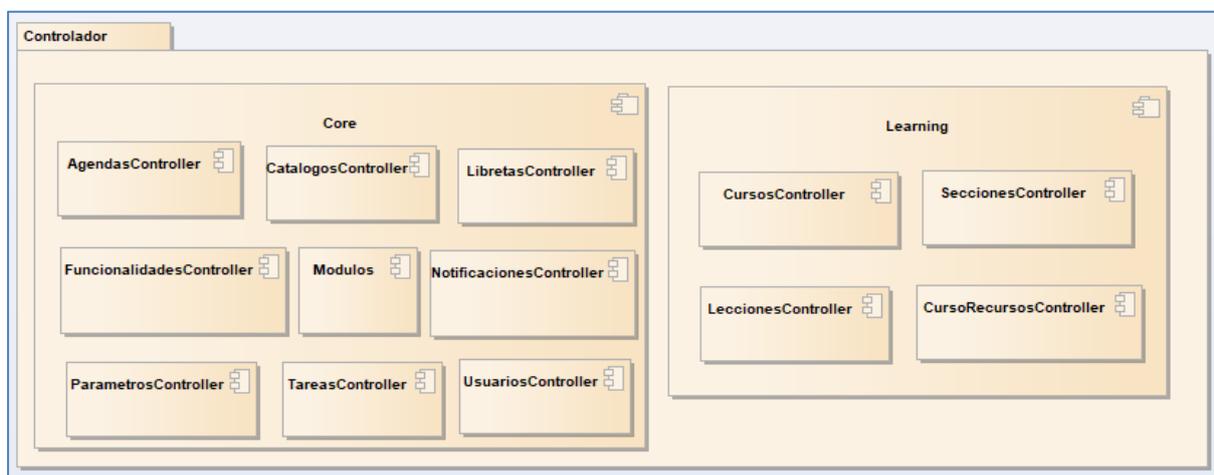


Fig. 31. Diagrama que detallan todos los controladores creados en el proyecto.

6.3 Objetivo 3: Validar y verificar el funcionamiento de la solución de software mediante pruebas de software en un ambiente Cloud.

En base al proyecto se definió el siguiente plan de pruebas

6.3.1 Contexto de las pruebas a ejecutar

6.3.1.1 Proyecto / Subprocesos de prueba

El trabajo de titulación: “Desarrollo de un Sistema de Gestión de Aprendizaje como herramienta que permita organizar, centralizar contenidos académicos y facilitar la exposición de recursos formativos” consta de los siguientes módulos:

- Funcionalidades
- Roles
- Cursos
- Enrolamiento
- Usuarios
- Configuración
- Seguridad

Módulos que se relacionan con las funcionalidades del sistema (ver Fig. 32):

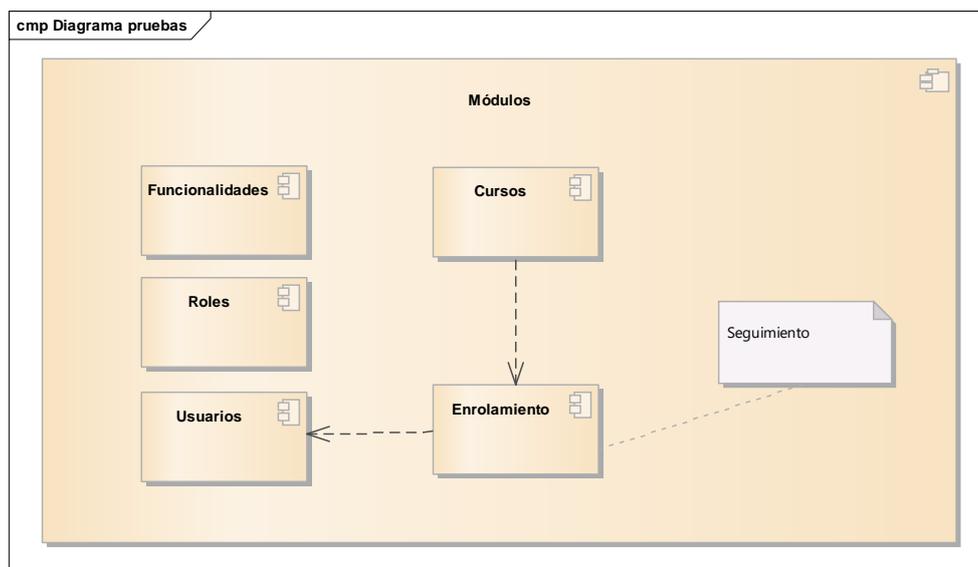


Fig. 32. Diagrama de pruebas.

6.3.1.2 Elementos de prueba

Se realizarán pruebas a los siguientes elementos:

- Registro de usuarios
- Gestión de Roles
- Gestión de funcionalidades
- Gestión de Usuario - Roles
- Módulo de administración de Cursos
- Enrolamiento

6.3.1.3 Alcance de la Prueba

El sistema compuesto por los módulos mencionados en el acápite 6.3.1.2

Los factores de la calidad no funcionales como el rendimiento, la seguridad informática y la usabilidad no se probarán en este proyecto de prueba, estas pruebas se externalizarán a otra empresa.

La empresa de outsourcing responsable de esas pruebas creará un Plan de Pruebas por separado para estos tipos de pruebas.

6.3.1.4 Suposiciones y Restricciones

Suposiciones:

- El ambiente de pruebas será un clon del ambiente de producción de forma que puedan administrar los cursos y gestionar enrolamientos de diversos usuarios.

Restricciones:

- Las reuniones o talleres que requieran la participación completa del equipo se realizarán antes de las 3.00pm debido a la geo-localización remota de parte del equipo. (1 miembro del equipo tiene otro uso horario).
- El hardware no cumple con los requerimientos y requisitos preestablecidos para el plan de pruebas.
- El equipo de arquitectura debe aprobar la propuesta arquitectónica, lo que puede conllevar a retrasos.

- El equipo de análisis de sistemas, no ratifica los planes de pruebas detallados, informes, etc., por no estar bajo los estándares de calidad de software.

6.3.1.5 Partes Interesadas

- Instructores: Realización de las pruebas.
- Estudiantes: Realización de las pruebas.
- Usuarios administrativos: Realización de las pruebas.
- Equipo de Sistemas: Revisión y aprobación del Plan de Pruebas y del Cronograma de pruebas.

6.3.2 Comunicación de las Pruebas

A continuación, se identifican (ver Tabla XIII) los principales puntos de comunicación:

TABLA XIII
COMUNICACIÓN DE LAS PRUEBAS

Punto de Comunicación	Propósito	Frecuencia	Medios	Responsable	Audiencia
Reunión de inicio	Inicio	Una vez	Reunión	Líder de Proyecto/ QA Manager	Equipo
Reuniones internas	Estado	Semanal	Reunión	Líder de Proyecto/ QA Manager	Equipo interno
Reportes de Estado	Estado	Semanal	Documento vía email	Líder de Proyecto/ QA Manager	Equipo
Reporte de Hitos	Alcance	Como sea apropiado	Documento vía email	Líder de Proyecto/ QA Manager	Equipo

6.3.3 Registro de Riesgos

En la siguiente tabla (ver Tabla XIV) se identifican los riesgos del proyecto, así como se determina la severidad de cada uno de los riesgos multiplicando el impacto por la probabilidad de ocurrencia.

El impacto y la probabilidad se determinan teniendo en cuenta una escala de 1 al 5, donde 5 es el más alto.

TABLA XIV
REGISTRO DE RIESGOS

No	Riesgos	Probabilidad (1-5)	Impacto (1-5)	Severidad (Prob*Impct)	Plan de Mitigación
1	Retrasos en la implementación de las funcionalidades.	2	5	10	Evaluar el avance del desarrollo de las funcionalidades y re-planificar acorde al avance de ser necesario.
2	Retraso en la entrega de insumos.	3	4	12	Generar reuniones periódicas para realizar el seguimiento de tareas.
3	Ambiente de pruebas inapropiado.	3	5	15	Realizar con anticipación el ambiente de pruebas adecuado.
4	Formación inadecuada del equipo de pruebas.	4	5	20	Capacitar al personal, previo a la realización de las pruebas.
5	Rotación de personal.	3	4	12	Brindar las condiciones necesarias.

6.3.4 Estrategia de pruebas

Se planificarán las pruebas específicas por cada uno de los subprocesos de prueba, los mismos que detallarán los casos de pruebas a aplicar, las herramientas, técnicas de prueba, criterios de aceptación y recursos a usar.

6.3.4.1 Subprocesos de prueba

Las pruebas para el Sistema de “Desarrollo de un Sistema de Gestión de Aprendizaje como herramienta que permita organizar, centralizar contenidos académicos y facilitar la exposición de recursos formativos” incluirán los siguientes subprocesos de prueba:

- Pruebas de componentes.
- Pruebas de integración.
- Pruebas de sistema.
- Pruebas de aceptación.
- Pruebas de usabilidad.
- Pruebas de estrés.
- Pruebas de seguridad

6.3.4.2 Entregables de pruebas

Para cada subproceso de pruebas se debe generar la siguiente documentación:

- Plan de pruebas del subproceso de prueba;
- Especificación de Casos de Pruebas
- Informes de Estado de las Pruebas.
- Informe de Finalización del subproceso de prueba.

6.3.4.3 Técnicas de diseño de pruebas

Se identifican las técnicas que se utilizarán para el diseño de las pruebas.

- Pruebas de Casos de Uso
- Pruebas Exploratorias

6.3.4.4 Criterio de finalización y pruebas

Las pruebas deben alcanzar una cobertura de requisitos del 80% y todos los procedimientos de pruebas deben ejecutarse sin fallas de gravedad 1 (alta).

6.3.4.5 Métricas

Las siguientes métricas se recogerán durante el transcurso de la ejecución de las pruebas:

- Número de casos de prueba ejecutados.
- Número de incidentes por categoría.
- Número de casos de prueba re-ejecutados.
- Número de incidentes resueltos por categoría.

Métricas internas

En Tabla XV se detallan las métricas internas.

TABLA XV
DETALLE DE METRICAS INTERNAS

Elemento	Métrica	Propone medir
Funcionalidad	Cubrimiento de funcionalidades	Cantidad de casos de prueba / Total de funcionalidades a probar
	Tiempo por funcionalidad	Total, de tiempo en ejecutar las pruebas / Total de funcionalidades probadas
	Densidad de defectos por funcionalidad	Total, de No conformidades (NC) encontrados / Total de funcionalidades probadas
Proceso de Ejecución PE	Esfuerzo de las PE	Contar el tiempo dedicado a las PE en el ciclo de pruebas
	N C encontradas en las PE	Contar las NC encontradas durante las PE
	Funcionalidades exploradas	Contar las funcionalidades exploradas durante las PE del ciclo de prueba
Esfuerzo	Esfuerzo planificado (5)	Sumar el tiempo planificado para todas las actividades del proyecto de pruebas

	Esfuerzo realizado en el ciclo de pruebas	Sumar el tiempo invertido en todas las actividades del ciclo de prueba
	Esfuerzo realizado hasta el momento (3)	Sumar el tiempo realizado desde que el proyecto comenzó hasta el último ciclo realizado
	Esfuerzo por realizar (4)	Sumar el tiempo que se planifica para cada actividad hasta que el proyecto de prueba termine
	Esfuerzo total (6)	Sumar (3) y (4)
	Desviación en el esfuerzo planificado	Restar (6) menos (5)
	Esfuerzo por persona	Contar el esfuerzo por persona del equipo de prueba

Métricas externas

En Tabla XVI se detallan las métricas externas.

TABLA XVI
DETALLE DE METRICAS EXTERNAS

Característica de funcionalidad		
Elemento	Métrica	Propone medir
Idoneidad	Estabilidad en las especificaciones funcionales	¿Cuán estable son las especificaciones funcionales después de entrar a funcionar?
Funcionalidad	Cumplimiento de estándar de interfaz	¿Cómo se cumplen las interfaces con los reglamentos aplicables, las normas y regulaciones?

La confiabilidad es una de las características clave que se evalúan en las pruebas de un sistema. La confiabilidad se refiere a la capacidad del sistema para realizar una tarea sin fallos durante un período de tiempo determinado, en Tabla XVII se detallan algunas de ellas.

TABLA XVII
CARACTERÍSTICAS DE CONFIABILIDAD

Característica de confiabilidad		
Elemento	Métrica	Propone medir
Madurez	Grado de solución ante fallos totales	¿Cuántas condiciones de fallo total están resueltas?
	Erradicación de fallos contra casos de prueba	¿Cuántos fallos han sido corregidos?
	Densidad de fallos totales contra casos de prueba	¿Cuántos fallos totales fueron detectados durante un periodo de pruebas definido?
	Madurez de las pruebas	¿Está bien probado el producto?

La usabilidad se refiere a la facilidad de uso y la satisfacción del usuario al utilizar un sistema. Las pruebas de usabilidad son una parte importante del proceso de desarrollo de software para garantizar que el sistema sea fácil de usar y satisfaga las necesidades del usuario final (ver Tabla XVIII).

TABLA XVIII
CARACTERÍSTICAS DE USABILIDAD

Característica de usabilidad		
Elemento	Métrica	Propone medir
Comprensibilidad	Accesibilidad a demo	¿A qué proporción de demos/tutoriales pueden acceder los usuarios?
Adoctrinamiento	Eficiencia de la documentación y/o sistemas de ayuda en el uso	¿Qué proporción de las funciones pueden ser utilizadas correctamente después de leer la documentación o el uso de sistemas de ayuda?

6.3.4.6 Requisitos del entorno de pruebas

Ambiente de pruebas

- **Navegadores:** Chrome, Mozilla Firefox.
- **Sistemas Operativos:** Windows, Linux, Android, IOS.

Herramientas de prueba

- **JUnit testing framework:** Ejecución y Reporte de las pruebas automatizadas.
- **JMeter:** Ejecución de pruebas de estrés.
- **JIRA:** Seguimiento de los casos de pruebas e incidencias.
- **Postman:** Prueba de los servicios web.

6.3.4.7 Re-testing y regresión de las Pruebas

Se deben realizar las pruebas de confirmación (re-testing) y regresiones necesarias para cumplir con los criterios de finalización.

Se estima que se realizarán no más de 3 ciclos de pruebas, el último ciclo incluirá una prueba de regresión completa.

6.3.4.8 Criterios de Suspensión y Reanudación

Criterios de suspensión

- Análisis, propuestas tecnológicas y de arquitectura no validadas por el departamento de sistemas.
- La solución no cumple con las funcionalidades especificadas en el documento de Especificación de Requisitos del Proyecto.
- Una de las características principales es que contenga un error que impida probar áreas críticas del sistema.
- El entorno de pruebas no sea estable y/o no retorne resultados confiables.
- La no disponibilidad de ambientes de prueba de sistemas o módulos de terceros.
- No contar con el hardware necesario.

Criterio de reanudación

Llegar a un acuerdo entre las partes interesadas para reanudar las pruebas o que se hayan solucionado los defectos/problemas encontrados.

6.3.4.9 Desviaciones de la Estrategia de prueba Organizacional

Este punto no se va a desarrollar porque no contamos con Estrategia de Prueba Organizacional.

6.3.5 Actividades y Estimados de Prueba

Las pruebas se dividirán en las siguientes actividades principales:

- Definición de una estructura general de las pruebas por conjuntos de casos de uso a probar.
- Especificación detallada de los casos de pruebas.
- Establecimiento del entorno de pruebas.
- Primer ciclo de ejecución de las pruebas.
- Segundo ciclo de ejecución de las pruebas (re test y regresión del primer ciclo).
- Tercer ciclo ejecución de las pruebas (re test y regresión del segundo ciclo y pruebas a elementos pendientes del primer ciclo).
- Informe de reporte de estado de las pruebas semanales.
- Informe de finalización de las pruebas.

Las actividades detalladas de las pruebas y sus estimaciones se pueden encontrar en la herramienta de gestión de pruebas del proyecto.

6.3.6 Personal

6.3.6.1 Roles, Actividades y Responsabilidades

La matriz que se detalla a continuación (ver Tabla XIX) ilustra qué rol está involucrado en qué actividad (es) y cuál es el nivel de participación.

TABLA XIX
DETALLE DE ROLES, ACTIVIDADES Y RESPONSABILIDADES

Rol/Actividad	Líder de QA	Analista QA	Ing. de Automatización	Audiencia	TI
Definición de una estructura general de las pruebas por conjuntos de casos de uso a probar.	R	C	I	I	A
Especificación detallada de los casos de pruebas.	A	R	C	I	A
Establecimiento del entorno de pruebas.	A	R	R	I	R
Primer ciclo de ejecución de las pruebas.	I	R	R	C	I
Segundo ciclo de ejecución de las pruebas (re test y regresión del primer ciclo).	I	R	C	C	I
Tercer ciclo ejecución de las pruebas (re test y regresión del segundo ciclo y pruebas a elementos pendientes del primer ciclo).	I	R	R	C	I
Informe de reporte de estado de las pruebas semanales.	I	R	C	C	I
Informe de finalización de las pruebas.	R	R	R	I	A

Notación: Responsable, Aprobar, Consultado, Informado.

6.3.7 Necesidades de Entrenamiento

Se necesita una introducción al sistema para los que van a ejecutar las pruebas. Se estima que esta actividad puede tomar 1 hora.

6.3.8 Cronograma

El cronograma general de las pruebas se muestra a continuación en Tabla XX.

TABLA XX
CRONOGRAMA GENERAL DE PRUEBAS

Actividades de pruebas	SEMANA					
	1	2	3	4	5	6
Análisis	X					
Diseño		X				
Entorno de pruebas		X				
Ejecución (3 ciclos)			X	X	X	
Finalización						X

6.3.9 Informe de pruebas de usabilidad

En el Anexo 4 se detalla el informe de pruebas de usabilidad aplicado al proyecto de titulación.

7 Discusión

7.1 Impacto de la educación virtual

Según la definición de Llumitasig Galarza, Mónica: “El estudiante pasa a ser el centro del proceso de aprendizaje, es el que construye el conocimiento, a través del apoyo y guía del profesor. En este contexto son de gran importancia el uso de entornos y metodologías facilitadoras del aprendizaje que permitan al estudiante aprender y convertir las informaciones en conocimientos. Las TIC, son elementos adecuados para la creación de estos entornos por parte de los profesores, apoyando el aprendizaje constructivo, colaborativo y por descubrimiento” [19] , queda demostrado que la implementación de la educación virtual en línea ha tenido un impacto significativo a nivel general. La educación en línea ha permitido a empresas de tecnología expandir su alcance y ofrecer servicios en todo el mundo. Con la creciente demanda de capacitación en tecnología, muchas empresas han visto la oportunidad de proporcionar cursos en línea a través de plataformas de aprendizaje SGA.

Según Cabero J: “en la actualidad ya contamos con un cúmulo de tecnologías sostenibles y razonables que pueden permitirnos realizar diferentes actividades que afiancen nuestras posibilidades comunicativas y expresivas, industriales, culturales, y lúdicas, como hace poco tiempo no nos imaginábamos que pudiera ocurrir. Así, por ejemplo, nos permiten: comunicarnos de forma sincrónica y asincrónica, y de manera fiable y rápida y con costos notablemente más reducidos que en las décadas anteriores” [20], la educación virtual ha mejorado la eficiencia en la formación de los empleados. En lugar de dedicar tiempo y recursos a la capacitación presencial, las empresas pueden ofrecer cursos en línea a sus empleados, lo que les permite estudiar a su propio ritmo y en su propio horario. Además, este tipo de educación es más rentable para todos, ya que no tienen que pagar por el alquiler de laboratorios de capacitación y otros gastos asociados con la capacitación tradicional.

Otro impacto importante de la educación virtual es la capacidad de las empresas para mantenerse actualizadas con los avances tecnológicos. Las plataformas de gestión de aprendizaje permiten la facilidad de crear nuevos contenidos o actualizarlos de acuerdo a su evolución lo que permite a los empleados aprender nuevas tecnologías o mantenerse al día con las últimas tendencias u actualizaciones en la industria del software.

7.2 Implementación de un Sistema de Gestión de Aprendizaje como herramienta que permita organizar, centralizar contenidos académicos y facilitar la exposición de recursos formativos

En base a la definición de Egas Castro P: “Los LMS cumplen con una función de llevar a los educandos a niveles superiores del pensamiento humano, donde no solo se tiene al estudiante como ente memorista y que puede llegar a comprender información, sino llegar al análisis, evaluación y creación del pensamiento” [21], el presente trabajo de titulación se coincide con la importancia de implementar este tipo de sistemas ya que facilita la gestión del aprendizaje, mejora la eficiencia, facilita el acceso al aprendizaje, permite la personalización del aprendizaje y proporciona análisis de datos.

Dentro de la implementación de este proyecto se construyó un sistema de gestión de aprendizaje que permite organizar, centralizar contenidos académicos y facilitar la exposición de recursos formativos con la finalidad de mejorar la experiencia educativa de los estudiantes y facilitar la tarea de los instructores. El sistema fue diseñado para ser fácil de usar y accesible desde cualquier lugar con una conexión a Internet. En la presente discusión, se abordarán los principales hallazgos y las implicaciones prácticas y teóricas.

En el trabajo de investigación: De la educación tradicional a la educación mediada por TIC se destaca: “Debido a la revolución tecnológica y el constante uso de la tecnología, la educación presencial ya no será estrictamente presencial, las llamadas clases presenciales irán perdiendo su nombre ya que poco a poco se van incorporando en las misma, estrategias orientadas al uso de la tecnología como parte del proceso de enseñanza-aprendizaje y como exigencia inminente en los estudiantes debido a sus ventajas” [22], esto se pudo comprobar ya que la gestión de aprendizaje fue muy bien recibido tanto por los estudiantes como por los potenciales instructores que lo utilizaron. Los estudiantes apreciaron la facilidad de acceso a los materiales de estudio, así como la capacidad de interactuar con sus compañeros. Los instructores destacaron la facilidad de administración de las actividades y contenidos académicos junto a la capacidad de evaluar y retroalimentar a los estudiantes en tiempo real.

En segundo lugar, se observó que el sistema de gestión de aprendizaje permitió una mayor flexibilidad y adaptabilidad en el proceso de enseñanza y aprendizaje. Los estudiantes pudieron acceder a los materiales de estudio en cualquier momento y lugar, lo que les permitió adaptar el ritmo de estudio a sus necesidades y preferencias. Los instructores pudieron diseñar y

administrar actividades académicas más variadas y personalizadas, lo que aumentó la motivación y el compromiso de los estudiantes.

En tercer lugar, se encontró que el sistema de gestión de aprendizaje permitió una mayor colaboración y participación de los estudiantes. Los estudiantes pudieron interactuar con sus compañeros y docentes a través de herramientas de comunicación en línea, lo que les permitió compartir ideas y conocimientos de manera más efectiva. Los instructores pudieron fomentar el trabajo en equipo y la colaboración entre los estudiantes, lo que mejoró el aprendizaje cooperativo.

En cuanto a las implicaciones prácticas, se encontró que el sistema de gestión de aprendizaje puede ser una herramienta útil para mejorar la experiencia educativa de los estudiantes y la eficiencia de los instructores. Además, puede ser una herramienta valiosa para facilitar la implementación de metodologías de enseñanza innovadoras y personalizadas. En cuanto a las implicaciones teóricas, se encontró que el sistema de gestión de aprendizaje puede ser un recurso valioso para la investigación educativa y el desarrollo de teorías y modelos de enseñanza y aprendizaje.

7.3 Uso de la metodología de desarrollo ágil ICONIX

En este proyecto se ha utilizado la metodología ICONIX para el desarrollo del sistema. Se ha podido constatar que el uso de esta metodología ha permitido una mayor eficiencia en el proceso de desarrollo, así como una mayor calidad del software final.

Según IMA: “Iconix deriva directamente del RUP y su fundamento es el hecho de que un 80% de los casos pueden ser resueltos tan solo con un uso del 20% del UML, con lo cual se simplifica muchísimo el proceso sin perder documentación al dejar solo aquello que es necesario.” [23], lo que confirma que una de las principales ventajas de ICONIX es su enfoque orientado a objetos, lo que permite una mayor modularidad y reutilización del código. Además, la metodología permite una mayor interacción y colaboración entre el equipo de desarrollo y el cliente, lo que resulta en un mayor entendimiento y satisfacción de los requerimientos del cliente.

El enfoque incremental y evolutivo es otra de las ventajas de ICONIX, lo que permite un desarrollo más ágil y una mayor adaptación a los cambios y requerimientos del cliente.

Además, el uso de modelos y diagramas UML ayuda a una mejor visualización y entendimiento del software en desarrollo.

Sin embargo, es importante destacar que el uso de ICONIX requiere una mayor planificación y documentación, lo que puede aumentar la carga de trabajo del equipo de desarrollo. Además, es necesario contar con un equipo de desarrollo altamente capacitado en la metodología y en el desarrollo orientado a objetos.

Iconix se guía a través de casos de uso y sigue un ciclo de vida iterativo e incremental. El objetivo es que a partir de los casos de uso se obtenga el sistema final [23].

7.4 Uso del diseño arquitectónico Modelo-Vista-Controlador

En este proyecto se ha utilizado el patrón arquitectónico Modelo-Vista-Controlador (MVC) para el diseño del sistema. Se ha podido constatar que el uso de este patrón ha permitido una mayor eficiencia en el proceso de desarrollo, así como una mayor calidad y mantenibilidad del software final.

Una de las principales ventajas del patrón MVC es su enfoque de separación de preocupaciones, lo que permite una mayor modularidad y reutilización del código. Además, el patrón permite una mayor independencia entre la interfaz de usuario y la lógica de negocio, lo que resulta en una mayor facilidad de mantenimiento y adaptación a cambios en la interfaz o en la lógica de negocio.

Según Vallejo Bermeo J: “El tiempo de desarrollo de una aplicación que implementa el patrón de diseño MVC es mayor, al menos en la primera etapa, que el tiempo de desarrollo de una aplicación que no lo implementa, ya que MVC requiere que el programador implemente una mayor cantidad de clases que en un entorno de desarrollo común no son necesarias” [24], esta afirmación no es válida ya que durante la ejecución de este proyecto se concluyó que el uso de MVC permite que cada componente se desarrolle y gestione de forma independiente, lo que facilita la mantenibilidad, escalabilidad y la reutilización del código, esto ayuda a reducir la complejidad del código y a mejorar la claridad y comprensión del mismo, lo que conduce a una mayor eficiencia en el desarrollo y una mayor calidad del software en general.

El enfoque en la escalabilidad y extensibilidad del sistema de software es otra ventaja importante del patrón MVC. Al separar la lógica de negocio de la interfaz de usuario, se pueden añadir nuevas funcionalidades al sistema sin afectar la interfaz existente. Además, el uso de

modelos y controladores facilita la realización de pruebas unitarias y la identificación de errores en el sistema.

Por lo tanto, el uso del patrón arquitectónico Modelo-Vista-Controlador ha demostrado ser una opción efectiva para el diseño del sistema, al permitir mayor modularidad, independencia entre la interfaz y la lógica de negocio, escalabilidad y extensibilidad del sistema.

8 Conclusiones

- La implementación de un Sistema de Gestión Académica es una inversión clave para cualquier empresa o institución educativa que desee mejorar la eficiencia y la calidad de sus procesos académicos. Un SGA reduce la carga de trabajo de los administradores e instructores, permitiéndoles centrarse en las actividades académicas.
- La creación de un SGA puede requerir un esfuerzo y recursos significativos, pero los beneficios son igualmente importantes. El SGA permite la integración de tecnologías de la información y comunicación en el proceso educativo, mejorando la motivación y el compromiso de los estudiantes.
- La seguridad y protección de datos son fundamentales en el desarrollo de un SGA, ya que maneja información personal y académica confidencial. Además, la formación de los usuarios y el soporte técnico son clave para asegurar una adopción efectiva y uso continuo del SGA. La recopilación y análisis de datos a través del SGA pueden proporcionar información valiosa para la toma de decisiones educativas o empresariales y la mejora continua de la calidad del proceso de enseñanza-aprendizaje.
- Uno de los principales beneficios de un SGA es la gestión automatizada de los registros académicos. Permite la creación y gestión eficiente de registros académicos precisos y actualizados, lo que a su vez mejora la planificación académica, la gestión de recursos y la toma de decisiones.
- La implementación de un SGA permite a los estudiantes acceder a recursos educativos en línea en cualquier momento y lugar. Además, la plataforma de aprendizaje en línea permite una mayor flexibilidad y personalización del proceso educativo para adaptarse a las necesidades individuales de los estudiantes.
- La comunicación entre estudiantes, instructores y personal administrativo es un aspecto fundamental para el éxito del proceso educativo. El SGA facilita la comunicación a través de herramientas síncronas y asíncronas, lo que mejora la interacción y la retroalimentación entre los miembros de la comunidad educativa.
- La evaluación del desempeño académico de los estudiantes es un aspecto crítico para garantizar la calidad del proceso educativo. El SGA permite la creación y gestión eficiente de sistemas de evaluación, lo que facilita la recolección y el análisis de datos y la generación de informes precisos y detallados sobre el desempeño académico de los estudiantes. Esto a su vez facilita la toma de decisiones y la identificación de oportunidades de mejora en el proceso educativo.

- La implementación de un SGA debe ir acompañada de una capacitación constante tanto para instructores como para estudiantes y personal administrativo, para asegurar un uso adecuado y eficiente de la herramienta.

9 Recomendaciones

- Para el desarrollo de un sistema de gestión académica eficiente, es importante contar con un equipo multidisciplinario que pueda garantizar una visión integral del proceso educativo. Este equipo debe estar compuesto por expertos en tecnología, pedagogía, administración y otros campos relacionados, que puedan aportar diferentes perspectivas y conocimientos para una solución efectiva.
- Es recomendable definir claramente los objetivos y metas del sistema de gestión académica, así como su alcance y funcionalidades. Esto permitirá tener una guía clara para el desarrollo y asegurar que la herramienta cumpla con las necesidades y requerimientos específicos de la institución educativa.
- Se sugiere que el sistema de gestión académica tenga una interfaz de usuario amigable e intuitiva, que permita un fácil acceso a la información y una navegación sencilla. Esto reducirá la curva de aprendizaje y facilitará su uso por parte de todos los usuarios.
- Es importante que el sistema de gestión académica tenga una alta capacidad de personalización y configuración, para que pueda adaptarse a las necesidades específicas de la institución educativa y garantizar su sostenibilidad en el tiempo.
- Se recomienda que el sistema de gestión académica tenga una sección dedicada a la evaluación y retroalimentación constante, tanto por parte de docentes como de estudiantes. Esto permitirá identificar áreas de mejora y asegurar un proceso de mejora continua.
- Es sugerible que el sistema de gestión académica cuente con una sección dedicada a la gestión de horarios y asignación de tareas, que permita una planificación eficiente de las actividades educativas y una optimización de los recursos.
- Se recomienda que el sistema de gestión académica cuente con herramientas de colaboración en línea y plataformas de videoconferencia, que permitan la interacción y el trabajo en equipo entre docentes y estudiantes.
- Es importante que el sistema de gestión académica tenga una sección dedicada a la seguridad y protección de datos, para garantizar la privacidad y confidencialidad de la información de los usuarios.
- Se sugiere que el sistema de gestión académica tenga una sección dedicada a la generación de informes y estadísticas, que permita una toma de decisiones basada en datos y una evaluación objetiva del proceso educativo.

- Es recomendable que el sistema de gestión académica tenga una estrategia clara de implementación y capacitación constante para todos los usuarios, que permita un uso adecuado y efectivo de la herramienta y asegure su sostenibilidad en el tiempo.

10 Bibliografía

- [1] M. de la C. Rodríguez Andino and H. M. Barragán Sánchez, “Entornos virtuales de aprendizaje como apoyo a la enseñanza presencial para potenciar el proceso educativo,” *Killkana Social*, vol. 1, no. 2, 2017, doi: 10.26871/killkana_social.v1i2.29.
- [2] B. Gilbert, “Online Learning Revealing the Benefits and Challenges,” *Fisher Digital Publications Education*, 2015.
- [3] V. Arkorful and N. Abaidoo, “The role of e-learning, advantages and disadvantages of its adoption in higher education,” *International Journal of Instructional Technology and Distance Learning*, vol. 12, no. 1, 2015.
- [4] Hemsley, “Jones international university’s focus on quality e-learning opens doors for students worldwide,” *Business Media*, pp. 26–29, 2002.
- [5] E. Mikołajewska and D. Mikołajewski, “E-learning in the education of people with disabilities,” *Advances in Clinical and Experimental Medicine*, vol. 20, no. 1, 2011.
- [6] K. Barker, “Dilemmas at a distance,” *Assess Eval High Educ*, vol. 11, no. 3, 1986, doi: 10.1080/0260293860110306.
- [7] B. Gros Salvat, “La evolución del e-learning: del aula virtual a la red,” *RIED. Revista Iberoamericana de Educación a Distancia*, vol. 21, no. 2, 2018, doi: 10.5944/ried.21.2.20577.
- [8] V. Hoffman, “Que es un sistema de gestión de aprendizaje (LMS)?,” 2017.
- [9] B. P. Alejo and A. F. Aparicio, “La planificación de estrategias de enseñanza en un entorno virtual de aprendizaje,” *Revista Científica UISRAEL*, vol. 8, no. 1, 2021, doi: 10.35290/rcui.v8n1.2021.341.
- [10] N. Vega, R. Flores-Jiménez, I. Flores-Jiménez, B. Hurtado-Vega, and J. S. Rodríguez-Martínez, “Teorías del aprendizaje,” *XIKUA Boletín Científico de la Escuela Superior de Tlahuelilpan*, vol. 7, no. 14, 2019, doi: 10.29057/xikua.v7i14.4359.
- [11] J. R. Jiménez Cruz, “Modelo de diseño instruccional semipresencial basado en proyectos a partir de un LMS y PLEs - Integrando ambientes organizacionales y personales. (Spanish),” *Blended Instructional Design Model Based on Projects from an LMS and PLEs - Integrating Organizational and Personal Environments. (English)*, no. 42, pp. 1–24, 2014, [Online]. Available: <https://libproxy.library.unt.edu:9443/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=fua&AN=98490508&scope=site>

- [12] P. Mendoza López, “Lineamientos de diseño de información para el desarrollo de sitios educativos en internet,” *Mendoza López, P.*, 2006, Accessed: Mar. 22, 2023. [Online]. Available: <https://es.scribd.com/document/14428907/Diseno-de-espacios-en-internet-educativos#>
- [13] Francisco Mora Vicarioli, “Objetos de aprendizaje: Importancia de su uso en la Educación Virtual,” May 2012.
- [14] V. Alfaro, “Un modelo de gestión de requerimientos para minimizar el porcentaje de incumplimiento,” *Cienc Desarro*, vol. 22, no. 1, 2019, doi: 10.21503/cyd.v22i1.1737.
- [15] A. Carranza, “Modelo en cascada: ¿Qué es y cómo funciona?,” *Crehana*, 2021.
- [16] C. A. López S., “Cómo mantener el patrón modelo vista controlador en una aplicación orientada a la WEB,” *INVENTUM*, vol. 4, no. 7, 2009, doi: 10.26620/uniminuto.inventum.4.7.2009.72-78.
- [17] Z. Subecz, “Web-development with Laravel framework,” *Gradus*, vol. 8, no. 1, 2021, doi: 10.47833/2021.1.csc.006.
- [18] N. R. Dissanayake and G. K. A. Dias, “Balanced Abstract Web-MVC Style: An Abstract MVC Implementation for Web-based Applications,” *GSTF Journal on Computing (JOC)*, vol. Volume 5, no. 1, 2017.
- [19] M. Llumitasig Galarza, “Implementación de una Plataforma Virtual Moodle para la gestión del curso Tecnologías de la Información y Comunicación aplicadas a la educación para el departamento de Ciencias Exactas de la Escuela Politécnica del Ejército Extensión Latacunga,” Ambato, 2012.
- [20] J. Cabero Almenara, “TICS en educación,” *Universidad de Sevilla*, 2002.
- [21] P. Egas Castro and Pontificia Universidad Católica del Ecuador, “Desarrollo de una Plataforma LMS para la Gestión del Aprendizaje en Instituciones,” Ambato, 2018.
- [22] E. Yong Castillo and D. Hernando Bedoya Ortiz, “De la educación tradicional a la educación mediada por TIC: Los procesos de enseñanza aprendizaje en el siglo XXI.”
- [23] IMA, “Manual Introductorio de Iconix: 1-¿Qué es Iconix?,” *UDG*, 2019.
- [24] J. Vallejo Bermeo, “Estudio comparativo de los patrones para interfaces de usuario MVVM y MVC,” Riobamba.

11. Anexos

Anexo 1. GitHub Framework Laravel.

El repositorio para realizar la descarga del Framework Laravel:

<https://github.com/laravel/laravel.git>

Anexo 2. Sitio web con la documentación oficial del Framework Laravel.

El sitio web que expone la documentación oficial del Framework Laravel:

<https://laravel.com/docs/master/installation>

Anexo 3. Diagrama físico de base de datos

Diagrama de base de datos, Parte 1: Sitio, Parámetros, Catálogo, Notificaciones y Agenda.

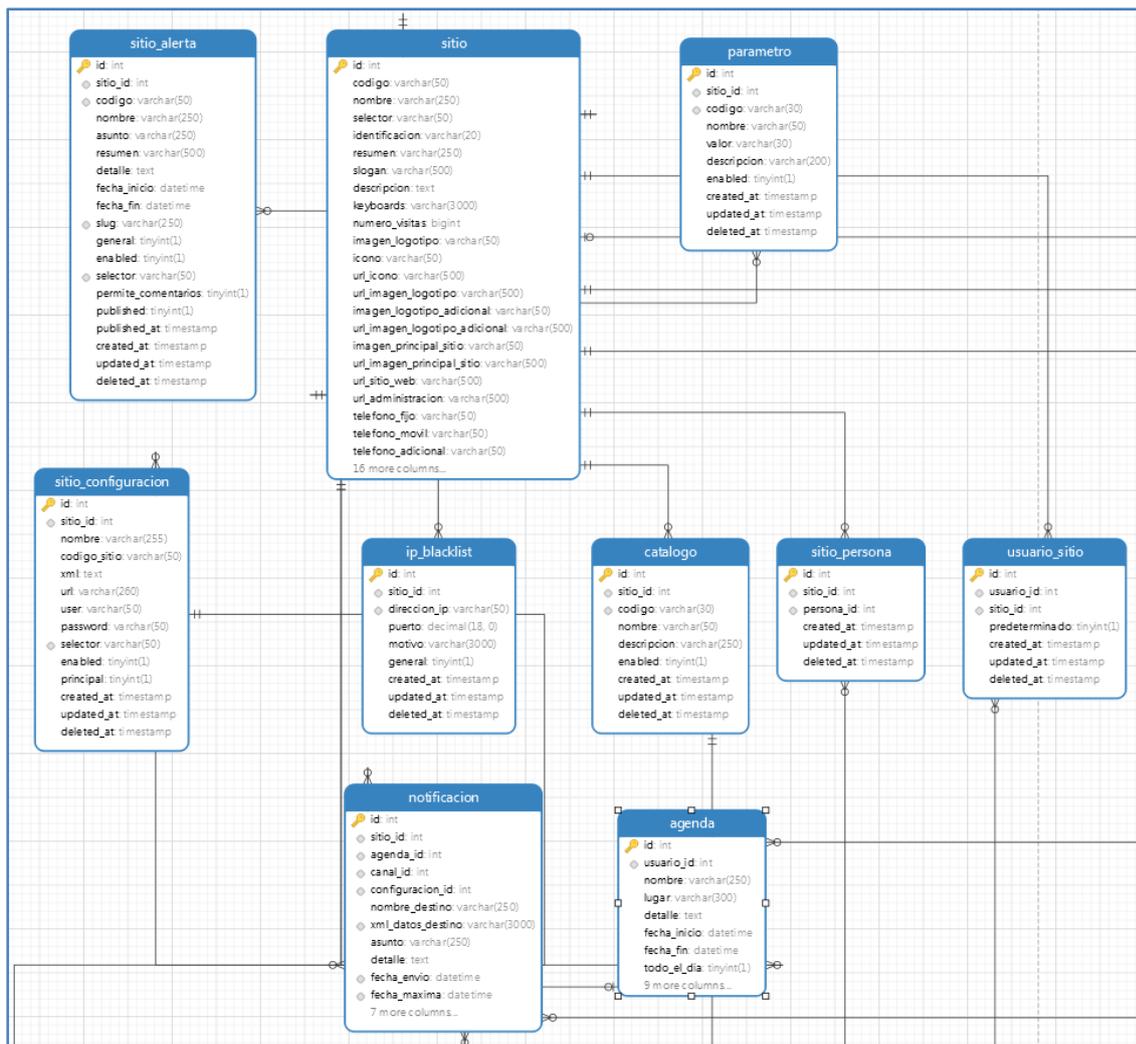
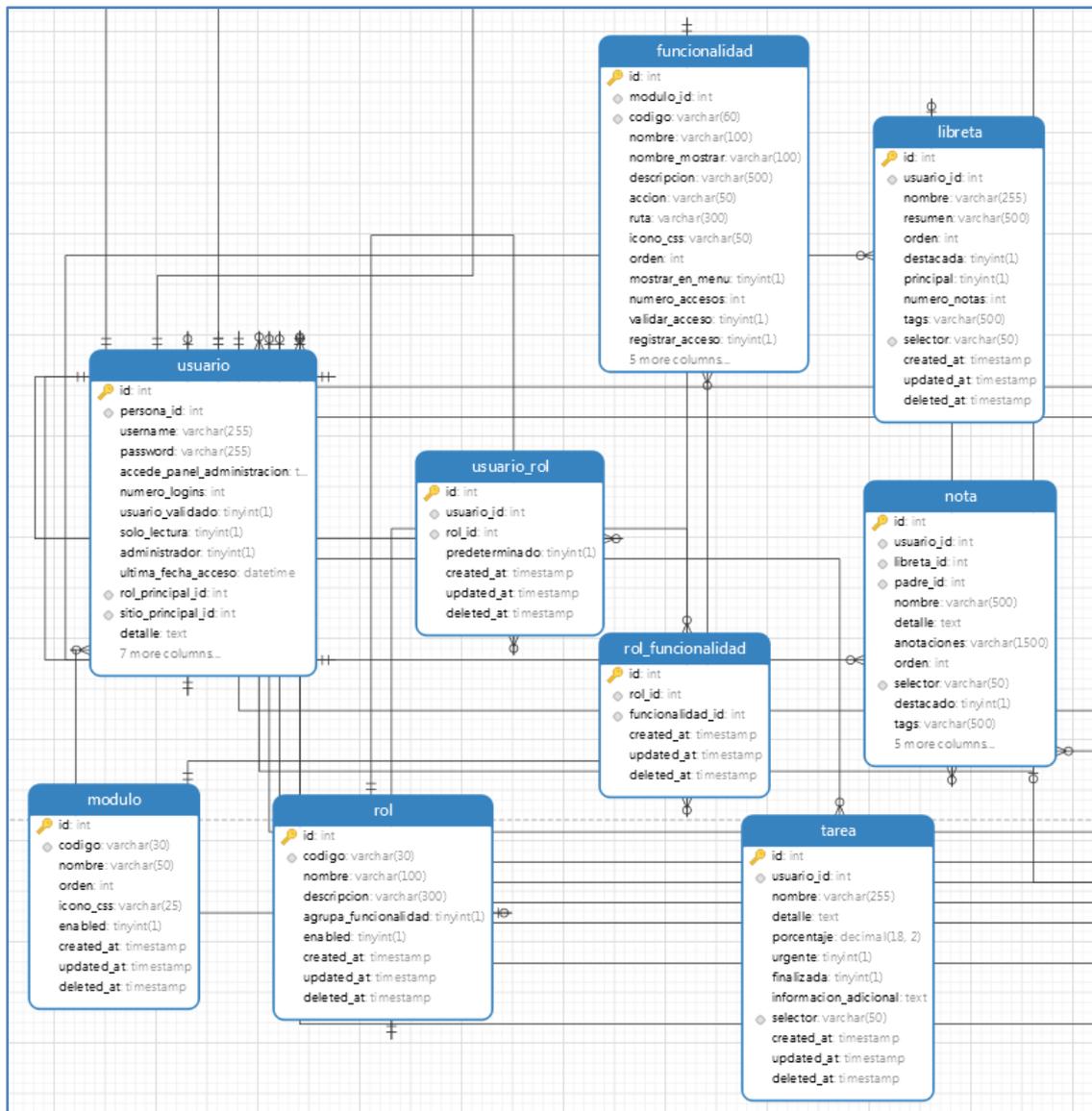


Diagrama de base de datos, Parte 3: Usuario, Tarea, Libreta, Nota, Funcionalidad, Rol, Módulo.



Anexo 4. Informe de pruebas de usabilidad

 **UNL** Universidad Nacional de Loja

FEIRNNR
Maestría en Ingeniería en Software

< Desarrollo de un Sistema de Gestión de Aprendizaje como herramienta que permita organizar, centralizar contenidos académicos y facilitar la exposición de recursos formativos >

Informe de Pruebas de USABILIDAD [Release Final]

Elaborado por:
Ramirez Coronel Edison Enrique

Versión 1.0

8 de abril de 2023

Educamos para Transformar



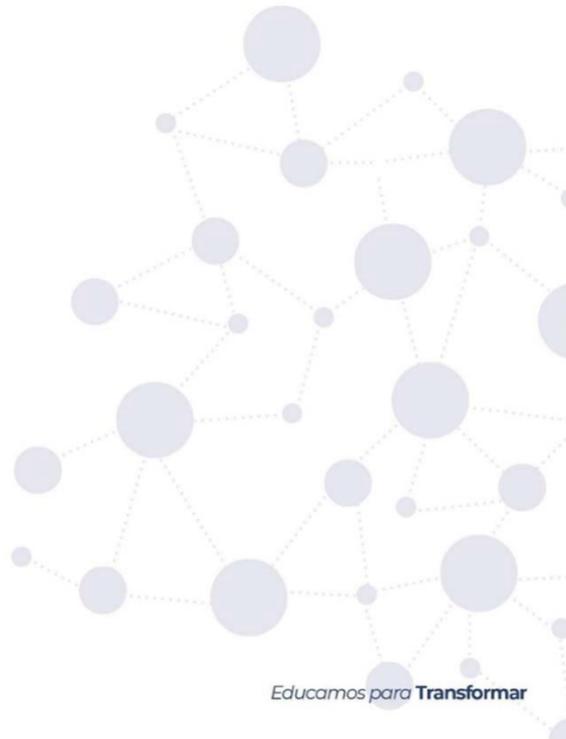
UNL

Universidad
Nacional
de Loja

FEIRNNR
Maestría en Ingeniería en
Software

Histórico de Revisiones

Fecha	Versión	Descripción	Autor
08/04/2023	Versión 1.0	Informe de pruebas	ERA





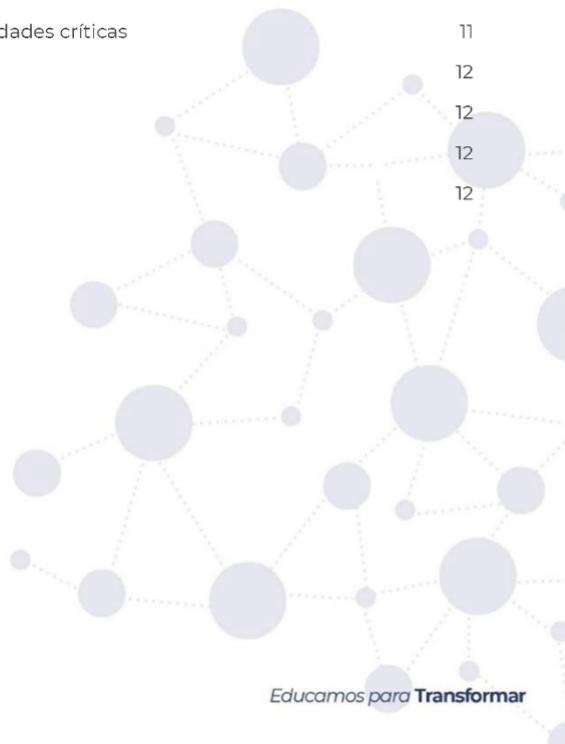
UNL

Universidad
Nacional
de Loja

FEIRNNR
Maestría en Ingeniería en
Software

CONTENIDOS

INTRODUCCIÓN	4
PLANIFICACIÓN Y DISEÑO	4
Cobertura de las pruebas	4
Tipos de prueba planificados y ejecutados	4
Secciones planificadas para evaluación	6
Casos de prueba planificados	6
Escenarios de negocio planificados y ejecutados	7
RESULTADOS	8
Ambiente de producción.	8
Bugs detectados	9
Total de Bugs reportados por evaluador y criticidad	10
Bugs diferidos	11
Pendientes	11
Tiempos de respuesta funcionalidades críticas	11
Aportes adicionales	12
Pruebas en producción	12
Pruebas de seguridad	12
Recomendaciones	12



Educamos para Transformar



INTRODUCCIÓN

El presente documento contiene los resultados de la ejecución de las pruebas de usabilidad, el mismo que se organiza en tres secciones: INTRODUCCIÓN, PLANIFICACIÓN Y DISEÑO orientado a la planificación de las pruebas y RESULTADOS orientado a la ejecución de las pruebas. La ejecución de las pruebas se realizó en un entorno de producción del Sistema de Gestión de Aprendizaje con la finalidad de hacer una validación post producción de los servicios ofertados en las secciones seleccionadas para el estudio.

PLANIFICACIÓN Y DISEÑO

1. Cobertura de las pruebas

Para la realización de las pruebas se tomó en cuenta el Sistema de Gestión Académica, para la cual se parte del frontal “<https://learning.vallex.com/login>” y las secciones: Login, Learning, Gestión de Seguridad, Administración de Usuarios. El plan de pruebas planificado en esta sección se enfoca a la usabilidad de la versión que se encuentra actualmente en producción (release final), para lo cual se detalla en el cuadro 1:

NÚMERO	ÁREAS	COBERTUR A
1	Inteligibilidad	Si
2	Aprendizaje	Si
3	Operabilidad	Si
4	Protección frente a errores	Si
5	Estética	Si
6	Accesibilidad	Si

Cuadro 1: Cobertura de las pruebas

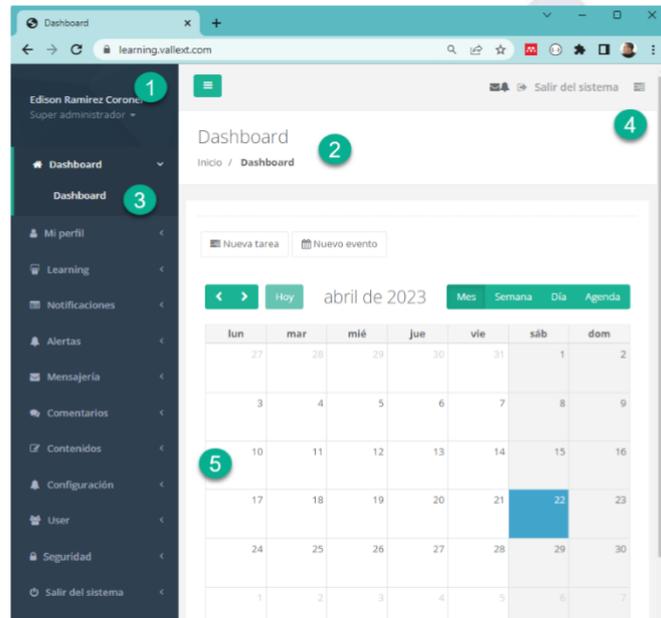
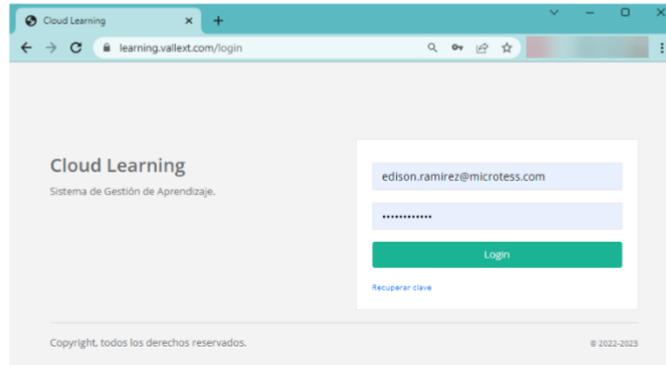
Estas secciones fueron cubiertas en cada una de las funcionalidades evaluadas.

2. Tipos de prueba planificados y ejecutados

Sobre cada sección se ejecutó la evaluación heurística, pruebas que se centran en la usabilidad.



La usabilidad de un sistema, a diferencia del diseño, es medible. Podemos determinar si cumple las funcionalidades necesarias para clasificarla en diferentes categorías de mala a excelente. Para ello, contamos con herramientas como la Evaluación Heurística, que consiste en examinar la calidad de uso de una interfaz por parte de varios evaluadores expertos, antes del test de usuarios, a partir del cumplimiento de unos principios reconocidos de usabilidad.





El equipo de pruebas está conformado por 1 integrante para lo cual las pruebas se dividieron en dos grupos con la finalidad de ejecutar dos ciclos de pruebas.

3. Secciones planificadas para evaluación

El objetivo principal de los candidatos seleccionados para las pruebas es verificar cuán enfocado está el tema de la usabilidad dentro sistemas que se encuentran disponibles a la ciudadanía en general en el **release final**. Dentro de las pruebas realizadas se consideran las siguientes secciones descritas en el cuadro 3.

SECCIONES	PLANIFICADAS	NO PLANIFICADAS	EJECUTADAS
Ingreso al sitio	X		SI
Consulta de cursos disponibles	X		SI
Ver información de un curso existente	X		SI
Enrolar a un curso disponible	X		SI
Ingresar a un curso enrolado	X		SI
Administración de un Curso	X		SI
Crear un nuevo curso	X		SI
Administración de Roles	X		SI
Administración de Módulos	X		SI
Administración de Funcionalidades	X		SI
Administración de Personas	X		SI
Administración de Usuarios	X		SI
Administración de Alertas	X		SI
Administración de Notificaciones	X		SI
Administración de Mi perfil	X		SI
Administración de Mis recursos	X		SI
Administración de Mis notas	X		SI
Administración de Mis tareas	X		SI
Administración de Mis Eventos	X		SI

Cuadro 3: Secciones planificadas y evaluadas

4. Casos de prueba planificados

En el **release final** se realizó un alcance a los casos de prueba generados en versiones anteriores. El número de casos de prueba planificados son 51 y considerando que son dos ciclos de evaluadores el total de casos de prueba a ejecutar son 102. En el cuadro 4 se muestra el detalle de los mismos.



HEURÍSTICA	CASOS DE PRUEBA	CASOS A EJECUTAR (2 EVALUADORES)	EJECUTADAS
Visibilidad del estado del sistema	3	6	Si
Relación entre el sistema	8	16	Si
Control	8	16	Si
Consistencia	7	14	Si
Errores	2	4	Si
Diseño minimalista	7	14	Si
Reconocimiento	5	10	Si
Flexibilidad	4	8	Si
Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores.	3	6	Si
Ayuda y documentación	4	8	Si
Total general	51	102	-

Cuadro 4: Casos de prueba planificados y ejecutados

Los casos de prueba se definieron usando la herramienta de evaluación heurística a partir del cumplimiento de los principios reconocidos de usabilidad, examinando la calidad de uso de una interfaz por parte de varios evaluadores.

5. Escenarios de negocio planificados y ejecutados

Los escenarios de negocio son diseñados para verificar si el sistema integrado cumple con los criterios de usabilidad planteados, validando de esta manera la usabilidad de manera integral de todas las secciones del portal. El resumen de los escenarios definidos y ejecutados se encuentra en el cuadro 5.

HEURÍSTICA	TOTAL CASOS DE PRUEBA	Nº EJECUTADOS	% DE EJECUCIÓN
Visibilidad del estado del sistema	6	6	100%
Relación entre el sistema	16	14	87.5%
Control	16	15	93.75%
Consistencia	14	14	100%
Errores	4	4	100%
Diseño minimalista	14	14	100%
Reconocimiento	10	10	100%
Flexibilidad	8	8	100%
Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores.	6	6	100%



Ayuda y documentación	8	7	87.50%
Total general	102	99	97.50%

Cuadro 5: Escenarios planificados y ejecutados

Con la ejecución de estos escenarios tenemos una cobertura del **97,5%** de los casos de prueba planificados en los dos ciclos de pruebas ejecutados en el ambiente de producción (release final).

El detalle de las heurísticas y sub heurísticas se encuentra cargado en un archivo cargado en la nube institucional con el siguiente nombre: [Evaluación Heurística Plantilla.xlsx](#)

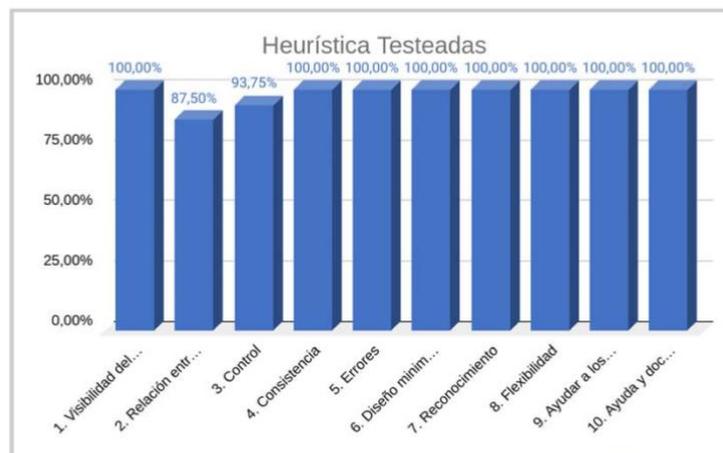


Gráfico 1: Escenarios de negocio

RESULTADOS

6. Ambiente de producción.

Las funcionalidades fueron testeadas en el ambiente de producción como usuarios de las diferentes opciones del Sistema de Gestión de Aprendizaje, con la finalidad de evaluar la usabilidad para que sea considerada en los futuros mantenimientos:

- **Portal base**
<https://learning.vallex.com/login>
- **Secciones del portal base**



UNL

Universidad
Nacional
de Loja

FEIRNNR
Maestría en Ingeniería en
Software

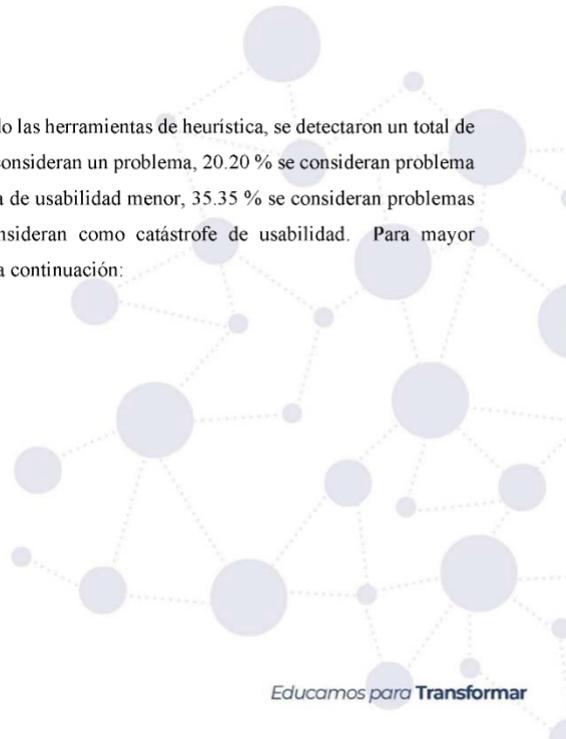
<https://learning.vallex.com/perfiles>
<https://learning.vallex.com/cursos>
<https://learning.vallex.com/notificaciones>
<https://learning.vallex.com/alertas>
<https://learning.vallex.com/mails/create>
<https://learning.vallex.com/parametros>
<https://learning.vallex.com/catalogos>
<https://learning.vallex.com/personas>
<https://learning.vallex.com/usuarios>
<https://learning.vallex.com/roles>
<https://learning.vallex.com/modulos>
<https://learning.vallex.com/funcionalidades>
<https://learning.vallex.com/sitios>

→ Navegadores

- ◆ Mozilla Firefox
- ◆ Chrome
- ◆ Brave (Android)
- ◆ Edge
- ◆ Internet Explorer

7. Bugs detectados

Durante la ejecución de las pruebas usando las herramientas de heurística, se detectaron un total de **99 errores**; de los cuales 15.15 % no se consideran un problema, 20.20 % se consideran problema estético, 21.21 % se consideran problema de usabilidad menor, 35.35 % se consideran problemas de usabilidad mayor y **8.08 %** se consideran como catástrofe de usabilidad. Para mayor comprensión ver el gráfico 1 presentado a continuación:



Educamos para Transformar

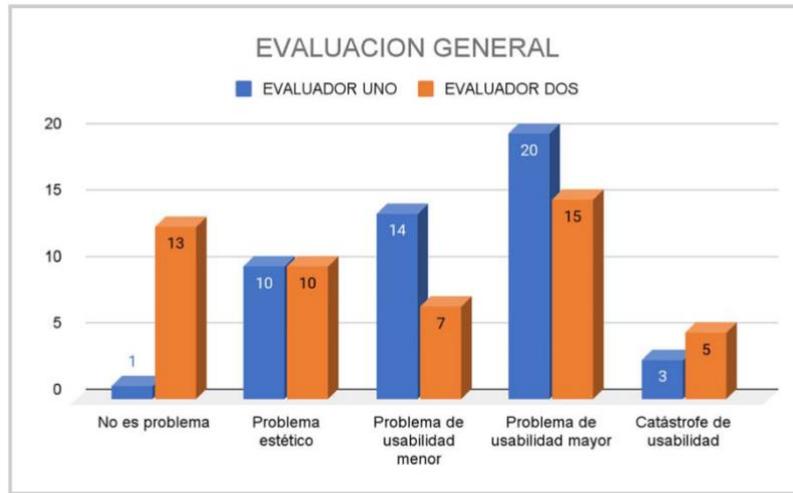


Gráfico 2: Bugs reportados por funcionalidad

8. Total de Bugs reportados por evaluador y criticidad

Los errores detectados en función de la usabilidad, son clasificados de acuerdo a la severidad. Esta severidad se usa como parámetro para evaluar que bugs detectados en la aplicación deben ser solucionados y poder priorizar la corrección.

Perfil	No es problema	Problema estético	Problema de usabilidad menor	Problema de usabilidad mayor	Catástrofe de usabilidad	TOTAL
EVALUADOR	15	20	21	35	8	99
TOTAL	15	20	21	35	8	99
%	15,15%	20,20%	21,21%	35,35%	8,08%	100%

Cuadro 6: Bugs reportados por funcionalidad y criticidad



UNL

Universidad
Nacional
de Loja

FEIRNNR
Maestría en Ingeniería en
Software

9. Bugs diferidos

Los bugs diferidos, corresponden a errores que necesitan solución, por parte de entidad prestante del servicio, al ser evaluador con rol de “usuario final” solo verificamos la usabilidad de la funcionalidad habilitada.

10. Pendientes

→ No se tiene casos de prueba pendientes.

11. Tiempos de respuesta funcionalidades críticas

Para las siguientes funcionalidades se tomaron los tiempos promedios al dar clic sobre el botón recargar:

FUNCIONALIDADES	TIEMPO (SEG)
Login	1
Crear un curso	1
Enrolar un curso	2
Administrar un curso	2

Cuadro 8: Tiempo de respuesta de funcionalidades

En las funcionalidades de estructura y oferta se podría sacrificar el tiempo de respuesta ya que su uso, aunque es constante es más esporádico que el de la matrícula que tiene un uso frecuente.

Educamos para Transformar



12. Aportes adicionales

12.1 Pruebas en producción

Se ejecutaron pruebas en producción con la base real, con el objetivo de garantizar que la aplicación funcionará bien con los datos existentes.

Para verificar el correcto funcionamiento se evaluaron 4 secciones del aplicativo web del SGA, en un ambiente de producción.

12.2 Pruebas de seguridad

A pesar de que la ejecución del plan de pruebas solo contempla la usabilidad se considera 2 casos de prueba planteados a partir de errores de seguridad detectados:

- El sitio si tiene el protocolo seguro https.
- El sitio tiene secciones con puertos abiertos, por estar alojado en un servicio de hosting compartido.

PRUEBAS DE SEGURIDADES	DE EJECUTADOS
Casos de prueba	2
Escenarios	3
Total general	6

Cuadro 9: Casos de prueba y escenarios

12.3 Recomendaciones

- En función de los errores de catástrofe de usabilidad se recomienda validar toda la información de los formularios para evitar que el navegador o el frontal se quede esperando una respuesta por largos periodos de tiempo, bloqueando la navegación total del aplicativo. Así mismo se le debe permitir a los usuarios el poder eliminar o cancelar operaciones en progreso.

Anexo 5. Certificado de traducción del resumen



**FINE-TUNED ENGLISH
LANGUAGE INSTITUTE**

Líderes en la Enseñanza del Inglés

Lic. Carlos Fernando Velastegui Aguilar
DOCENTE DE FINE-TUNED ENGLISH CÍA. LTDA.

CERTIFICA:

Que el documento aquí compuesto es fiel traducción del idioma español al idioma inglés, del resumen que forma parte del Informe del Trabajo de Titulación: "DESARROLLO DE UN SISTEMA DE GESTIÓN DE APRENDIZAJE COMO HERRAMIENTA QUE PERMITA ORGANIZAR, CENTRALIZAR CONTENIDOS ACADÉMICOS Y FACILITAR LA EXPOSICIÓN DE RECURSOS FORMATIVOS", autoría del Ing. Edison Enrique Ramírez Coronel, con CI. 1103796973, previo a la obtención del Título de Magíster en Ingeniería en Software, de la Universidad Nacional de Loja.

Lo certifica en honor a la verdad y autoriza al interesado, hacer uso del presente en lo que a sus intereses convenga.

Loja, 25 de abril de 2023.


Lic. Carlos Fernando Velastegui Aguilar
DOCENTE DE FINE-TUNED ENGLISH CÍA. LTDA.



Matriz - Loja: Macará 205-51 entre Rocafuerte y Miguel Riofrío - Teléfono: 072578899
Zamora: García Moreno y Pasaje 12 de Febrero - Teléfono: 072608169
Yantzaza: Jorge Mosquera y Luis Bastidas - Edificio Sindicato de Choferes - Teléfono: 072301329

www.fte.edu.ec