



1859

UNL

Universidad
Nacional
de Loja

Universidad Nacional de Loja
Facultad de la Energía, Las Industrias y los Recursos Naturales
No Renovables

Carrera de Computación

Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja

Specialized support software for learning C programming for the Computer Science laboratories at the National University of Loja

Trabajo de integración Curricular, previo a la obtención del título en Ciencias de la Computación.

AUTOR:

Juan Carlos Armijos Sánchez

DIRECTOR:

Ing. José Oswaldo Guamán Quinche, Mg. Sc.

Loja – Ecuador

2025

CERTIFICACIÓN

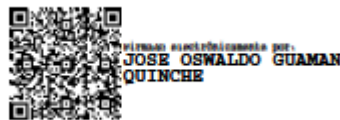
Loja, 20 de enero del 2025

Ing. José Oswaldo Guamán Quinche, Mg. Sc.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Integración Curricular denominado: **Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja**, previo a la obtención del título de **Ingeniero en Ciencias de la Computación**, de la autoría del estudiante **Juan Carlos Armijos Sánchez**, con cédula de identidad Nro. **1150329884**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.



Ing. José Oswaldo Guamán Quinche, Mg. Sc.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

AUTORIA

Yo, **Juan Carlos Armijos Sánchez**, declaro ser autor del presente Trabajo de Integración Curricular y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Integración Curricular o de Titulación, en el Repositorio Digital Institucional – Biblioteca Virtual.



Firma

Cedula de identidad: 1150329884

Fecha: 20 de enero de 2025

Correo electrónico: juan.c.armijos@unl.edu.ec

Teléfono: 0985559155

Carta de autorización por parte de los autores, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Integración Curricular.

Yo, **Juan Carlos Armijos Sánchez**, declaro ser el autor del Trabajo de Integración Curricular denominado: **Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja**, 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 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 Integración Curricular que realice un tercero.

Para constancia de esta autorización, suscribo, en la ciudad de Loja, a los veinte días del mes de enero de dos mil veinticinco.



Firma

Autor: Juan Carlos Armijos Sánchez

Cedula de identidad: 1150329884

Dirección: Francisco Santander y Francisco de Nariño – Loja – Ecuador

Correo electrónico: juan.c.armijos@unl.edu.ec

Teléfono: 0985559155

DATOS COMPLEMENTARIOS:

Director del Trabajo de Integración Curricular: Ing. José Oswaldo Guamán Quinche, Mg. Sc.

Dedicatoria

En primer lugar, deseo expresar mi infinita gratitud a mis padres, Carlos Armijos y Mercy Sánchez, quienes han sido mi pilar fundamental en todo este camino. Gracias por su amor incondicional, sus consejos sabios y su fe inquebrantable en mí, incluso en los momentos más desafiantes. A mis hermanos, Michael, Camila y Lucas, quienes con su cariño, apoyo y compañía me recordaron siempre la importancia de la familia como fuente de fortaleza y motivación.

A mi amada Arianna, quien ha sido mi refugio, mi apoyo y mi inspiración. Gracias por estar a mi lado, por creer en mí y por brindarme el amor que me impulsa a dar lo mejor de mí cada día. A mis amigos, quienes con su amistad, risas y palabras de aliento hicieron que este trayecto fuera más llevadero y lleno de memorias inolvidables en especial a mi amigo Luis.

Finalmente, a todos aquellos con quienes he compartido este camino, ya sea en las aulas, en los proyectos o en los momentos de descanso. Cada uno de ustedes ha dejado una huella en mi vida, contribuyendo de una u otra manera a mi crecimiento personal y profesional.

A todos ustedes, mi más profundo agradecimiento por ser parte de esta etapa tan importante en mi vida. Este logro también es suyo.

Juan Carlos Armijos Sánchez

Agradecimiento

Quiero expresar mi más sincero agradecimiento a la Universidad Nacional de Loja, en especial a su prestigiosa Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables. Durante mi formación, tanto el personal administrativo como los docentes de esta facultad han sido piezas clave en mi desarrollo humano y profesional, ofreciéndome orientación con un notable profesionalismo y una calidad humana que marcó profundamente mi carrera universitaria.

Mi especial gratitud al Ing. José Oswaldo Guamán Quinche, Mg. Sc, por su invaluable tiempo, dedicación y guía. Su compromiso y apoyo constante fueron fundamentales para la realización y culminación exitosa del presente Trabajo de Integración Curricular.

A mi familia, pareja y amigos, les agradezco profundamente por ser mi mayor fortaleza. Su amor, compañía y apoyo incondicional me brindaron la motivación necesaria para superar cada desafío y alcanzar esta meta tan importante en mi vida.

A todos ustedes, mi reconocimiento y gratitud eterna por formar parte de este camino. Este logro también es suyo.

Juan Carlos Armijos Sánchez

Índice de contenidos

Portada	i
Certificación	ii
Autoría.....	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
índice de contenidos.....	vii
Índice de figuras	xi
Índice de tablas	x
Índice de anexos.....	xii
1 Título.....	1
2 Resumen	2
Abstract	3
3 Introducción.....	4
4 Marco Teórico	6
4.1 Antecedentes.....	6
4.2 Fundamentación Teórica.....	6
4.2.1 Software de apoyo educativo	6
4.2.2 Programación	7
4.2.3 Entornos para el aprendizaje de programación	7
4.2.4 Tecnologías de desarrollo	10
4.2.5 Single Page Application (SPA).....	13
4.2.6 Metodología de desarrollo ágil Extreme Programming (XP).....	13
4.2.7 Modelo de Aceptación Tecnológica (TAM).....	16
4.3 Trabajos relacionados	18

5	Metodología.....	20
5.1	Área de Estudio.....	20
5.2	Procedimiento	20
5.2.1	Objetivo Específico 1. Construir un software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante la metodología XP.	20
5.2.2	Objetivo Específico 2. Determinar la aceptación tecnológica del software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante el modelo TAM	22
5.3	Recursos.....	23
5.3.1	Recursos científicos.....	23
5.3.2	Recursos técnicos	23
5.4	Participantes.....	25
6	Resultados	26
6.1	Objetivo 1: Construir un software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante la metodología XP.	26
6.1.1	Fase de planificación.....	26
6.1.2	Fase de diseño.....	32
6.1.3	Fase de codificación.	36
6.1.4	Fase de pruebas.....	37
6.1.5	Fase de lanzamiento.....	40
6.2	Objetivo 2: Determinar la aceptación tecnológica del software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante el modelo TAM.	45
6.2.1	Fase de elaboración.....	45
6.2.2	Fase de medición	46
6.2.3	Fase de análisis.	47
7	Discusión	53

7.1	Objetivo 1: Construir un software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante la metodología XP.	53
7.2	Objetivo 2: Determinar la aceptación tecnológica del software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante el modelo TAM.	55
8	Conclusiones	57
9	Recomendaciones	59
9.1	Trabajos futuros.....	59
10	Bibliografía	60
11	Anexos	64

Índice de tablas:

Tabla 1. Escala de likert.....	17
Tabla 2. Ejemplos resultados de evaluación TAM.....	17
Tabla 3. Trabajos relacionados.	18
Tabla 4. Actores del sistema	26
Tabla 5. Requisitos funcionales del sistema	27
Tabla 6. Requisitos no funcionales del sistema	27
Tabla 7. Modelo HU	29
Tabla 8. Tabla de historias de usuario.....	29
Tabla 9. Plan de iteración.....	30
Tabla 10. Resumen general de las pruebas unitarias realizadas	37
Tabla 11. Resumen general de las pruebas de aceptación realizadas	39
Tabla 12. Variables para medir la aceptación de los usuarios.....	45
Tabla 13. Valoraciones numéricas para cada respuesta escogida por el usuario	46
Tabla 14. Matriz de resultados en base a cada constructo	47

Índice de figuras:

Figura 1. Diagrama de dominio. [Fuente propia].....	32
Figura 2. Diagrama de casos de uso [Fuente propia]	33
Figura 3. Diagrama de clases [Fuente propia].....	34
Figura 4. Diagrama de despliegue [Fuente propia]	35
Figura 5. Prototipo de ejemplo [Fuente propia]	35
Figura 6. Estructura del ambiente backend [Fuente propia].	36
Figura 7. Estructura del ambiente frontend [Fuente propia].	37
Figura 8. Archivo de configuración Docker del Frontend.....	40
Figura 9. Archivo de configuración Docker del Backend.	40
Figura 10. Archivo Docker Compose de los servicios del software de apoyo.....	41
Figura 11. Conexión al servidor mediante SSH.	41
Figura 12. Inicio de sesión dentro de la MV con Centos 9	42
Figura 13. Transferencia de la aplicación al servidor.	42
Figura 14. Levantamiento de servicios y verificación de contenedores activos.	42
Figura 15. Configuración de los puertos en el servidor.....	43
Figura 16. Configuración de Apache para servir el frontend.	43
Figura 17. Archivo de configuración del servicio Camigo.	44
Figura 18. Validación de contenedores activos.	44
Figura 19. Software de apoyo desplegado.....	44
Figura 20. Respuestas graficadas de Utilidad Percibida [Fuente propia].....	48
Figura 21. Respuestas graficadas de la Facilidad de Uso Percibida [Fuente propia]..	49
Figura 22. Respuestas graficadas de la Actitud Hacia el Uso [Fuente propia].	51
Figura 23. Respuestas graficadas de la Intención de Uso [Fuente propia].....	52

Índice de anexos:

Anexo 1. Entrevistas realizadas a los docentes del área de programación	64
Anexo 2. Requerimientos esenciales del sistema.....	73
Anexo 3. Historias de usuario	79
Anexo 4. Acta de socialización del temario	121
Anexo 5. Implementación de la metodología de desarrollo.....	126
Anexo 6. Arquitectura 4+1	227
Anexo 7. Resultados de la aceptación tecnológica del sistema mediante TAM	248
Anexo 8. Manual de usuario para estudiante	258
Anexo 9. Manual de usuario para docente	281
Anexo 10. Manual de usuario para administrador.....	300
Anexo 11. Pruebas de aceptación realizadas a los estudiantes del primer ciclo de la carrera de Computación de la materia “Teoría de la programación”	321
Anexo 12. Pruebas de aceptación realizadas a los docentes del área de programación	326
Anexo 13. Pruebas de aceptación realizadas al encargado de los laboratorios de software de la carrera de computación, Ing. Luis Sinche.....	329
Anexo 14. Pruebas de aceptación realizadas al director de la carrera de computación, Ing. Pablo Ordoñez.....	332
Anexo 15. Reporte de plagio en la herramienta Turnitin	335
Anexo 16. Documentos firmados electrónicamente	340
Anexo 17. Certificado de traducción del resumen	342

1 Título

Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja

Specialized support software for learning C programming for the Computer Science laboratories at the National University of Loja

2 Resumen

En la Universidad Nacional de Loja no se dispone de un software de apoyo específico basado en el plan de estudios del primer ciclo de la carrera de Computación, donde se enseña programación en C. Por ello, el presente Trabajo de Integración Curricular (TIC) tiene como objetivo general desarrollar un software de apoyo especializado en el aprendizaje de programación en C para evaluar su aceptación tecnológica dentro de los laboratorios de la carrera de Computación en la Universidad Nacional de Loja. Para cumplir con este propósito, se plantean dos objetivos específicos. El primero consiste en construir un software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera, mediante la metodología de desarrollo de software eXtreme Programming (XP). El segundo objetivo busca determinar la aceptación tecnológica del software desarrollado, utilizando el Modelo de Aceptación Tecnológica (TAM) haciendo uso de los cuatro constructos de este modelo (Utilidad percibida, Facilidad de uso percibida, Actitud hacia el uso, Intención de uso). El software se desarrolla empleando tecnologías como Node.js, Express y React, y se implementa en los laboratorios de la carrera de Computación, integrando funcionalidades que permiten a los estudiantes practicar y reforzar los contenidos fundamentales del lenguaje C, alineados al plan de estudios. La evaluación de la aceptación tecnológica se realiza a través de encuestas aplicadas a los usuarios finales, basadas en el modelo TAM, donde los resultados obtenidos indican un nivel de aceptación del 93.7%, validando la idoneidad del software como herramienta de apoyo para el aprendizaje de programación en C.

Palabras clave: Programación en C, Metodología XP, Software de apoyo, Modelo de Aceptación Tecnológica (TAM).

Abstract

At the National University of Loja, there is no specific support software aligned with the study plan of the first cycle of the Computer Science degree program, where C programming is taught. For this reason, this Curricular Integration Project (CIP) aims to develop specialized support software for learning C programming and evaluate its technological acceptance within the laboratories of the Computer Science degree program at the National University of Loja. To achieve this goal, two specific objectives were defined. The first is to build specialized support software for learning C programming in the degree program's laboratories using the eXtreme Programming (XP) software development methodology. The second objective is to determine the technological acceptance of the developed software using the Technology Acceptance Model (TAM), employing the four constructs of this model (Perceived usefulness, Perceived ease of use, Attitude toward use, and Intention to use). The software is developed using technologies such as Node.js, Express, and React and implemented in the Computer Science degree program's laboratories. It includes functionalities that allow students to practice and reinforce the fundamental concepts of the C language, aligned with the study plan. The evaluation of technological acceptance is conducted through surveys applied to end users based on the TAM model, with results indicating a 93.7% acceptance rate, validating the software's suitability as a support tool for learning C programming.

Keywords: C Programming, XP Methodology, Supporting Software, Technology Acceptance Model (TAM).

3 Introducción

Un software de apoyo es una herramienta tecnológica diseñada para facilitar el proceso de aprendizaje en áreas específicas, proporcionando características que permiten fortalecer conocimientos y practicar habilidades. En el ámbito de la programación, este tipo de software resulta esencial para apoyar a los estudiantes en la comprensión de conceptos, la realización de ejercicios prácticos y la adquisición de competencias técnicas que son esenciales en el campo de la informática [4]. Estas aplicaciones no solo complementan la formación en el aula, sino que también ofrecen un entorno estructurado para practicar de forma autónoma, adaptándose al nivel y ritmo de aprendizaje de cada estudiante; esto las convierte en un recurso valioso para áreas como la programación en C, donde la práctica constante y el acceso a recursos específicos son fundamentales para reforzar el aprendizaje [5].

La problemática identificada en la carrera de Computación radica en la inexistencia de un software de apoyo especializado en el aprendizaje de programación en C, que esté diseñado específicamente en función del plan de estudios del primer ciclo (Silabo de materia de Teoría de la Programación). Este plan incluye los fundamentos de programación en C, pero no dispone de una herramienta que ofrezca a los estudiantes la posibilidad de practicar y reforzar estos conceptos de manera autónoma, facilitando un recurso práctico que pueda ser utilizado tanto dentro como fuera del aula.

La pregunta de investigación que guía este trabajo es: **¿Qué aceptación tecnológica se logrará obtener al utilizar el software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja, mediante el método de aceptación tecnológica TAM?** Esta pregunta busca evaluar cómo los usuarios, en este caso los estudiantes de la carrera de Computación, perciben el software desarrollado, enfocándose en los cuatro constructos de este modelo de aceptación [35].

La importancia de este problema radica en la necesidad de contar con herramientas tecnológicas que se alineen con el plan de estudios del primer ciclo de la carrera de Computación, específicamente en la asignatura de “Teoría de la programación”. Disponer de un software de apoyo especializado proporciona una alternativa para complementar las clases impartidas, ofreciendo a los estudiantes un recurso que permita revisar y practicar contenido según las necesidades del usuario. La relevancia de desarrollar una solución adaptada a este contexto específico reside en su potencial para integrar tecnología en el proceso de aprendizaje, promoviendo un acceso más estructurado a los contenidos del curso, acorde con los requerimientos de la carrera.

Con el fin de responder a la pregunta de investigación, se planteó el siguiente **objetivo general**: "Desarrollar un software de apoyo especializado en el aprendizaje de programación en C para evaluar su aceptación tecnológica dentro de los laboratorios de la carrera de Computación en la Universidad Nacional de Loja". Para alcanzarlo, se definieron dos **objetivos específicos**. El primero consiste en "Construir un software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante la metodología XP". Este objetivo se centra en la aplicación de una metodología de desarrollo ágil para desarrollar un software funcional y adecuado a los requerimientos del plan de estudios [27]. El segundo objetivo es "Determinar la aceptación tecnológica del software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante el modelo TAM". Este objetivo se orienta a evaluar la percepción de los usuarios sobre la utilidad, facilidad de uso, actitud hacia el uso e intención de uso del software, siguiendo las fases propuestas por el modelo TAM [35].

El alcance de este proyecto se enfoca en el desarrollo de un software de apoyo especializado en el aprendizaje de programación en C, diseñado específicamente para los laboratorios de la carrera de Computación de la Universidad Nacional de Loja. Este software está alineado con el plan de estudios del primer ciclo de la carrera y tiene como objetivo principal brindar a los estudiantes una herramienta tecnológica de apoyo para su aprendizaje de programación en C. El software fue desarrollado utilizando la metodología de desarrollo XP, garantizando un proceso ágil. Además, se evaluará su aceptación tecnológica mediante el Modelo de Aceptación Tecnológica (TAM), considerando a los estudiantes que utilizan los laboratorios de computación como el área de estudio principal.

El presente TIC se compone por las siguientes secciones: en el **Marco Teórico** se encuentran los antecedentes, fundamentación teórica, tecnologías y metodologías utilizadas y trabajos relacionados; en la **Metodología** está presente el área de estudio, el procedimiento seguido para ambos objetivos del TIC, los recursos científicos y técnicos necesarios para su ejecución, y los participantes; la sección de **Resultados** se divide en los dos objetivos que dictaminan el alcance del TIC; la **Discusión** que expone la interpretación de los resultados obtenidos; la sección de **Conclusiones** que aborda los hitos destacables del proyecto; las **Recomendaciones** que detallan las sugerencias y trabajos futuros y, finalmente los Anexos que presentan material complementario para este TIC.

4 Marco Teórico

4.1 Antecedentes

El avance constante en los ámbitos educativo y tecnológico ha generado una creciente necesidad de herramientas especializadas que faciliten el aprendizaje, especialmente en áreas técnicas como la programación. En la carrera de Computación de la Universidad Nacional de Loja, esta necesidad se hace evidente en la enseñanza de programación en C, una materia fundamental que presenta desafíos significativos para los estudiantes debido a que es el inicio en su aprendizaje en el mundo de la programación. Aunque se han implementado diversos esfuerzos por parte de los docentes, los recursos disponibles suelen ser genéricos y no están alineados con las necesidades específicas del plan de estudios de la carrera. Esta desconexión puede afectar tanto la calidad del aprendizaje como la motivación de los estudiantes, dificultando el desarrollo de habilidades clave en programación. La integración de una herramienta de apoyo especializada en el aprendizaje de programación en C tiene el potencial de apoyar este escenario. Diseñada específicamente para alinearse con el plan de estudios, una herramienta de este tipo podría abordar los puntos críticos del aprendizaje.

4.2 Fundamentación Teórica

4.2.1 *Software de apoyo educativo*

El Software de apoyo educativo se lo conoce como un medio pedagógico en el cual su objetivo principal es facilitar el proceso de enseñanza y aprendizaje. Este programa a través de una plataforma digital colabora con el proceso de enseñanza y aprendizaje que facilitan la adquisición de conocimientos, de esta manera cualquier alumno puede afianzar sus conocimientos en las más diversas áreas desde lo más práctico, hasta las más teóricas. Los softwares educativos pueden ser usados dentro del aula, en el contexto escolar, o incluso en un local externo a este ambiente, como en casa. Los softwares educativos también son conocidos como una herramienta innovadora la misma que despierta el interés del estudiante facilitando su aprendizaje [1]. El software de apoyo educativo en el entorno de planteles educativos, ayuda a crear bases en las diferentes asignaturas mejorando la interactividad entre docente – estudiante. Este tipo de softwares están predeterminado al aprendizaje y enseñanza autodirigida, diseñado con el fin de facilitar sus procesos de aprendizaje. En resumen, el software de apoyo en el aprendizaje de programación es una herramienta tecnológica con el fin de ayudar a los

estudiantes a comprender y practicar los conceptos teóricos y prácticos pueden ser utilizados tanto como herramienta dentro del aula como fuera de la misma [2].

Los sistemas informáticos educativos pueden tocar temas de diferentes asignaturas de forma diversificada, en su entorno debe ofrecer interacción, dado que estos softwares comparten 5 características primordiales como: “Su material es elaborado con fines didácticos, Utilizan las computadoras como soportes, Son interactivos, El trabajo del estudiante es individual, Su uso es fácil”. Necesariamente deben existir todas estas características dentro del software educativo, ya que genera un resultado un sistema autocontenido y amigable logrando captar la atención del usuario.

4.2.2 Programación

La programación se la denomina al proceso de diseñar, codificar, depurar y de brindar instrucciones que una computadora debe seguir para realizar una tarea específica, las mismas deben escribirse en lenguajes de programación lo cual permite a los programadores dar a entender a la computadora lo que se desea realizar [3]. La programación es una habilidad primordial en el mundo informático y se utiliza en una amplia variedad de campos profesionales desde la medicina hasta la educación [4]. Detrás de todos estos programas informáticos que se conocen y se usan de manera cotidiana las mismas que facilitan diversas actividades cotidianas, detrás de todo esto existe todo un proceso para poderlos crear el mismo que se lo reconoce conocido programación. Por medio de la programación se establecen pasos a seguir para la creación de un código fuente de los diversos programas informáticos. Dicho código indicara al programa informático que tiene que hacer y cómo debe realizarlo, este se debe realizar mediante el uso de algoritmos, que se logran explicar cómo reglas o instrucciones a seguir para resolver el problema y lograr su objetivo.

4.2.3 Entornos para el aprendizaje de programación

4.2.3.1 Aprendizaje de programación.

El aprendizaje de programación es un proceso el cual implica adquirir varias habilidades y conocimientos para realizar un código de computadora y desarrollar su software. En la actualidad, la programación se ha vuelto cada vez más importante en diversos campos profesionales debido al constante avance tecnológico. Cabe recalcar que no solo los ingenieros e informáticos generan código, sino que también profesionales de otras disciplinas, como lo que son economistas, matemáticos, físicos y geógrafos, los cuales han encontrado en la programación una herramienta esencial para lograr enfrentar los desafíos existentes en la actualidad, este recurso no solo es valioso para quienes desean dedicarse profesionalmente al

desarrollo de software, sino también para quienes buscan crear su propio sitio web o aplicación. Además, el aprendizaje de programación fomenta habilidades como la resolución de problemas y pensamiento lógico, debido a que requiere un enfoque sistemático y estructurado para resolver los desafíos que se presentan. Por eso en la actualidad, diversos países han introducido asignaturas de fundamentos de informática y programación en las escuelas para lograr que de esta manera los niños puedan explorar y descubrir las posibilidades que ofrece la programación desde temprana edad [5].

4.2.3.2 La Computadora y la Enseñanza de la Programación.

La computadora resulta ser una herramienta útil para tener conocimientos en la programación y la enseñanza de distintas asignaturas. Para esto se debe contar con entornos de aprendizaje que conduzcan e induzcan dentro de este proceso educativo, por ende, la calidad de este entorno está ligada a lograr un proceso efectivo de enseñanza y de aprendizaje. Este tipo de entornos debe contemplar lo que es un nivel cognitivo, la capacidad de percepción, de respuesta, de abstracción de la audiencia a la que va dirigido, la capacidad de interacción hombre-máquina, etc. Para esto se debe disponer de sistemas específicos que sean amigables con el usuario. Además, la programación suscita interés psicopedagógico debido a que sus efectos en las capacidades cognitivas, desarrollando una actividad intelectual la cual permite establecer planificaciones y estrategias, construir algoritmos, estructurar instrucciones, analizar y comprender los programas propios o escritos por otros, aprender la sintaxis de los distintos lenguajes de programación, comparar estrategias entre los programadores expertos y principiantes.

Los entornos de lenguaje de programación ofrecen diversas capacidades cada vez más útiles al programador, como lo que es la posibilidad de indentación, depuradores. También pretender plantear un desarrollo de entornos específicos que ayuden y conduzcan efectivamente a los estudiantes en el proceso de aprendizaje de la programación que sea más gráfica, diversa y comprensible [6].

4.2.3.3 Enfoques de aprendizaje.

Los enfoques de aprendizaje se refieren a las estrategias y métodos que los estudiantes utilizan para adquirir y procesar información. Estos enfoques pueden clasificarse generalmente en dos categorías: centrados en el contenido y centrados en el estudiante. Los enfoques centrados en el contenido se enfocan en la transmisión de conocimientos específicos, mientras que los enfoques centrados en el estudiante promueven la participación activa del alumno en su propio proceso de aprendizaje [7].

4.2.3.3.1 Enfoques centrados en el contenido.

El enfoque centrado en el contenido o (tradicional unidireccional) se lo conoce como un método pedagógico, el mismo que tiene relevancia en el contenido académico indicado, los mismos se tratan en transmitir tiene como objetivos indicar los métodos, indicar el conocimiento de los métodos de análisis.

4.2.3.3.2 Enfoques centrados en el estudiante.

Los enfoques centrados en el estudiante fueron los seleccionados para este proyecto, ya que este software de apoyo está diseñado para que los estudiantes sean los protagonistas de su aprendizaje, es por ello que se seleccionó tres enfoques centrados en el estudiante para su correcto análisis:

- **Aprendizaje basado en proyectos (PBL):** Este modelo de aprendizaje se basa en que los estudiantes trabajen de una manera participativa con lo cual puedan plantear, implementar y evaluar proyectos los cuales contienen aplicación dentro del mundo real, fuera del aula, esta herramienta a resultado útil para docentes dentro del aprendizaje no solo de su contenido, sino también del uso efectivo en TIC, cabe destacar que este no debe ser confundido con el aprendizaje por problemas [8]. Este aprendizaje es una metodología de docentes basada en sus estudiantes para que sea el personaje principal de sus conocimientos gracias a esto adquieren varias habilidades y actitudes esta es una perfecta opción metodológica para promover el desarrollo profesional por estas metodologías didácticas innovadoras [9]. Este aprendizaje se basa en experiencias basadas en interés y necesidades de estudiantes la cual tiene objetivos de aprendizaje en problemáticas reales, este enfoque puede ser usado tanto como intra e interdisciplinaria, la cual tiene como favorecimiento un trabajo colaborativo entre docente, estudiante.
- **Aprendizaje autodirigido:** Se lo conoce como un aprendizaje con diseño, conducción y evaluación el cual su elemento base es el control que posee el estudiante para decidir dentro de su estudio, pudiendo seleccionar variedad de recursos que puede implementar en su proceso de aprendizaje, este también suele asociarse con un enfoque reflexivo el mismo que rebasa los límites en su estudio ya que se regula según necesidades e intereses [10]. Se puede considerar como un proceso en el cual el estudiante toma la iniciativa en su aprendizaje teniendo en cuenta sus recursos y necesidades implementado estrategias de aprendizaje propicias evaluando sus resultados.

- **Aprendizaje basado en la resolución de problemas:** El aprendizaje basado en la resolución de problemas se basa en el pensamiento reflexivo que realizan los alumnos para obtener una solución a un problema planteado por el docente, este se manifiesta como un punto de partida donde se obtiene nuevos conocimientos ya que su estrategia es eficaz y flexible, todo esto favorece a sus habilidades dentro de la búsqueda y su manejo de información, desarrollando habilidades investigativas [11]. Este Aprendizaje es una estrategia de aprendizaje para obtener conocimientos, habilidades y aptitudes importantes donde elaboran su propio diagnóstico acorde a sus necesidades, ayuda a desarrollar el pensamiento crítico, buscando que el alumno aprenda las respuestas a sus problemas.

4.2.4 Tecnologías de desarrollo

4.2.4.1 React

React se la conoce como una librería JavaScript, creada por Facebook, en la cual su funcionamiento se basa en la abstracción de las vistas mediante el uso de componentes. Un componente de estos puede ser tanto como un formulario de entrada, un botón, o cualquier otra cosa del interfaz de usuario. Esto proporciona un contraste con respecto a enfoques anteriores ya que, por diseño, React no está vinculado al árbol DOM. Puedes utilizarlo por ejemplo para desarrollar aplicaciones móviles. Para usarlo tendrás que complementarlo con otras librerías que te den aquello que te falte, ya que React se centra únicamente en la vista. Esto brinda un contraste con respecto a utilizar frameworks que traen mucho más de serie. Ambos enfoques tienen sus méritos. En este libro nos centraremos en el uso de librerías. Las ideas presentadas por React han tenido su influencia el desarrollo de frameworks, aunque más importante es que nos ha ayudado a entender cómo de bien encaja el pensar en componentes en el desarrollo de aplicaciones web. [12].

4.2.4.2 NodeJS

NodeJS es un entorno de ejecución multiplataforma para el lenguaje de programación JavaScript. Es de código abierto y su licencia es de tipo MIT Licence, lo que significa que cualquier persona puede descargarlo e instalarlo en su computadora sin tener que pagar una licencia. Existe una gran comunidad alrededor de todo el mundo involucrada con NodeJS. En la actualidad, NodeJS cuenta con un gran número de módulos y componente orientados y optimizados para networking (redes), que sirven de soporte para el manejo de los estándares y protocolos más populares de internet como DNS, HTTP, TCP, TLS/SSL y UDP. NodeJS utiliza

un modelo de entrada/salida sin bloqueo y que se orienta a eventos, lo que significa que puede manejar muchas solicitudes de forma simultánea. Lo que hace a NodeJS como una herramienta ideal para aplicaciones en tiempo real. Una de las características principales de NodeJS es su facilidad de adaptación con módulos externos a través del sistema de gestión de paquetes npm [13].

4.2.4.3 ExpressJS

ExpressJS está construido en base a Node la cual brinda una API simple, la cual se la describe como abstracción sobre el módulo HTTP de API, Express proporciona diversidad de funciones sobre su modulo, dentro de este existe el middleware el cual es su principal característica ya que se encarga de respuestas y solicitudes, gracias a esta funcionalidad se puede ejecutar en cualquier código y realizar cambios en diferentes objetos de solicitud y respuesta. También se lo utiliza para iniciar un servidor a través del createServer la cual necesita tres parámetros la solicitud objeto, el objeto de respuesta y la función del middleware. Aquí se encuentra también el enrutamiento, archivos estáticos [14]. Express JS se la conoce también como un framework que se construye a base de Nodejs la cual brinda una API para algunas funciones principales sobre el módulo HTTP la cual simplifica las APIs facilitando una organización con middleware y enrutamiento, facilitando igual la representación de vistas HTML dinámicas definiendo un estándar extensible fácil de implementar.

4.2.4.4 MySQL

Se conoce a MySQL como un sustituto mejorado y compatible con MySQL, debido a que no estaban satisfechos con la calidad de las presentes versiones de MySQL y su lenta mejoría, MySQL encontró varias formas para mejorar las funcionalidades de los usuarios y desarrolladores algunas mejoras que realizaron son la proporciona de estadísticas de índice y tabla mediante la función denominada “Estadísticas de usuario” la cual crea un esquema de información de comandos denominado FLUSH y SHOW, Maria DB es una base de datos relacionados a un código abierto que pretende continuar como MySQL mejorando continuamente ya que varios desarrolladores tienen como primera opción a Maria DB como alternativa [15].

4.2.4.5 Definición de tecnologías

La elección de las tecnologías se basó en base a las siguientes pautas:

4.2.4.5.1 *React*.

La elección de React como parte integral del desarrollo se basó en su arquitectura orientada a componentes, y en la necesidad de construir una interfaz de usuario dinámica y modular, la decisión se sustenta en [16]:

- **Interfaz dinámica y modular:** La estructura de componentes de React facilita la creación de una interfaz de usuario interactiva y modular lo que facilita el desarrollo y mantenibilidad del código.
- **Reutilización y mantenibilidad:** React permite la fácil reutilización de componentes, lo que permitió en el desarrollo hacerlo de forma ágil y así mismo volverlo más mantenible a largo plazo por su modularidad.
- **Adaptabilidad:** La capacidad de complementar React a otras librerías según sean las necesidades del proyecto es algo clave que se tomó en cuenta para adaptarlo a las exigencias del software de apoyo.

4.2.4.5.2 *NodeJS*

La elección de NodeJS se fundamenta por su capacidad para construir aplicaciones web escalables y en tiempo real, aspectos que se consideraron cruciales para este proyecto, la decisión se sustenta en [17]:

- **Rendimiento y escalabilidad:** NodeJS, con su modelo de entrada/salida sin bloqueo, proporciona un rendimiento eficiente para soportar múltiples usuarios simultáneos, lo cual es esencial para un entorno de aprendizaje en línea concurrente.
- **Soporte para aplicaciones en tiempo real:** La capacidad de manejar eventos en tiempo real es fundamental para la interactividad y la retroalimentación instantánea, elementos clave en la enseñanza de programación.

4.2.4.5.3 *ExpressJS*

La elección de expressJS, se fundamenta en su capacidad para potenciar el desarrollo del backend y gestionar solicitudes entrantes, permitiendo la de una API Restful eficiente y eficaz, la decisión se sustenta en [14]:

- **Potenciando el backend:** ExpressJS, como framework minimalista y flexible sobre NodeJS, facilitó la creación de rutas, el manejo de solicitudes y respuestas, y la implementación de lógica de servidor que fue necesaria para el software de apoyo.

- **Facilidad de creación de API Restful:** ExpressJS, facilitó la creación de una API Restful, lo que permitió una comunicación efectiva entre el backend y el frontend.

4.2.4.5.4 *MySQL*

La elección de MySQL como sistema gestor de base de datos se fundamenta en su capacidad para almacenar y gestionar eficientemente los datos necesarios para el software de apoyo, la decisión se sustenta en [15]:

- **Gestión eficiente de datos:** MySQL ofrece la capacidad de almacenar grandes cantidades de datos lo que facilita el correcto funcionamiento del sistema de apoyo.
- **Facilidad de integración y estabilidad:** La estabilidad y gran familiaridad con MySQL facilitaron su integración con el resto de las tecnologías seleccionadas, lo que permitió un desarrollo fluido del software.

4.2.5 *Single Page Application (SPA).*

Single Page Application o Aplicación de página única, se la denomina a una aplicación web donde diversidad de pantallas se muestran en una sola sin tener que recargar el navegador, se lo puede denominar también como un punto de entrada dentro del archivo index.html con el cual se puede acceder de una manera individual que nuestre su contenido [17]. El SPA este compuesto por diversos componentes cada uno individual, los cuales se pueden reemplazar o a su vez ser actualizados de manera individual, permite de una forma más flexible manejar los datos ya que requiere de una interfaz más interactiva y da como resultado una mejor experiencia de usuario.

4.2.6 *Metodología de desarrollo ágil Extreme Programming (XP).*

Es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck, autor del primer libro sobre la materia, Extreme Programming Explained: Embrace Change (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad [18].

4.2.6.1 **Valores de XP**

Según [18] los valores de XP son:

- **Simplicidad:** La cual está enfocada en plantear objetivos y en mitigar fallos existentes que suelen suceder de la forma más rápida.

- **Coraje:** Se basa en comunicar de manera adecuada los avances y estado del proyecto que se desarrolla, el no documentar problemas que existan no ayudara mejorar el proyecto.
- **Comunicación:** Quienes estén interesados en el proyecto, deben realizar la observación critica ya que tienen como objetivo mejorar el desarrollo del proyecto.
- **Retroalimentación:** Este valor pretende realizar la mayor responsabilidad posible, ya que el producto final de la interacción se debe presentar a tiempo con el objetivo a revisar por posibles mejoras que se puedan implementar a tiempo.
- **Respeto:** Debe existir respeto en todos los responsables del proyecto, ya que se valora el trabajo de cada miembro, para dar como resultado un proyecto final con valor agregado.

4.2.6.2 Fases de la metodología XP

Existen varias fases dentro de la metodología XP según [19] son:

- **Planificación:** Es un dialogo entre los integrantes de proyecto, el cual cuenta con usuarios, interacciones dando como producto lo que se espera con interacción.
- **Diseño:** esta metodología se basa en uso de diseños simples y claros los cuales permiten aprovechar su simplicidad.
- **Codificación:** En esta fase se realiza un código de calidad, gracias a esto cualquier desarrollador podrá trabajar y entender.
- **Pruebas:** Aquí se realiza la verificación de funcionalidad del sistema con pruebas acordadas al proyecto.
- **Lanzamiento:**

4.2.6.3 Historias de usuario

Las historias de usuario de la metodología XP es una técnica usada para lograr especificar los requisitos del software, donde se describe de manera breve las características y funcionalidades del software, Cabe recalcar que el uso de estas historias de usuario es dinámica y flexible y que puede ser mejorada en cualquier momento por una más concreta y cada historia de usuario debe ser entendible, entonces se puede concluir que estas son descripciones breves y centradas en la funcionalidad dada por el usuario especificada en el software, estas siguen un formato simple, independiente y completa que priorizan el valor del usuario [20].

4.2.6.4 Pruebas de software.

Este es un conjunto entre técnicas y actividades las cuales son usadas para su evaluación y verificación de rendimiento validando la calidad de software antes de pasar al proceso de comercialización y aplicación en una empresa ya que su principal propósito se basa en identificar y reportar posibles fallas dentro del software ya que así se asegura que cumplan los requisitos y expectativas para el usuario. También existe una idea errónea en la cual se base en que probar el software es simplemente ejecutar y observa su desenvolvimiento, pero no es así, sus pruebas se basan en varias etapas como planificación, análisis, diseño, implementación y evaluación de calidad y también existen variedad de pruebas como prueba de caja negra y prueba de aceptación cada uno con diferentes fines y enfoques de evaluación para el software [21].

4.2.6.4.1 Pruebas unitarias

Pruebas principales se las conoce como un componente esencial en el desarrollo de software en las cuales sus pruebas son realizadas de forma individual a cada módulo existente planteada por el programador, dichas pruebas son basadas en pruebas de caja blanca probando los bloques más pequeños del software detectando errores de manera rápida, este tipo de pruebas se enfocan en la lógica interna de los componentes, dado esto se crea un tipo de prueba con modulo conductor para poder realizar la prueba de forma aislada utilizando parámetros reales [22].

4.2.6.4.2 Pruebas de aceptación

Las pruebas de aceptación se llevan a cabo como una entrega formal del software, esta prueba se realiza para determinar si el software satisface cada requisito y expectativa del usuario o cliente, en estas pruebas de software se examina desde la perspectiva del usuario final, en la cual se asegura que se cumplan las expectativas en cuanto a requisitos funcionales y no funcionales [24]. La manera más frecuente de realizar estas pruebas es:

- **Pruebas de aceptación de usuario:** Se basa en que los propios usuarios realizan pruebas que validan las funcionalidades del sistema evaluando la alineación con las necesidades y procesos comerciales.
- **Pruebas de aceptación operativa:** Se basa en que este tipo de prueba es realizada por programadores, quienes realizan pruebas que están relacionadas con restauración, instalación, migración, actualización mantenimiento o gestión de usuarios.

- **Pruebas de aceptación contractual:** Se basa en realizar las pruebas mediante criterios de aceptación establecidos en el contrato, ya que se ejecutarán comúnmente por los mismos usuarios o evaluadores. [25]

4.2.7 Modelo de Aceptación Tecnológica (TAM).

Modelo de aceptación de tecnología se la conoce como una estructura investigativa la que está diseñada para explicar el comportamiento de los usuarios ya que así demuestran la aceptación o rechazo hacia el software ya que se establece como base la estimación de utilidad y percepción de facilidad en su uso. Este modelo de aceptación tecnológica dice que las interacciones y acciones del usuario están interconectadas y son cruciales para poder aceptar o rechazar el software, la intención de esto es que el usuario emplee una tecnología la cual determine su percepción en base a dicha tecnología [26].

4.2.7.1 Utilidad percibida

La utilidad percibida se la denomina a la percepción de que una tecnología de la información contribuya al mejoramiento de productividad y facilidad en la ejecución de las tareas. Las cuales se relacionan con lo que cada persona perciba el uso del sistema en la cual su rendimiento es amplio.

4.2.7.2 Facilidad de uso percibida.

Se conoce como facilidad de uso al modelo teórico usado para la evaluación de aceptación, está en otros términos hace referencia a la percepción que tiene el usuario sobre su facilidad en cuanto a su uso, lo cual quiere decir que hasta qué punto puede facilitar y realizar el esfuerzo menos posible para que se desempeñe la tarea. Según TAM entre más fácil el usuario pueda percibir uso de esta tecnología, existe más probabilidad de usarlo de forma continua. [27]

4.2.7.3 Actitud hacia el uso

La actitud hacia el uso se conoce como la evaluación que tiene el usuario sobre la tecnología específica, la cual se influencia por dos componentes primordiales, la primera es la utilidad percibida y la segunda es la facilidad de uso percibida. De acuerdo a TAM el usuario al momento de utilizar la plataforma decide en si la tecnología es útil y fácil de usar, la actitud hacia el uso de la plataforma influirá positiva y también significativamente [28]

4.2.7.4 Intensión de uso

Se conoce a la intención de uso como la disposición del usuario para usar la tecnología planteada, lo cual nos dice que entre mayor sea la intención del usuario existe más probabilidad de que utilice la tecnología propuesta, ya que esta se influencia por dos actores principales la

cual es la utilidad percibida y la facilidad de uso percibida, las mismas que influyen an la actitud del usuario.[27]

4.2.7.5 Escala de Likert

La escala de Likert es aquella que se utiliza para lograr determinar las variables cualitativas, la cual ha sido implementada en diversidad de estudios, donde contiene indicadores estadísticos específicos brindando confiabilidad en sus resultados, debido a que sus respuestas van desde un acuerdo a un desacuerdo y sabe constar de 5 a 7 opciones, las mismas que pueden sumarse para brindar un resultado total, la cual indica la posición del tema encuestado. [29].

Tabla 1. Escala de likert

Respuesta	Valor asignado
Totalmente en desacuerdo	0
En desacuerdo	1
Ni de acuerdo, ni en desacuerdo	2
De acuerdo	3
Totalmente de acuerdo	4

Para calcular los constructos en base a la escala de Likert se plantea un cuestionario de evaluación la cual se basa en el tema tratado, ya obtenido los temas tratados y ya obtenidos mediante la encuesta como se demuestra en la **Tabla 2** la cual se tiene la tabulación análisis de los datos tomados y asignados por la escala de Likert para medir el nivel de aceptación de los encuestados [26].

Tabla 2. Ejemplos resultados de evaluación TAM

Factor	Totalmente en desacuerdo (%)	En desacuerdo (%)	Ni de acuerdo, ni en desacuerdo (%)	De acuerdo (%)	Totalmente de acuerdo (%)
UE1	-	-	-	-	-
UE2	-	-	-	-	-
UE3	-	-	-	-	-
UE4	-	-	-	-	-

4.2.7.6 Arquitectura 4+1

El modelo arquitectónico 4+1 surgió como una respuesta a los desafíos que se enfrentan las arquitecturas de software que se enfocan exclusivamente en una única perspectiva. Esta

visión propone una solución dividiendo la arquitectura del software en cinco vistas, cada una diseñada para abordar aspectos específicos [33].

- Vista lógica: También conocida como vista estructural, adopta la perspectiva del usuario final, se centra en los requerimientos (funcionales y no funcionales) que el sistema debe ofrecer a los mismos. Para conseguir esto, considera el dominio del problema y lo segmenta en clases y objetos.
- Vista de procesos: Dedicada en los requisitos no funcionales, esta vista representa el comportamiento dinámico y flujo de trabajo del software, se representa mediante el diagrama de procesos, mostrando como interactúan las partes del sistema para cumplir su objetivo.
- Vista de desarrollo: Concebida desde la visión del desarrollador, esta vista describe como se organizan los componentes del software dentro del entorno de desarrollo, representa bibliotecas, módulos y otros componentes, se representa a mediante el diagrama de componentes o paquetes.
- Vista física: De igual manera conocida como vista de despliegue, se centra en los aspectos físicos del sistema, desde conexiones hasta infraestructura para que el sistema funcione en su totalidad, se representa a mediante el diagrama de despliegue.
- Vista de escenarios: En esta vista concuerdan las cuatro anteriores vistas para formar escenarios que detalles de funcionalidades clave o generales del software, estos escenarios representan una abstracción de los requisitos más relevantes, se representa a mediante el diagrama de casos de uso [33].

4.3 Trabajos relacionados

Tabla 3. Trabajos relacionados.

Trabajos relacionados	Resumen
Desarrollo de una aplicación web interactiva para el aprendizaje de conceptos básicos de la asignatura de programación para estudiantes de primer ciclo de la carrera de Ciencias de la Computación [30].	La educación superior tiene el desafío de adaptar los métodos de aprendizaje a las nuevas tecnologías, las generaciones actuales optan por nuevas formas de aprendizaje como son: webs y apps. El presente Trabajo de Integración Curricular se enfocó en diseñar e implementar una aplicación web interactiva para el aprendizaje de conceptos básicos de la asignatura de programación para estudiantes de primer ciclo de la carrera de Ciencias de la Computación.
Desarrollo de un Chatbot como soporte al proceso de aprendizaje de la programación en la UPS [31].	Optimizar el aprendizaje es un desafío continuo a todo nivel en toda institución educativa, donde las tecnologías son aprovechadas para interactuar mejor con los estudiantes. El objetivo es evaluar los beneficios de un chatbot como soporte virtual para el proceso de aprendizaje de las estructuras de control en la materia programación en la carrera de computación de la UPS a través de Google Dialogflow. El

presente trabajo de investigación es exploratorio para identificar y definir el problema, con enfoque cualitativo para determinar las intenciones del chatbot; se utiliza la técnica de la observación y encuesta; la metodología es cuasi-experimental que está encaminada a determinar la utilidad de un asistente virtual que gestione y complemente el conocimiento de los estudiantes. Los resultados obtenidos son: definición de intenciones, eventos, frases de entrenamiento y posibles respuestas, además de una evaluación y beneficios del chatbot.

Aprendizaje de programación usando inteligencia artificial: Obtención de retroalimentación del código fuente de un programa escrito con lenguaje C mediante inteligencia artificial [32].

Este proyecto de integración curricular desarrolla una prueba de concepto para obtener retroalimentación automática de programas en C mediante Inteligencia Artificial Generativa, para ayudar a estudiantes novatos en su aprendizaje de programación. La metodología utilizada incluyó una revisión de propuestas existentes, análisis de requisitos, diseño de artefactos de software, implementación y validación de la prueba de concepto. Se utilizó Kanban para organizar las tareas llevadas a cabo en estas fases. La prueba de concepto se diseñó considerando los requisitos del Grupo de Investigación en Ingeniería de Sistemas de Tecnologías de Información (GI-INTEC) de la Escuela Politécnica Nacional y los hallazgos de la revisión de artículos relacionados relevantes. Se implementó una arquitectura que incluye una API Gateway, un servidor de retroalimentación y el consumo de los servicios de IA de OpenAI.

5 Metodología

En este apartado se presenta la metodología utilizada para llevar a cabo la creación y ejecución de este trabajo de titulación. Para ello en la sección 5.1 se señala el área de estudio en donde se realizó el TIC. La sección 5.2 aborda el procedimiento para ejecutar los dos objetivos planteados. Asimismo, la sección 5.3 muestra los recursos necesarios para el desarrollo del presente trabajo. Y para finalizar la sección 5.4 detalla los participantes en el TIC.

5.1 Área de Estudio.

El presente trabajo de titulación se llevó a cabo en los laboratorios de software de la carrera de Computación, perteneciente a la Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables (FEIRNNR) de la Universidad Nacional de Loja, ubicada en la ciudad de Loja, Ecuador. Estos laboratorios son espacios diseñados para fomentar la enseñanza y el aprendizaje de disciplinas relacionadas con la programación. En este contexto, la asignatura "Teoría de la Programación", impartida en el primer ciclo, utiliza estos laboratorios para enseñar los fundamentos de la programación en lenguaje C, constituyéndose en una parte fundamental de la formación académica ofrecida por la carrera.

5.2 Procedimiento

En el siguiente punto se presenta el procedimiento establecido para ejecutar los objetivos planteados en el TIC.

5.2.1 Objetivo Específico 1. Construir un software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante la metodología XP.

Para la ejecución del objetivo específico uno se tomó como base la metodología de desarrollo eXtreme Programming (XP), la cual consta de cinco fases: planificación, diseño, codificación, pruebas y lanzamiento. La metodología de desarrollo ágil seleccionada desempeñó un papel clave en el desarrollo del primer objetivo, (véase sección **Resultados, apartado 6.1**). Al tratarse de una metodología ágil, su naturaleza iterativa e incremental permitió dividir el desarrollo del sistema en iteraciones. Cada iteración abarcó las fases definidas en la metodología XP, tal como se detalla en el plan de iteración (véase sección **Resultados, apartado 6.1.1.4**). A continuación, se presenta un desglose del procedimiento seguido en cada una de las fases de esta metodología.

5.2.1.1 Planificación

Se llevó a cabo un conjunto de entrevistas a docentes de la carrera de Computación encargados de impartir asignaturas relacionadas al área de programación (véase sección **Metodología, apartado 5.3.2.4**). Estas entrevistas permitieron identificar los actores clave involucrados en este proyecto TIC (véase sección **Resultados, apartado 6.1**), lo que a su vez posibilitó la identificación de requisitos esenciales para delimitar el alcance de este TIC (véase sección **Resultados, apartado 6.1.1.2**), y la creación de historias de usuario en base a estos requisitos esenciales (véase sección **Resultados, apartado 6.1.1.3**). Con esta información, se elaboró el plan de iteración (véase sección **Resultados, apartado 6.1.1.4**).

5.2.1.2 Diseño

Se analizó las historias de usuario del sistema, lo que permitió seleccionar la arquitectura 4+1 como modelo principal para el presente TIC, (véase sección **Resultados, apartado 6.1.2.1**). En este modelo, se utilizaron los siguientes diagramas: Diagrama de dominio, casos de uso, clases y diagrama de despliegue. Además, se diseñaron los prototipos correspondientes en cada iteración del plan de iteración, (véase sección **Resultados, apartado 6.1.2.2**), con el fin de representar gráficamente los criterios de aceptación que fueron planteados a los usuarios finales.

5.2.1.3 Codificación

El código se desarrolló conforme a cada una de las iteraciones definidas en el plan de iteración (véase sección **Resultados, apartado 6.1.1.4**), permitiendo validar los criterios de aceptación de cada historia de usuario. Este enfoque permitió garantizar que el desarrollo siempre estuviera alineado con el objetivo del proyecto y lo que realmente necesitaban los usuarios finales.

5.2.1.4 Pruebas

Se desarrolló pruebas unitarias en todas las iteraciones del plan (véase sección **Resultados, apartado 6.1.4.16.1.1.4**), para validar el correcto funcionamiento del sistema. Además, se realizaron pruebas de aceptación en la que participaron los clientes del software de apoyo, esto mediante una reunión con cada uno de ellos (véase sección **Resultados, apartado 6.1.4.2**), donde se pudo evidenciar el correcto funcionamiento y criterios de aceptación que los mismos clientes plantearon, donde para finalizar se procedió a la firma de dichas pruebas, donde se detalla y valida que el sistema se encuentra finalizado de acuerdo a su alcance (véase sección **Resultados, apartado 6.1.4**).

5.2.1.5 Lanzamiento

Se desplegó el sistema funcional y validado por los clientes en los servidores de la carrera de Computación en el entorno de producción, permitiendo a los estudiantes acceder al software de apoyo tanto desde el laboratorio de Computación como desde cualquier ubicación con conexión a internet, además se elaboró el conjunto de manuales de usuario necesarios para el producto final, las cuales abarcaron: manual de usuario para estudiante (véase sección **Resultados, apartado 6.1.5.1**), manual de usuario para docente (véase sección **Resultados, apartado 6.1.5.2**), y manual de usuario para administrador (véase sección **Resultados, apartado 6.1.5.3**).

5.2.2 Objetivo Específico 2. Determinar la aceptación tecnológica del software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante el modelo TAM

Para la ejecución del objetivo específico dos se tomó como base el modelo de aceptación tecnológica TAM, la cual consta de tres fases: fase de elaboración, fase de medición, fase de análisis. A continuación, se describe en detalle el proceso que se llevó a cabo en cada fase.

5.2.2.1 Fase de elaboración

Se elaboró un cuestionario para determinar la percepción del usuario del software de apoyo teniendo en cuenta los 4 constructos que son: Utilidad Percibida (UP), Facilidad de Uso Percibida (FUP), Actitud hacia el Uso (AU) y percepción de Intención de Uso (IU), donde para cada pregunta se le definió un ID, para saber a qué constructo forma parte, para más información (véase sección **Resultados, apartado 6.2.1**), se realizó una encuesta usando la herramienta Google Forms, Esta encuesta se dirigió a una muestra de 55 estudiantes, la cual será la representación del 100% de la población, la cual está compuesta por estudiantes y docentes de la carrera de Computación que hacen uso del laboratorio de software, esta muestra se debe a que a no resulta viable evaluar a todos los estudiantes y docentes de la carrera que usarán el software de apoyo, pero la muestra seleccionada son una gran parte de los usuarios que utilizaran el sistema.

5.2.2.2 Fase de medición

En base a la fase anterior se elaboró una tabla que recopila la información de manera organizada por cada constructo, lo que facilita un análisis más detallado de cada variable y sus respectivas preguntas, es por ello que se realizó la tabulación en base a la escala de Likert lo

que permitió ponderar las respuestas en una escala de 0 a 4, para más información (véase sección **Resultados, apartado 6.2.2**).

5.2.2.3 Análisis de resultados.

Se analizó la matriz de resultados obtenida en la anterior fase. Este análisis se llevó a cabo por constructo, lo que permitió analizar cada pregunta dentro de dicho constructo. Mediante la escala de Likert se pudo observar el nivel de aceptación de cada pregunta, el análisis detallado por pregunta permitió sustentar una conclusión final basada en el nivel de aceptación identificado, para más información (véase sección **Resultados, apartado 6.2.3**).

5.3 Recursos

Con la finalidad de ejecutar las fases anteriormente nombradas y cumplir con los dos objetivos específicos del presente TIC, se emplearon los siguientes recursos.

5.3.1 Recursos científicos.

5.3.1.1 Investigación bibliográfica

Esta técnica permitió buscar información bibliográfica existente relacionada al presente TIC, donde se buscó información en libros, revistas científicas, trabajos de titulación y artículos científicos, esto con el fin de sustentar la información del marco teórico (**véase sección Marco Teórico, 4**).

5.3.1.2 Experimento

Se llevó a cabo un proceso de experimentación, el cual detalla las actividades a ejecutarse para poder evaluar el software de apoyo, dicho proceso, se empleó de forma específica en el segundo objetivo, el cual abarca la evaluación de la aceptación tecnológica del software mediante TAM, esto, por parte de todos los partes usuarios (**véase sección Metodología, apartado 5.2.2**).

5.3.2 Recursos técnicos

5.3.2.1 Metodología XP

Se empleó la metodología XP a lo largo de este TIC para alcanzar el cumplimiento del primer objetivo, siguiendo el desarrollo de cada una de sus fases (**véase sección Metodología, apartado 5.2.1**).

5.3.2.2 Metodología TAM

Se empleó la metodología TAM a lo largo de este TIC para alcanzar el cumplimiento del segundo objetivo, siguiendo el desarrollo de cada una de sus fases (**véase sección Metodología, apartado 5.2.2**).

5.3.2.3 Muestreo por conveniencia

Se empleó la técnica de muestreo por conveniencia para determinar una población accesible dentro de la carrera de Computación en la Universidad Nacional de Loja. Esta decisión se basó en la inviabilidad de evaluar a toda la población de la carrera, que cuenta con más de 200 estudiantes, debido a restricciones de tiempo y recursos. Por ello, se seleccionaron los cursos Primero “A” y Primero “B” del primer ciclo, los cuales reúnen a un total de 55 estudiantes. Estos cursos fueron elegidos debido a su relación directa con la materia " Teoría de la Programación", centrada en la enseñanza de programación en C, que constituye el núcleo del sistema desarrollado.

La selección también consideró la practicidad y accesibilidad de los estudiantes y docentes, ya que las actividades de evaluación se realizaron durante horas curriculares en los laboratorios de software de la universidad. Esto facilitó la demostración del sistema y la aplicación de encuestas basadas en el Modelo de Aceptación Tecnológica (TAM). Este enfoque permitió analizar el nivel de aceptación tecnológica del software en un entorno académico controlado y representativo de los usuarios finales (véase sección **Metodología, apartado 5.3.2.5**).

5.3.2.4 Entrevista

Se empleó entrevistas como técnica inicial de recolección de información, lo que posibilitó al autor de este TIC extraer las historias de usuario en base a estos datos (véase sección **Resultados, apartado 6.1.1.3**), los requisitos esenciales del sistema (véase sección de **Resultados, apartado 6.1.1.2**). Esto, mediante las entrevistas realizadas, a los docentes encargados de impartir asignaturas relacionadas al área de programación, donde se realizó tres entrevistas, la primera fue realizada al Ing. Wilman Chamba, docente de la materia “Teoría de la programación” (véase sección **Anexo 1**), la segunda entrevista se realizó al docente de la materia de “Programación Orientada a Objetos”, Ing. Edison Coronel ” (véase sección **Anexo 1**), y la tercera entrevista a la Ing. María Ruilova, docente de la materia “Diseño de software” (véase sección **Anexo 1**).

5.3.2.5 Encuesta

Se empleó encuestas como técnica para evaluar el Modelo de Aceptación Tecnológica (TAM), entre los estudiantes que utilizaron el software de apoyo implementado (véase sección **Resultados, apartado 6.2**), con el objetivo de evaluar la aceptación tecnológica del software por parte de los usuarios. Esta encuesta fue desarrollada a través de la herramienta Google Forms.

5.4 Participantes

Los participantes involucrados en el desarrollo del presente TIC son los siguientes:

- Juan Carlos Armijos Sánchez como autor del presente TIC.
- Ing. José Oswaldo Guamán Quinche, Mg. Sc, como director del TIC, el cual guio y supervisó el desarrollo del proyecto.
- Ing. Pablo Fernando Ordoñez Ordoñez, Mg. Sc, como director de la carrera de Computación en la UNL, el cual aprobó y validó el proceso del TIC.
- Docentes del área de programación como facilitadores de información para la realización del TT.
- Estudiantes de la carrera de Computación de la Universidad Nacional de Loja como facilitadores de información para la realización del TT.

6 Resultados

En este apartado se presentan los resultados obtenidos por cada objetivo y el conjunto de tareas realizadas al cumplirlos. En la sección 6.1 se realizó el primer objetivo cumpliendo las fases de la metodología XP; en la sección 6.2 se aborda el segundo objetivo, desarrollado siguiendo el modelo de aceptación tecnológica TAM.

6.1 Objetivo 1: Construir un software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante la metodología XP.

Para dar por cumplido el primer objetivo se aplicó la metodología XP, que consta de cinco fases que son las siguientes: en la sección 6.1.1 se encuentra la fase de planificación, en la sección 6.1.2 se encuentra la fase de diseño, en la sección 6.1.3 se encuentra la fase de codificación, en la sección 6.1.4 se encuentra la fase de pruebas, y para finalizar en la sección 6.1.5 se encuentra la fase de lanzamiento.

6.1.1 Fase de planificación.

En la fase de planificación se logró identificar los diferentes actores del sistema que interactúan en el software de apoyo, (véase sección **Resultados, apartado 6.1.1.1**), de igual forma se logró identificar los requisitos esenciales del software de apoyo (véase sección **Resultados, apartado 6.1.1.2**), y sus respectivas historias de usuario (véase sección **Resultados, apartado 6.1.1.3**), y para finalizar con esta fase se presenta el plan de iteración a seguir a lo largo del proyecto, (véase sección **Resultados, apartado 6.1.1.4**).

6.1.1.1 Actores del sistema

En cuanto a los usuarios que interactúan con el sistema, se identificaron tres tipos: docente, estudiante y administrador. A continuación, se presenta la **Tabla 4** donde se presenta de forma más detallada las funciones que pueden realizar los usuarios en el software de apoyo.

Tabla 4. Actores del sistema

Actor	Descripción
Administrador	<ul style="list-style-type: none">• Registrar nuevos periodos académicos• Gestionar comentarios (Registrar, Editar)• Visualizar temas destacados
Docente	<ul style="list-style-type: none">• Gestionar temas (Registrar, Editar, Activar/Desactivar, Buscar)• Gestionar subtemas (Registrar, Editar, Activar/Desactivar, Buscar)

	<ul style="list-style-type: none"> • Gestionar ejercicios (Registrar, Editar, Activar/Desactivar, Buscar) • Gestionar preguntas de control (Registrar, Editar, Activar/Desactivar, Buscar) • Gestionar comentarios (Registrar, Editar) • Visualizar temas destacados
Estudiante	<ul style="list-style-type: none"> • Visualizar contenido activo (temas, subtemas, ejercicios, preguntas de control) • Retroalimentación de ejercicios • Visualizar progreso • Responder pregunta de control • Gestionar comentarios (Registrar, Editar) • Visualizar temas destacados

6.1.1.2 Requisitos esenciales del sistema

En la **Tabla 5**, se muestra un ejemplo de 3 requisitos funcionales del sistema, para obtener el listado completo de los requisitos funcionales (véase sección **Anexo 2**).

Tabla 5. Requisitos funcionales del sistema

Código	Requisito	Descripción	Prioridad
RF001	Listar temas	El sistema permitirá al docente listar todos los temas activos e inactivos.	Media
RF002	Registrar temas	El sistema permitirá al docente crear nuevos temas al proporcionar: título, objetivos de aprendizaje, descripción y recursos adicionales.	Alta
RF003	Editar temas	El sistema permitirá al docente la capacidad de modificar información relacionada con temas, tanto activos como inactivos, lo que incluirá la posibilidad de editar el título, los objetivos de aprendizaje, descripción y recursos adicionales.	Media

Tabla 6. Requisitos no funcionales del sistema

Código	Requisito	Descripción	Prioridad
RNF01	Usabilidad	El sistema debe contar con interfaces amigables para el usuario, que permitan al usuario	Alta

		usar el software de apoyo de forma sencilla.	
RNF02	Seguridad	El sistema debe garantizar el acceso al software de apoyo mediante credenciales (correo electrónico y contraseña) de los usuarios	Alta
RNF03	Disponibilidad	El sistema debe estar disponible dentro del laboratorio de la carrera de Computación, siempre que el estado de los servidores sea optimo.	Alta
RNF04	Desarrollo	El sistema deberá ser desarrollado utilizando las siguientes tecnologías: Node Js, Express Js, React y MySQL	

6.1.1.3 Historias de usuario

En este apartado se describen los campos de las historias de usuario, el modelo utilizado para las HU en este TIC y una historia de usuario de ejemplo del software de apoyo, para obtener el listado completo de los requisitos funcionales (véase sección

Anexo 2).

6.1.1.3.1 Descripción de los campos de la historia de usuario

Los campos a tomar en cuenta dentro de la historia de usuario (HU) son los siguientes:

- **Número:** Constituye la numeración única de cada Historia de Usuario.
- **Nombre de la historia de usuario:** Consiste en una breve descripción identificativa de la HU.
- **Usuario:** Designación del individuo responsable de la actividad mencionada en la HU.
- **Interacción Asignada:** Hace referencia al número de iteración asociado a la actividad.
- **Referencia de requisito:** Se refiere al identificador del requisito al que la Historia de Usuario está vinculada.
- **Prioridad en el negocio:** Se asigna de acuerdo a la relevancia que ostenta la actividad.
- **Riesgo en el desarrollo:** Representa la complejidad de llevar a cabo la actividad de desarrollo.
- **Fecha de inicio:** La fecha de creación de la HU.
- **Fecha de entrega:** La fecha estimada de finalización de la HU.

- **Descripción:** Detalla las tareas y acciones específicas contenidas en las Historias de Usuario.
- **Observaciones:** Contempla los aspectos significativos en cada Historia de Usuarios.
- **Criterios de aceptación:** Establece las condiciones que el desarrollo debe satisfacer para que la historia de usuario pueda ser considerada como aceptada.

6.1.1.3.2 Formato de historia de usuario

El modelo utilizado para las historias de usuario es el siguiente:

Tabla 7. Modelo HU

Historia de usuario	
Número:	Nombre:
Usuario:	
Iteración asignada:	Referencia de requisito:
Prioridad en el negocio (Alta/Media/Baja):	Fecha de inicio:
Riesgo en el desarrollo (Alta/Media/Baja):	Fecha de entrega:
Descripción:	
Observaciones:	
Criterios de aceptación:	

6.1.1.3.3 Desglose de historias de usuario

En la **Tabla 8** se presenta un resumen de todas las historias de usuario donde se detalla su identificador, el nombre de cada HU y la referencia al requisito funcional al cual está vinculada. Este desglose permite una visualización clara de la información facilitando el seguimiento y comprensión de las relaciones entre las HU y los requisitos.

Tabla 8. Tabla de historias de usuario

Historias de usuario		
Identificador	Nombre	Requisito vinculado
HU001	Crear usuario	RF001
HU002	Editar usuario	RF002
HU003	Listar temas	RF003
HU004	Registrar temas	RF004
HU005	Editar temas	RF005
HU006	Historial de cambios por tema	RF006
HU007	Activar/Desactivar temas	RF007
HU008	Buscar temas	RF008
HU009	Visualizar temas	RF009
HU010	Listar subtemas	RF010
HU011	Registrar subtemas	RF011
HU012	Editar subtemas	RF012

HU013	Historial de cambios por subtema	RF013
HU014	Activar/Desactivar subtemas	RF014
HU015	Buscar subtemas	RF015
HU016	Visualizar subtemas	RF016
HU017	Listar ejercicios	RF017
HU018	Registrar ejercicios	RF018
HU019	Editar ejercicios	RF019
HU020	Historial de cambios por ejercicio	RF020
HU021	Activar/Desactivar ejercicios	RF021
HU022	Buscar ejercicios	RF022
HU023	Visualizar ejercicios	RF023
HU024	Proporcionar retroalimentación de ejercicios	RF024
HU025	Listar preguntas de control	RF025
HU026	Registrar pregunta de control	RF026
HU027	Editar pregunta de control	RF027
HU028	Historial de cambios por pregunta de control	RF028
HU029	Activar/Desactivar pregunta de control	RF029
HU030	Buscar preguntas de control	RF030
HU031	Visualizar preguntas de control	RF031
HU032	Responder pregunta de control	RF032
HU033	Listar comentarios	RF033
HU034	Registrar comentarios	RF034
HU035	Editar comentarios	RF035
HU036	Visualizar comentarios	RF036
HU037	Visualizar progreso	RF037
HU038	Cambiar periodo académico	RF038
HU039	Temas destacados	RF039
HU040	Notificaciones por email de nuevo contenido	RF040

6.1.1.4 Plan de iteración

En la **Tabla 9** se detalla el plan de iteración empleado en el desarrollo del software de apoyo. En ella se distribuyen las historias de usuario en cinco iteraciones, adicionalmente se incluye el documento completo que detalla el progreso de cada iteración, abarcando todas las fases de la metodología XP (véase sección **Anexo 5**).

Tabla 9. Plan de iteración

ID.	Historia de usuario	Fecha de creación	Fecha de entrega	Iteración
HU001	Crear usuario	20/11/2023	12/12/2024	Primera
HU002	Editar usuario			
HU003	Listar temas			
HU004	Registrar temas			
HU005	Editar temas			

HU006	Historial de cambios por tema			
HU007	Activar/Desactivar temas			
HU008	Buscar temas			
HU009	Visualizar temas			
HU010	Listar subtemas	30/11/2023	20/12/2023	Segunda
HU011	Registrar subtemas			
HU012	Editar subtemas			
HU013	Historial de cambios por subtema			
HU014	Activar/Desactivar subtemas			
HU015	Buscar subtemas			
HU016	Visualizar subtemas			
HU017	Listar ejercicios	06/12/2023	24/12/2023	Tercera
HU018	Registrar ejercicios			
HU019	Editar ejercicios			
HU020	Historial de cambios por ejercicio			
HU021	Activar/Desactivar ejercicios			
HU022	Buscar ejercicios			
HU023	Visualizar ejercicios			
HU024	Proporcionar retroalimentación de ejercicios			
HU025	Listar preguntas de control	19/12/2023	05/01/2024	Cuarta
HU026	Registrar pregunta de control			
HU027	Editar pregunta de control			
HU028	Historial de cambios por pregunta de control			
HU029	Activar/Desactivar pregunta de control			
HU030	Buscar pregunta de control			
HU031	Visualizar pregunta de control			
HU032	Responder pregunta de control			
HU033	Listar comentarios	15/01/2024	06/02/2024	Quinta
HU034	Registrar comentarios			
HU035	Editar comentarios			
HU036	Visualizar comentarios			

HU037	Visualizar progreso
HU038	Cambiar periodo académico
HU039	Temas destacados
HU040	Notificaciones por email de nuevo contenido

Para obtener una descripción más detallada de la fase de planificación realizada en cada iteración, (véase sección **Anexo 5**).

6.1.2 Fase de diseño.

En esta sección se detalla el diseño del sistema, en base a la arquitectura 4+1 (véase sección **Anexo 6**), de igual manera se presentan los prototipos del software de apoyo (véase sección **Anexo 5**).

6.1.2.1 Diagramas de acuerdo a la Arquitectura 4+1

Se realizaron los respectivos diagramas UML de acuerdo arquitectura nombrada, a continuación, se presentan los principales resultados, para ver en mejor calidad los diagramas (véase sección **Anexo 16**) para ver el anexo de la arquitectura 4+1 en mejor calidad.

6.1.2.1.1 Diagrama de dominio

En la **Figura 1**, se representa el diagrama de dominio para el software de apoyo.

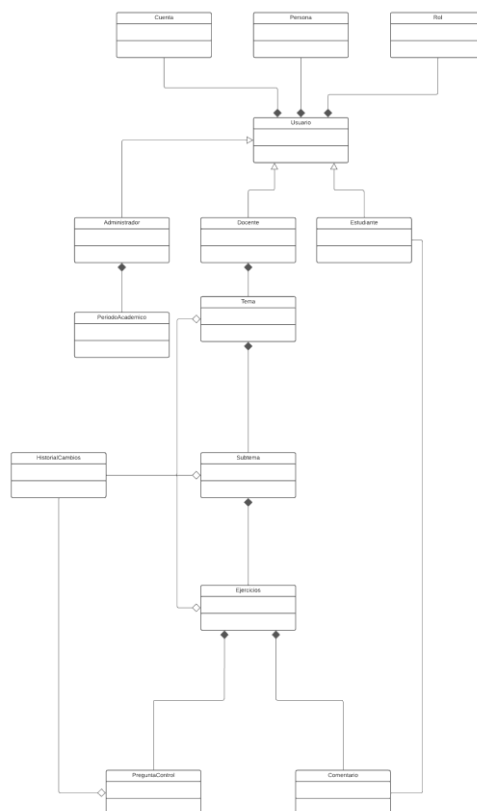


Figura 1. Diagrama de dominio. [Fuente propia].

6.1.2.1.2 Diagrama de casos de uso

En la **Figura 2**, se representa el diagrama de casos de uso para el software de apoyo.

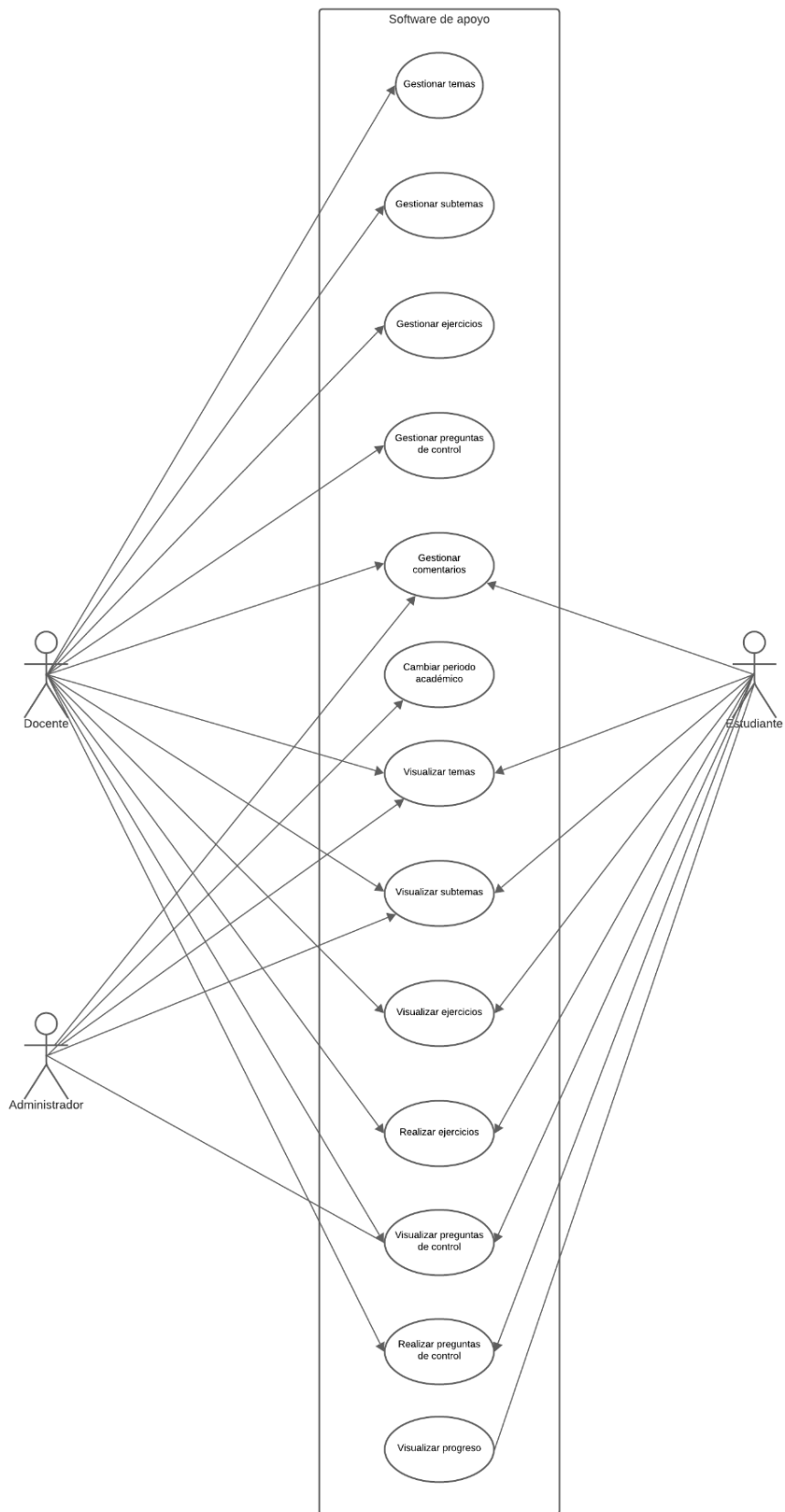


Figura 2. Diagrama de casos de uso [Fuente propia]

6.1.2.1.3 Diagrama de clases

En la **Figura 3**, se muestra el diagrama de clases el cual representa las entidades clave y sus relaciones dentro del software de apoyo.

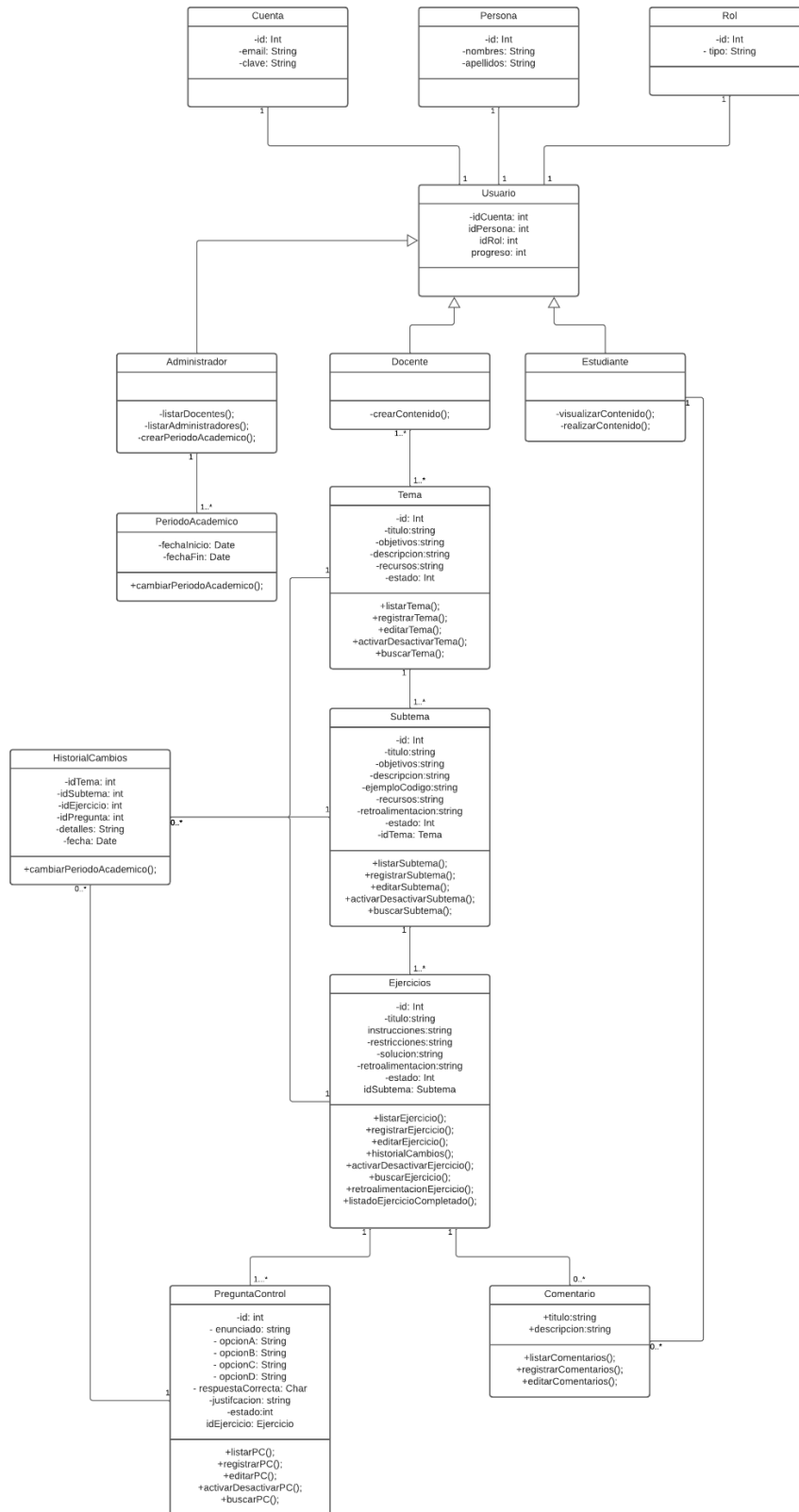


Figura 3. Diagrama de clases [Fuente propia]

6.1.2.1.4 Diagrama de despliegue

En la **Figura 4**, se representa el diagrama de casos de uso para el software de apoyo.

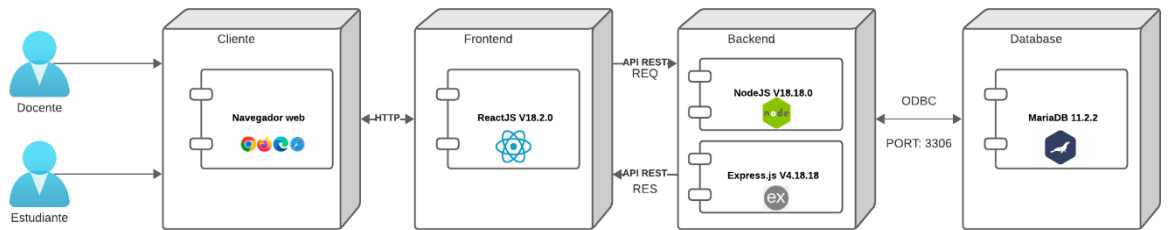


Figura 4. Diagrama de despliegue [Fuente propia]

6.1.2.2 Prototipos

En este apartado se diseñaron prototipos de interfaces de usuario, esto con el fin de obtener una idea ya definida de como codificar y que el sistema sea más visual no solo para el desarrollar al momento de crear sino también para los usuarios. A continuación, se presenta un prototipo de ejemplo en la **Figura 5**, estos prototipos hacen parte del plan de iteración (véase sección **Anexo 5**).

Temas

Temas inactivos de rojo

Q Buscar tema

Tema 1
Tema 2
Tema 3
Tema 4

Crear Editor Activar/Desactivar Historial de cambios

Figura 5. Prototipo de ejemplo [Fuente propia]

6.1.3 Fase de codificación.

En la fase de codificación se realizó la implementación de todas las historias de usuario planteadas en la fase de planificación (véase sección **Resultados, Fase planificación 6.1.1**), durante este proceso, en la fase de diseño (véase sección **Resultados, Fase de diseño 6.1.2**) se implementó una segmentación del software en dos entornos distintos: el backend y el frontend.

La **Figura 6** detalla la estructura del servidor backend, ofreciendo una representación de la organización y las relaciones entre sus diversos componentes.

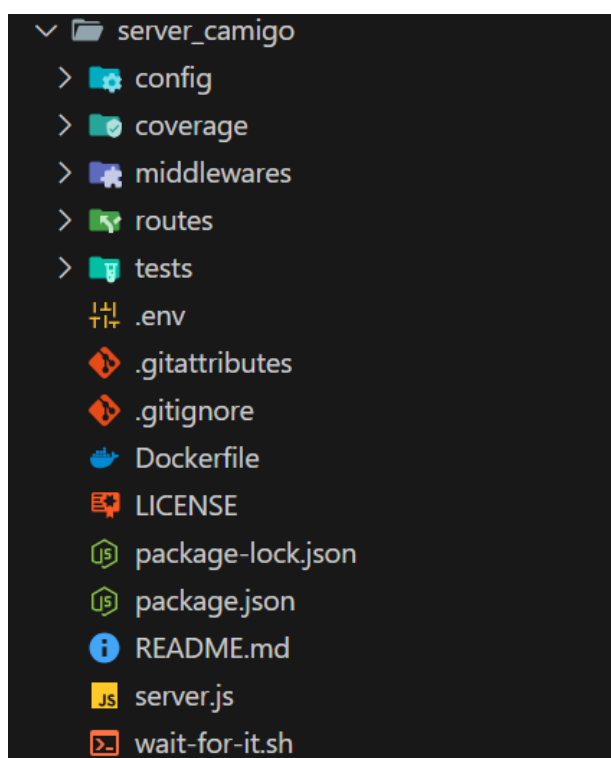


Figura 6. Estructura del ambiente backend [Fuente propia].

Para el desarrollo del frontend, se optó por la utilización de la biblioteca React, que facilitó la creación de interfaces interactivas. Se utilizaron APIs para la comunicación con el backend, los componentes modulares y los contextos aseguraron una interfaz coherente y funcional. Además, mediante funciones que manejaron la lógica del software de apoyo, se logró una implementación eficiente. La **Figura 7** detalla la estructura del servidor frontend, ofreciendo una representación de la organización y las relaciones entre sus componentes.

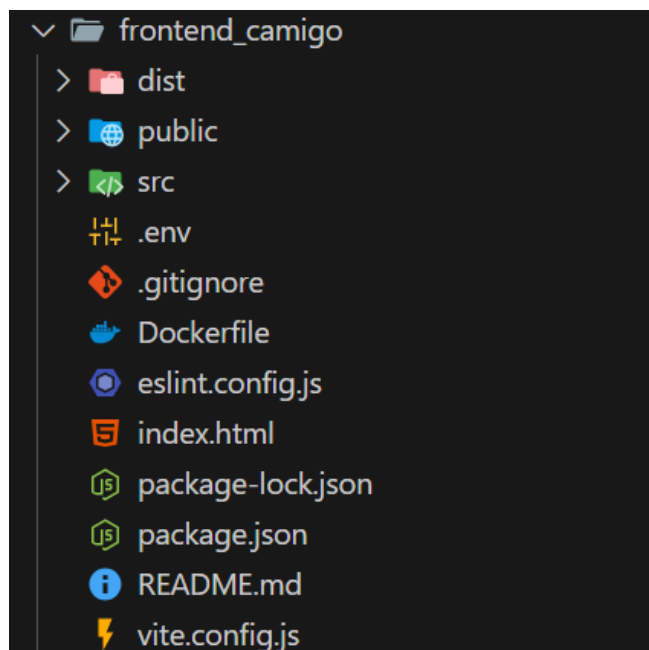


Figura 7. Estructura del ambiente frontend [Fuente propia].

6.1.4 Fase de pruebas.

En la fase de pruebas, se llevaron a cabo **Pruebas unitarias** para validar el correcto funcionamiento del software de apoyo. Y de igual manera, se presentan las **Pruebas de aceptación** que permiten validar que el sistema cumple con el alcance definido.

6.1.4.1 Pruebas unitarias

La **Tabla 10** presenta el listado de las pruebas unitarias realizadas para validar el correcto funcionamiento del software de apoyo (véase sección **Anexo 5**).

Tabla 10. Resumen general de las pruebas unitarias realizadas

ID	Descripción	Estado
PU001	Registrar un nuevo estudiante	Aprobado
PU002	Registrar un nuevo docente	Aprobado
PU003	Registrar un nuevo administrador	Aprobado
PU004	Registro con un email duplicado	Aprobado
PU005	Editar usuario con todos sus campos incluida la clave	Aprobado
PU006	Editar usuario con todos sus campos sin incluir clave	Aprobado
PU007	Edición de usuario inexistente	Aprobado
PU008	Listar temas activos	Aprobado
PU009	Listar temas cuando existe ningún tema	Aprobado
PU010	Listar temas activos e inactivos.	Aprobado
PU011	Registrar un nuevo tema	Aprobado
PU012	Editar un tema existente	Aprobado
PU013	Editar un tema inexistente.	Aprobado
PU014	Registrar un nuevo cambio en tema	Aprobado
PU015	Activar/Desactivar tema	Aprobado
PU016	Buscar contenido con término existente	Aprobado
PU017	Visualizar temas	Aprobado

PU018	Listar subtemas activos	Aprobado
PU019	Listar subtemas cuando existe ningún tema	Aprobado
PU020	Listar subtemas activos e inactivos.	Aprobado
PU021	Registrar un nuevo subtema	Aprobado
PU022	Editar un subtema existente	Aprobado
PU023	Editar un subtema inexistente.	Aprobado
PU024	Registrar un nuevo cambio en subtema	Aprobado
PU025	Activar/Desactivar subtema	Aprobado
PU026	Buscar contenido con término existente	Aprobado
PU027	Visualizar subtemas	Aprobado
PU028	Listar ejercicios activos	Aprobado
PU029	Listar ejercicios cuando existe ningún subtema	Aprobado
PU030	Listar ejercicios activos e inactivos.	Aprobado
PU031	Registrar un nuevo ejercicio	Aprobado
PU032	Editar un ejercicio existente	Aprobado
PU033	Editar un ejercicio inexistente.	Aprobado
PU034	Registrar un nuevo cambio en ejercicio	Aprobado
PU035	Activar/Desactivar ejercicio	Aprobado
PU036	Buscar contenido con término existente	Aprobado
PU037	Visualizar ejercicios	Aprobado
PU038	Mostrar la retroalimentación del ejercicio	Aprobado
PU039	Listar preguntas activas	Aprobado
PU040	Listar ejercicios cuando existe ningún ejercicio.	Aprobado
PU041	Listar preguntas activas e inactivas.	Aprobado
PU042	Registrar una nueva pregunta	Aprobado
PU043	Editar una pregunta existente	Aprobado
PU044	Editar una pregunta inexistente.	Aprobado
PU045	Registrar un nuevo cambio en pregunta	Aprobado
PU046	Activar/Desactivar pregunta	Aprobado
PU047	Buscar contenido con término existente	Aprobado
PU048	Visualizar preguntas	Aprobado
PU049	Responder pregunta de control	Aprobado
PU050	Listar comentarios existentes	Aprobado
PU051	Listar comentarios cuando no existe ningún ejercicio.	Aprobado
PU052	Registrar un nuevo comentario	Aprobado
PU053	Editar un comentario existente	Aprobado
PU054	Editar un comentario inexistente.	Aprobado
PU055	Editar un comentario con datos incompletos	Aprobado
PU056	Visualizar comentarios	Aprobado
PU057	Visualizar progreso de un usuario existente	Aprobado
PU058	Visualizar progreso de un usuario inexistente	Aprobado
PU059	Obtener progreso sin proporcionar información del usuario	Aprobado
PU060	Registrar un nuevo periodo académico exitosamente	Aprobado
PU061	Obtener valoración existente de los temas	Aprobado
PU062	Registrar nuevo contenido y enviar notificaciones por email.	Aprobado

6.1.4.2 Pruebas de aceptación

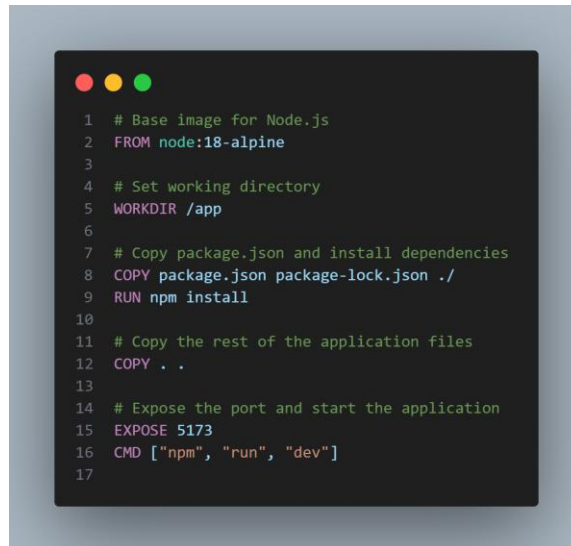
La **Tabla 11** presenta el listado de las pruebas de aceptación realizadas a los usuarios finales del software de apoyo (véase secciones **Anexo 11**, **Anexo 12**, **Anexo 13**, **Anexo 14**).

Tabla 11. Resumen general de las pruebas de aceptación realizadas

HU #	Historia de usuario	Estado
HU001	Crear usuario	Aprobado
HU002	Editar usuario	Aprobado
HU003	Listar temas	Aprobado
HU004	Registrar temas	Aprobado
HU005	Editar temas	Aprobado
HU006	Historial de cambios por tema	Aprobado
HU007	Activar/Desactivar temas	Aprobado
HU008	Buscar temas	Aprobado
HU009	Visualizar temas	Aprobado
HU010	Listar subtemas	Aprobado
HU011	Registrar subtemas	Aprobado
HU012	Editar subtemas	Aprobado
HU013	Historial de cambios por subtema	Aprobado
HU014	Activar/Desactivar subtemas	Aprobado
HU015	Buscar subtemas	Aprobado
HU016	Visualizar subtemas	Aprobado
HU017	Listar ejercicios	Aprobado
HU018	Registrar ejercicios	Aprobado
HU019	Editar ejercicios	Aprobado
HU020	Historial de cambios por ejercicio	Aprobado
HU021	Activar/Desactivar ejercicios	Aprobado
HU022	Buscar ejercicios	Aprobado
HU023	Visualizar ejercicios	Aprobado
HU024	Proporcionar retroalimentación de ejercicios	Aprobado
HU025	Listar preguntas de control	Aprobado
HU026	Registrar pregunta de control	Aprobado
HU027	Editar pregunta de control	Aprobado
HU028	Historial de cambios por pregunta de control	Aprobado
HU029	Activar/Desactivar pregunta de control	Aprobado
HU030	Buscar preguntas de control	Aprobado
HU031	Visualizar preguntas de control	Aprobado
HU032	Responder pregunta de control	Aprobado
HU033	Listar comentarios	Aprobado
HU034	Registrar comentarios	Aprobado
HU035	Editar comentarios	Aprobado
HU036	Visualizar comentarios	Aprobado
HU037	Visualizar progreso	Aprobado
HU038	Cambiar periodo académico	Aprobado
HU039	Temas destacados	Aprobado
HU040	Notificaciones por email de nuevo contenido	Aprobado

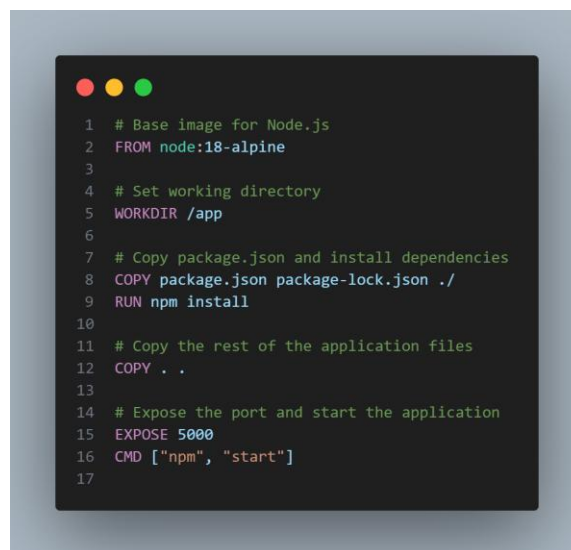
6.1.5 Fase de lanzamiento.

En la fase de lanzamiento se realizó el despliegue del software de apoyo usando la herramienta Docker para tener el sistema en contenedores y que sea más sencilla su implementación en los servidores en la **Figura 8** se presenta el archivo de configuración de Docker para el frontend y en la **Figura 9** el archivo de configuración del backend.



```
1 # Base image for Node.js
2 FROM node:18-alpine
3
4 # Set working directory
5 WORKDIR /app
6
7 # Copy package.json and install dependencies
8 COPY package.json package-lock.json ./
9 RUN npm install
10
11 # Copy the rest of the application files
12 COPY . .
13
14 # Expose the port and start the application
15 EXPOSE 5173
16 CMD ["npm", "run", "dev"]
17
```

Figura 8. Archivo de configuración Docker del Frontend.



```
1 # Base image for Node.js
2 FROM node:18-alpine
3
4 # Set working directory
5 WORKDIR /app
6
7 # Copy package.json and install dependencies
8 COPY package.json package-lock.json ./
9 RUN npm install
10
11 # Copy the rest of the application files
12 COPY . .
13
14 # Expose the port and start the application
15 EXPOSE 5000
16 CMD ["npm", "start"]
17
```

Figura 9. Archivo de configuración Docker del Backend.

Para vincular y ejecutar los servicios se usó Docker Compose, el archivo donde se configuración se presenta en la **Figura 10**:

```
1 version: '3.8'
2 services:
3   backend:
4     build:
5       context: ./server_camigo
6       dockerfile: Dockerfile
7     ports:
8       - "5000:5000"
9     environment:
10      - HOST=mysql
11      - PORT=3306
12      - USER=root
13      - PASSWORD=root
14      - BD=camigo
15     depends_on:
16       mysql:
17         condition: service_healthy
18         restart: always
19
20   mysql:
21     image: mysql:8.0
22     container_name: mysql
23     ports:
24       - "3307:3306"
25     environment:
26       MYSQL_ROOT_PASSWORD: root
27       MYSQL_DATABASE: camigo
28     volumes:
29       - mysql_data:/var/lib/mysql
30       - ./server_camigo/config/schema.sql:/docker-entrypoint-initdb.d/schema.sql
31     healthcheck:
32       test: ["CMD-SHELL", "mysql -u root -proot -e 'SELECT 1'"]
33       interval: 40s
34       timeout: 20s
35       retries: 10
36     restart: always
37
38 volumes:
39   mysql_data:
```

Figura 10. Archivo Docker Compose de los servicios del software de apoyo.

El software de apoyo fue instalado en los servidores de la carrera de computación utilizando CentOS 9 como sistema operativo. Los pasos principales para el despliegue incluyeron lo siguiente:

- **Conexión al servidor:** Para acceder al servidor se utilizó SSH mediante el cliente PuTTY, configurando el puerto 9988 y autenticándose en el sistema. En la Figura 11 se muestra la conexión establecida al servidor mediante PuTTY.

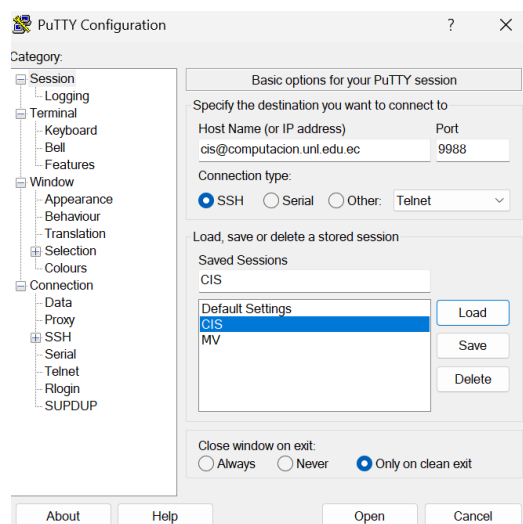


Figura 11. Conexión al servidor mediante SSH.

```

cis@localhost:~
Using username "cis".
cis@computacion.unl.edu.ec's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Dec 4 10:35:59 2024 from 10.20.200.189
[cis@localhost ~]$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:c6:51:09:1bc txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens18: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.50.85 netmask 255.255.255.0 broadcast 192.168.50.255
    inet6 fe80::be24:11ff:feb3:5f54 prefixlen 64 scopeid 0x20<link>
    ether bc:24:11:b3:5f:54 txqueuelen 1000 (Ethernet)
    RX packets 2319 bytes 524020 (511.7 KiB)
    RX errors 0 dropped 2 overruns 0 frame 0
    TX packets 522 bytes 47237 (46.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 10 bytes 1564 (1.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 1564 (1.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Figura 12. Inicio de sesión dentro de la MV con Centos 9

- **Instalación de Docker y Docker Compose:** Se instalaron Docker y Docker Compose en el servidor utilizando los comandos necesarios para la configuración de un entorno de contenedores.
- **Transferencia de la aplicación:** La aplicación fue transferida al servidor utilizando el comando scp, asegurando su ubicación en el directorio /opt/aplicaciones_docker. En la **Figura 13** se presenta la ejecución del comando scp para la transferencia de archivos desde el equipo local al servidor.

```

[cis@localhost ~]$ scp -P 9988 -r "C:\Users\juanc\Documents\camigo_mv" cis@computacion.unl.edu.ec:/opt/aplicaciones_docker

```

Figura 13. Transferencia de la aplicación al servidor.

- **Levantamiento de los servicios:** Se utilizó el comando docker compose up --build para levantar los servicios del sistema. Este proceso permitió crear y ejecutar los contenedores del backend y base de datos de manera eficiente. En la **Figura 14** se muestra la ejecución de este comando y la lista de imágenes, obtenida con el comando docker images.

```

[cis@localhost ~]$ docker images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
aplicacion_camigo-backend  latest      cfd2df4d701a  4 days ago    165MB
<none>              <none>     e4d8c366fa74  4 days ago    165MB
mysql                8.0         6c55ddbef969  3 months ago  591MB

```

Figura 14. Levantamiento de servicios y verificación de contenedores activos.

- **Configuración de los puertos:** Para garantizar la correcta exposición de los servicios, se configuraron los puertos del backend (5000/TCP) y base de datos (3306/TCP) mediante el uso del comando firewall-cmd. En la **Figura 15** se observan el comando docker ps, que muestra la información completa de los contenedores en conjunto con sus puertos ya configurados.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
63737cf4b96a	aplicacion_camigo-backend	"docker-entrypoint.s..."	4 days ago	Up 8 hours	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
e5be76ca9a41	mysql:8.0	"docker-entrypoint.s..."	4 days ago	Up 8 hours (healthy)	33060/tcp, 0.0.0.0:3307->3306/tcp, [::]:3307->3306/tcp

Figura 15. Configuración de los puertos en el servidor.

- **Configuración de Apache para el frontend:** Se instaló Apache directamente en el servidor CentOS 9 para servir los archivos estáticos del frontend. Esto permitió que las solicitudes HTTP realizadas por los usuarios fueran gestionadas eficientemente por el servidor web, proporcionando los archivos necesarios para el funcionamiento de la interfaz gráfica. Se configuró un archivo en `/etc/httpd/conf.d/appsdk.conf` que definió las reglas para la dirección raíz de la aplicación y el manejo de peticiones esto se muestra en la **Figura 16**.

```
[cis@localhost ~]$ cat /etc/httpd/conf.d/appsdk.conf
<VirtualHost *:80>
    ServerName computacion.unl.edu.ec
    ProxyPreserveHost On
    DocumentRoot "/opt/aplicaciones_dockerizadas/aplicacion_camigo/frontend_camigo/dist"

    <Directory "/opt/aplicaciones_dockerizadas/aplicacion_camigo/frontend_camigo/dist">
        AllowOverride All
        Require all granted
        Options -Indexes +FollowSymLinks
    </Directory>

    Alias /appsdk "/opt/aplicaciones_dockerizadas/aplicacion_camigo/frontend_camigo/dist"

    <Directory "/opt/aplicaciones_dockerizadas/aplicacion_camigo/frontend_camigo/dist">
        AllowOverride All
        Require all granted
    </Directory>

    ProxyPass /api/ http://localhost:5000/ retry=1 acquire=3000 timeout=600 keepalive=On
    ProxyPassReverse /api/ http://localhost:5000/

    ErrorLog "/var/log/httpd/appsdk_error.log"
    CustomLog "/var/log/httpd/appsdk_access.log" combined
    RequestHeader set Host computacion.unl.edu.ec
</VirtualHost>
```

Figura 16. Configuración de Apache para servir el frontend.

- **Automatización del servicio:** Se creó un archivo de configuración en `/etc/systemd/system/camigo.service` para garantizar que los contenedores del software se inicien automáticamente con el sistema operativo véase la **Figura 17**. En este archivo se definieron las instrucciones para levantar y detener los contenedores, con la opción `Restart=always` para asegurar la persistencia del servicio.

```

[cis@localhost ~]$ systemctl status aplicacion_camigo.service
● aplicacion_camigo.service - Aplicación C'amigo - Docker Compose
   Loaded: loaded (/etc/systemd/system/aplicacion_camigo.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-01-17 16:49:47 -05; 560ms ago
     Main PID: 14983 (docker-compose)
       Tasks: 9 (limit: 22397)
      Memory: 13.0M
         CPU: 106ms
   CGroup: /system.slice/aplicacion_camigo.service
           └─14983 /usr/local/bin/docker-compose up --build --no-deps -d

ene 17 16:49:47 localhost.localdomain systemd[1]: Started Aplicación C'amigo - Docker Compose.
ene 17 16:49:48 localhost.localdomain docker-compose[14983]: time="2025-01-17T16:49:48-05:00" level=warning msg="/opt/aplicaciones_dockerizadas/aplicacion_c
ene 17 16:49:48 localhost.localdomain docker-compose[14983]: #0 building with "default" instance using docker driver
ene 17 16:49:48 localhost.localdomain docker-compose[14983]: #1 [backend internal] load build definition from Dockerfile
ene 17 16:49:48 localhost.localdomain docker-compose[14983]: #1 transferring dockerfile: 531B done
ene 17 16:49:48 localhost.localdomain docker-compose[14983]: #1 DONE 0.1s
ene 17 16:49:48 localhost.localdomain docker-compose[14983]: #2 [backend internal] load metadata for docker.io/library/node:18-alpine
ene 17 16:49:48 localhost.localdomain docker-compose[14983]: #2 [backend internal] load metadata for docker.io/library/node:18-alpine

```

Figura 17. Archivo de configuración del servicio Camigo.

- **Validación del sistema:** Finalmente, se validó el funcionamiento del sistema ejecutando el comando `docker ps`, confirmando que todos los contenedores se encontraban activos después de reiniciar el servidor. En la **Figura 18** se muestra la validación realizada.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
63737cf4b96a	aplicacion_camigo-backend	"docker-entrypoint.s..."	4 days ago	Up 8 hours	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
aplicacion_camigo-backend-1					
e5be76ca8a41	mysql:8.0	"docker-entrypoint.s..."	4 days ago	Up 8 hours (healthy)	33060/tcp, 0.0.0.0:3307->3306/tcp, [::]:3307->3306/tcp
p	mysql				

Figura 18. Validación de contenedores activos.

Finalmente se tuvo como resultado la aplicación web funcionando en los servidores de la carrera, en la **Figura 19** se presenta la aplicación web desplegada.

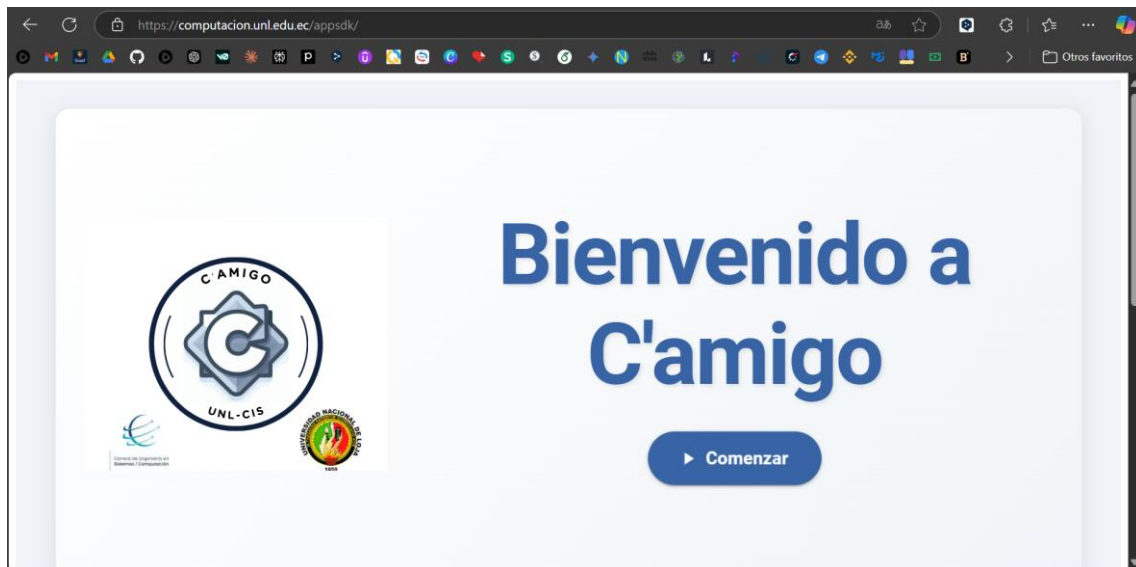


Figura 19. Software de apoyo desplegado.

6.1.5.1 Manual de usuario para estudiante

El manual de usuario que pertenece a los usuarios con rol “estudiante”, se encuentra en la sección de Anexos, (véase sección **Anexo 8**).

6.1.5.2 Manual de usuario para docente

El manual de usuario que pertenece a los usuarios con rol “docente”, se encuentra en la sección de Anexos, (véase sección **Anexo 9**).

6.1.5.3 Manual de usuario para administrador

El manual de usuario que pertenece a los usuarios con rol “administrador”, se encuentra en la sección de Anexos, (véase sección **Anexo 10**).

6.2 Objetivo 2: Determinar la aceptación tecnológica del software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante el modelo TAM.

Para dar por cumplido el segundo objetivo se evaluó la aceptación del software de apoyo mediante el modelo de aceptación tecnológica TAM, donde en la sección **6.2.1**. se muestra la elaboración del cuestionario de evaluación en base a los cuatro constructos del modelo TAM, en la sección **6.2.2** muestra la tabulación de resultados considerando la escala Likert, la última sección **6.2.3** muestra el análisis de resultados

6.2.1 Fase de elaboración

En esta fase, se presenta el cuestionario elaborado para poder determinar la percepción del usuario final al hacer uso del software de apoyo. En la **Tabla 12**, se presentan las variables latentes o constructores y el cuestionario con los 14 ítems llamados indicadores y también en la **Tabla 13**, se presentan las valoraciones numéricas para cada respuesta escogida por el usuario.

Tabla 12. Variables para medir la aceptación de los usuarios.

Constructo	Identificadores	Preguntas
Utilidad percibida (UP)	UP1	¿Considera que el software de apoyo especializado en el aprendizaje de programación en C facilita el proceso de comprensión de los conceptos de programación en C en comparación con otras herramientas como por ejemplo HackerRank, Khan Academy?
	UP2	¿Considera que el software de apoyo especializado en el aprendizaje de programación en C es práctico de utilizar en comparación con otras herramientas como por ejemplo HackerRank, Khan Academy?
	UP3	¿Considera que es útil el software de apoyo especializado en el aprendizaje de programación en C, como herramienta para el proceso de enseñanza/aprendizaje de programación en C?
Facilidad de uso percibida (FUP)	FUP1	¿Está de acuerdo que es fácil de utilizar el software de apoyo especializado en el aprendizaje de programación en C en el laboratorio de software de la carrera de computación?

Actitud hacia el uso (AU)	FUP2	¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C es una herramienta que ayuda en su proceso de enseñanza/aprendizaje?
	FUP3	¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C es fácil navegar y encontrar la información que necesita?
	AU1	¿Se siente cómodo/a al utilizar el software de apoyo especializado en el aprendizaje de programación en C?
	AU2	¿Considera que el uso del software de apoyo especializado en el aprendizaje de programación en C es beneficioso para agilizar su proceso de enseñanza/aprendizaje?
Intención de uso (IU)	AU3	¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C, es una buena idea en general?
	AU4	¿Considera agradable el uso del software de apoyo especializado en el aprendizaje de programación en C?
	IU1	¿Tiene usted la intención de utilizar el software de apoyo especializado en el aprendizaje de programación en C?
	IU2	¿Está de acuerdo que el uso del software de apoyo especializado en el aprendizaje de programación en C sería beneficioso en el futuro?
	IU3	¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C mejorará la experiencia académica en el laboratorio?

Las repuestas para cada pregunta están basadas en la escala Likert y se le asignó un valor.

Tabla 13. Valoraciones numéricas para cada respuesta escogida por el usuario

Respuesta	Valor asignado
Totalmente en desacuerdo	0
En desacuerdo	1
Ni de acuerdo, ni en desacuerdo	2
De acuerdo	3
Totalmente de acuerdo	4

6.2.2 Fase de medición

En esta fase, se presentan los resultados tras realizar la encuesta a los usuarios mediante la herramienta Google Forms donde en la **Tabla 14** se presenta la matriz de resultados donde se recopila las respuestas seleccionadas por los usuarios.

Tabla 14. Matriz de resultados en base a cada constructo

Up	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo, ni desacuerdo	De acuerdo	Totalmente de acuerdo
UP1	0%	0%	10,9%	50%	39,1%
UP2	0%	0%	13%	50%	37%
UP3	0%	0%	2,1%	37%	60,9%
Fup	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo, ni desacuerdo	De acuerdo	Totalmente de acuerdo
FUP1	0%	0%	4,4%	39,1%	56,5%
FUP2	0%	4,4%	0%	47,8%	47,8%
FUP3	0%	0%	4,4%	47,8%	47,8%
Au	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo, ni desacuerdo	De acuerdo	Totalmente de acuerdo
AU1	0%	0%	8,7%	34,8%	56,5%
AU2	0%	0%	2,2%	50%	47,8%
AU3	0%	0%	4,4%	30,4%	65,2%
AU4	0%	0%	4,4%	41,3%	54,3%
Iu	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo, ni desacuerdo	De acuerdo	Totalmente de acuerdo
IU1	0%	0%	13%	32,6%	54,4%
IU2	0%	0%	2,1%	37%	60,9%
IU3	0%	0%	2,2%	34,8%	63%

6.2.3 Fase de análisis.

El análisis de resultados se presenta mediante un análisis por pregunta de cada constructo, lo que facilita una comprensión más ordenada y clara de los resultados obtenidos de las fases anteriores.

6.2.3.1 Análisis de los resultados de la utilidad percibida (UP)

En el elemento UP1 correspondiente a la interrogante de: “¿Considera que el software de apoyo especializado en el aprendizaje de programación en C facilita el proceso de comprensión de los conceptos de programación en C en comparación con otras herramientas como por ejemplo HackerRank, Khan Academy?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en desacuerdo 0%, ni de acuerdo, ni desacuerdo 10,9%, de acuerdo 50% totalmente de acuerdo 39,1% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 89,1% de acuerdo con la utilidad percibida, mientras que el 10,9% se encuentra en un estado neutro.

En el elemento UP2 correspondiente a la interrogante de: “¿Considera que el software de apoyo especializado en el aprendizaje de programación en C es práctico de utilizar en comparación con otras herramientas como por ejemplo HackerRank, Khan Academy?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en

desacuerdo 0%, ni de acuerdo, ni desacuerdo 13%, de acuerdo 50% totalmente de acuerdo 37% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 87% de acuerdo con la utilidad percibida, mientras que el 13% se encuentra en un estado neutro.

En el elemento UP3 correspondiente a la interrogante de: “¿Considera que es útil el software de apoyo especializado en el aprendizaje de programación en C, como herramienta para el proceso de enseñanza/aprendizaje de programación en C?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en desacuerdo 0%, ni de acuerdo, ni desacuerdo 2,1%, de acuerdo 37% totalmente de acuerdo 60,9% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 97,9% de acuerdo con la utilidad percibida, mientras que el 2,1% se encuentra en un estado neutro.

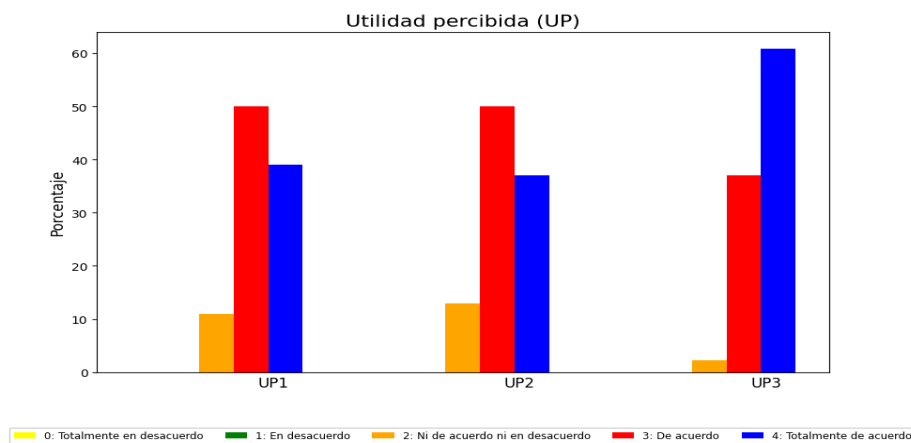


Figura 20. Respuestas graficadas de Utilidad Percibida [Fuente propia].

En resumen, el total del constructo de Utilidad Percibida (UP), muestra que el 91.33% de los participantes está entre de acuerdo y totalmente de acuerdo con la utilidad percibida del software de apoyo, mientras que un 8.66% se encuentra en estado neutro, ósea ni en de acuerdo ni en desacuerdo.

6.2.3.2 Análisis de los resultados de la Facilidad de uso percibida (FUP)

En el elemento FUP1 correspondiente a la interrogante de: “¿Está de acuerdo que es fácil de utilizar el software de apoyo especializado en el aprendizaje de programación en C en el laboratorio de software de la carrera de computación?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en desacuerdo 0%, ni de acuerdo, ni desacuerdo 4,4%, de acuerdo 39,1% totalmente de acuerdo 56,5% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas

variables nos arroja un 95,6% de acuerdo con la facilidad de uso percibida, mientras que el 4,4% se encuentra en un estado neutro.

En el elemento FUP2 correspondiente a la interrogante de: “¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C es una herramienta que ayuda en su proceso de enseñanza/aprendizaje?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en desacuerdo 4,4%, ni de acuerdo, ni desacuerdo 0%, de acuerdo 47,8% totalmente de acuerdo 47,8% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 95,6% de acuerdo con la facilidad de uso percibida, mientras que el 4,4% se encuentra en desacuerdo.

En el elemento FUP3 correspondiente a la interrogante de: “¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C es fácil navegar y encontrar la información que necesita?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en desacuerdo 0%, ni de acuerdo, ni desacuerdo 4,4%, de acuerdo 47,8% totalmente de acuerdo 47,8% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 95,6% de acuerdo con la facilidad de uso percibida, mientras que el 4,4% se encuentra en un estado neutro.

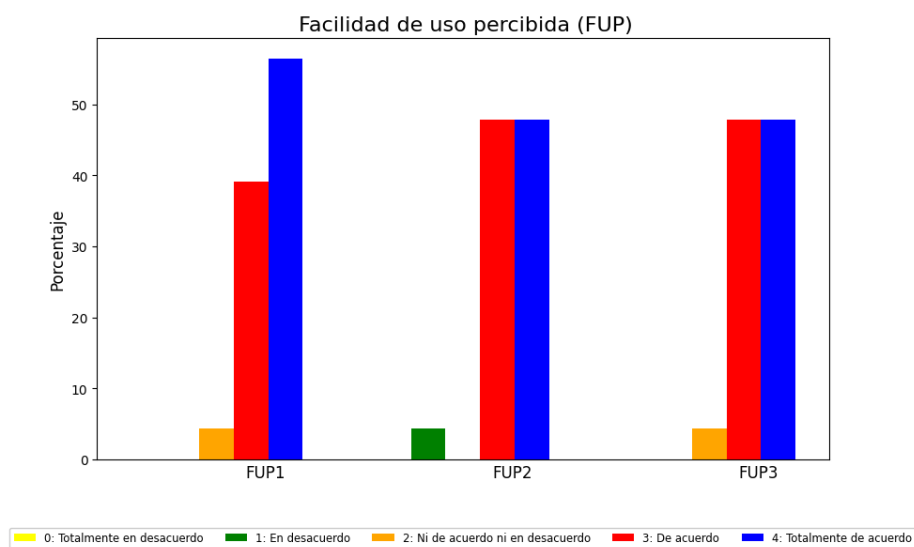


Figura 21. Respuestas graficadas de la Facilidad de Uso Percibida [Fuente propia].

En resumen, el total del constructo de Facilidad de Uso Percibida (FUP), muestra que el 95.6% de los participantes están entre de acuerdo y totalmente de acuerdo con la facilidad de uso percibida del software de apoyo, mientras que un 2.93% se encuentra en estado neutro, ósea

ni en de acuerdo ni en desacuerdo y una mínima cantidad de usuarios con el 1.46% de usuarios se encuentra en desacuerdo

6.2.3.3 Análisis de los resultados de la Actitud hacia el uso (AU)

En el elemento AU1 correspondiente a la interrogante de: “¿Se siente cómodo/a al utilizar el software de apoyo especializado en el aprendizaje de programación en C?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en desacuerdo 0%, ni de acuerdo, ni desacuerdo 8,7%, de acuerdo 34,8% totalmente de acuerdo 56,5% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 91,3% de acuerdo con la actitud hacia el uso, mientras que el 8,7% se encuentra en un estado neutro.

En el elemento AU2 correspondiente a la interrogante de: ¿Considera que el uso del software de apoyo especializado en el aprendizaje de programación en C es beneficioso para agilizar su proceso de enseñanza/aprendizaje?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en desacuerdo 0%, ni de acuerdo, ni desacuerdo 2,2%, de acuerdo 50% totalmente de acuerdo 47,8% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 97,8% de acuerdo con la actitud hacia el uso, mientras que el 2,2% se encuentra en un estado neutro.

En el elemento AU3 correspondiente a la interrogante de: “¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C, es una buena idea en general?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en desacuerdo 0%, ni de acuerdo, ni desacuerdo 4,4%, de acuerdo 30,4% totalmente de acuerdo 65,2% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 95,6% de acuerdo con la actitud hacia el uso, mientras que el 4,4% se encuentra en un estado neutro.

En el elemento AU4 correspondiente a la interrogante de: “¿Considera agradable el uso del software de apoyo especializado en el aprendizaje de programación en C?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en desacuerdo 0%, ni de acuerdo, ni desacuerdo 4,4%, de acuerdo 41,3% totalmente de acuerdo 54,3% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 95,6% de acuerdo con la actitud hacia el uso, mientras que el 4,4% se encuentra en un estado neutro.

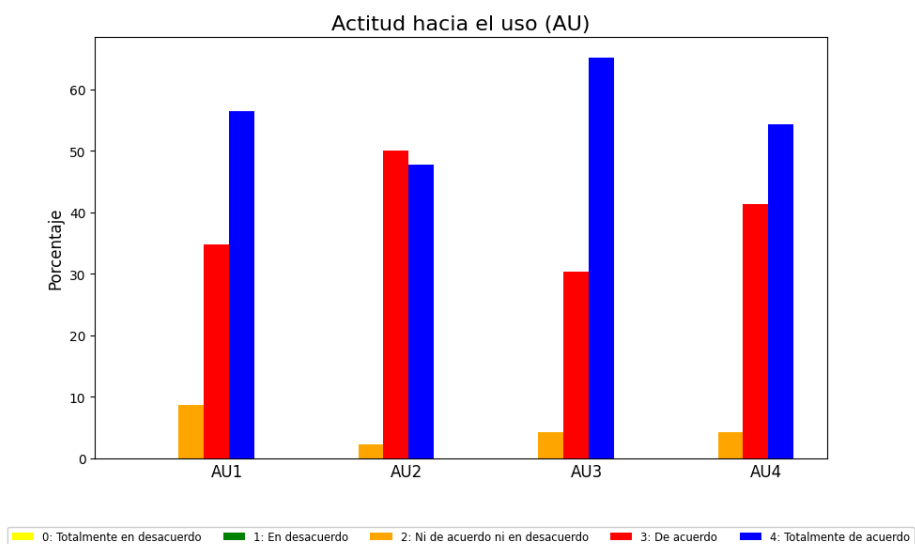


Figura 22. Respuestas graficadas de la Actitud Hacia el Uso [Fuente propia].

En resumen, el total del constructo de Actitud hacia el Uso (AU), muestra que el 95.07% de los participantes están entre de acuerdo y totalmente de acuerdo con la actitud hacia el uso del software de apoyo, mientras que un 4.92% se encuentra en estado neutro, ósea ni en de acuerdo ni en desacuerdo.

6.2.3.4 Análisis de los resultados de la Intención de Uso (IU)

En el elemento IU1 correspondiente a la interrogante de: “¿Tiene usted la intención de utilizar el software de apoyo especializado en el aprendizaje de programación en C?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en desacuerdo 0%, ni de acuerdo, ni desacuerdo 13%, de acuerdo 32,6% totalmente de acuerdo 54,4% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 87% de aceptación. Mientras que el 13% se encuentra en un estado neutro.

En el elemento IU2 correspondiente a la interrogante de: “¿Está de acuerdo que el uso del software de apoyo especializado en el aprendizaje de programación en C sería beneficioso en el futuro?”, nos arrojó los siguientes resultados, encuestados que están totalmente en desacuerdo un 0%, en desacuerdo 0%, ni de acuerdo, ni desacuerdo 2,1%, de acuerdo 37% totalmente de acuerdo 60,9% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 97,9% de aceptación. Mientras que el 2,1% se encuentra en un estado neutro.

En el elemento IU3 correspondiente a la interrogante de: “¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C mejorará la experiencia académica en el laboratorio?”, nos arrojó los siguientes resultados, encuestados que

están totalmente en desacuerdo un 0%, en desacuerdo 0%, ni de acuerdo, ni desacuerdo 2,2%, de acuerdo 34,8% totalmente de acuerdo 63% lo cual nos indica que la mayoría de personas encuestas están, de acuerdo y totalmente de acuerdo lo cual entre ambas variables nos arroja un 97,8% de aceptación. Mientras que el 2,2% se encuentra en un estado neutro.

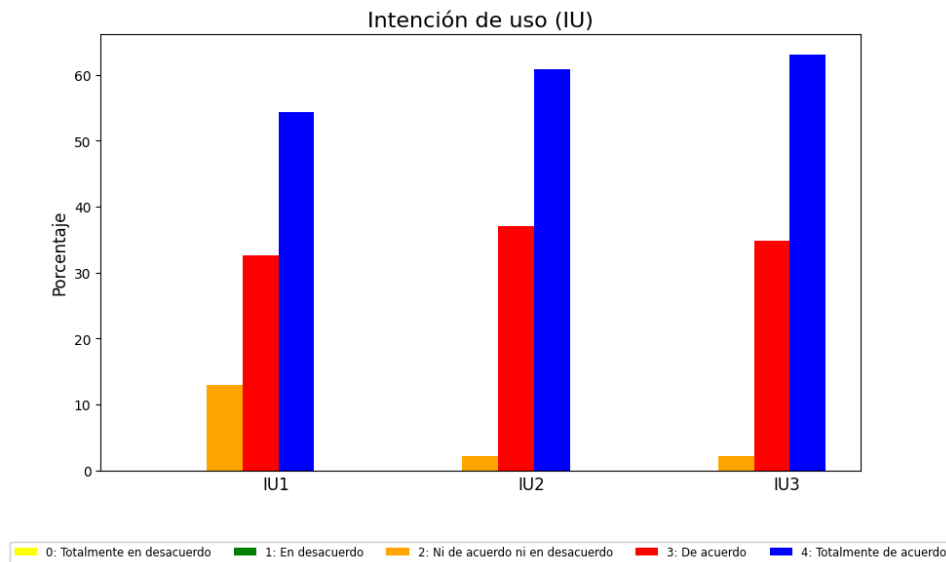


Figura 23. Respuestas graficadas de la Intención de Uso [Fuente propia].

En resumen, el total del constructo de Intención de Uso (IU), muestra que el 94.23% de los participantes está entre de acuerdo y totalmente de acuerdo con la intención de uso del software de apoyo, mientras que un 5.76% se encuentra en estado neutro, ósea ni en de acuerdo ni en desacuerdo. Los resultados obtenidos en la implementación del modelo de aceptación tecnológica TAM fueron los siguientes en el primer constructo de **Utilidad Percibida (UP)**, demuestra que el 91.33% de los encuestados está entre de acuerdo y totalmente de acuerdo con la utilidad percibida del software de apoyo, mientras que un 8.66% se encuentra en estado neutro, ósea ni en de acuerdo ni en desacuerdo, en el segundo constructo de **Facilidad de Uso Percibida (FUP)**, muestra que el 95.6% de los participantes están entre de acuerdo y totalmente de acuerdo con la facilidad de uso percibida del software de apoyo, mientras que un 2.93% se encuentra en estado neutro, lo que quiere decir que no estan ni en de acuerdo ni en desacuerdo mientras que el 1.46% de usuarios se encuentra en desacuerdo, el tercer constructo de **Actitud hacia el Uso (AU)**, muestra que el 95.07% de los participantes están entre de acuerdo y totalmente de acuerdo con la actitud hacia el uso del software de apoyo, mientras que un 4.92% se encuentra ni en de acuerdo ni en desacuerdo, para finalizar el ultimo constructo **Intención de Uso (IU)**, muestra que el 94.23% de los participantes está entre de acuerdo y totalmente de, mientras que un 5.76% se encuentra en estado neutro, ósea ni en de acuerdo ni en desacuerdo.

7 Discusión

7.1 Objetivo 1: Construir un software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante la metodología XP.

Este objetivo se centró en la construcción de un software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación de la Universidad Nacional de Loja, utilizando la metodología de desarrollo ágil XP. A continuación, se presentan las experiencias obtenidas y los resultados alcanzados durante la ejecución de este objetivo.

Durante el desarrollo de este objetivo, se lograron varios aciertos relacionados con las tecnologías seleccionadas, la metodología aplicada y el funcionamiento del software desarrollado. Sin embargo, también surgieron desafíos que requirieron ajustes en cada fase de la metodología, lo que influyó en los tiempos de desarrollo del software.

Entre los principales aciertos, se destaca la implementación del software como una herramienta de apoyo educativo en programación en C, diseñada específicamente para el plan de estudios del primer ciclo de la carrera de Computación. Este enfoque permitió que el software sea una herramienta de apoyo a las necesidades curriculares y prácticas de los estudiantes, facilitando un entorno para reforzar conceptos clave y practicar ejercicios de manera intra o extra clase.

Además, la elección de tecnologías como Node.js, Express y React resultó adecuada para el desarrollo del software, ya que permitieron la construcción de una aplicación web modular.

La aplicación de la metodología de desarrollo XP fue otro acierto destacado. Al tratarse de una metodología de desarrollo ágil, permitió que el desarrollo del software se realizara de manera iterativa e incremental, asegurando la entrega de prototipos funcionales en cada iteración. Esto facilitó la validación y retroalimentación constante, garantizando que el producto final cumpliera con los requisitos establecidos.

Durante el desarrollo del primer objetivo, se identificaron diversas limitaciones en cada una de las fases de la metodología XP, que generaron desafíos significativos para el cumplimiento del propósito planteado. A continuación, se detallan estas limitaciones, junto con las estrategias aplicadas para superarlas.

En la fase de planificación, uno de los principales retos fue la recopilación y análisis de los requisitos e historias de usuario del sistema. A pesar de contar con un plan de estudios (silabo) claro y bien definido, se requirió un trabajo adicional para abstraer los aspectos

específicos que debían integrarse en el software. Para abordar esta dificultad, se recurrió a técnicas como la revisión de documentos curriculares y entrevistas con docentes de la carrera, lo que permitió clarificar y detallar los requisitos esenciales para el desarrollo.

La fase de diseño presentó complicaciones pequeñas, principalmente en la conceptualización de los diagramas UML, tales como el de clases y componentes. La necesidad de crear un diseño que sea fácil de entender por todos los involucrados y que sea mantenible. Sin embargo, gracias al uso de herramientas de modelado y a la colaboración constante con los interesados, se logró construir un diseño que reflejaba adecuadamente los requisitos y que facilitaba la transición hacia la etapa de codificación.

En la fase de codificación, las dificultades más notables estuvieron relacionadas con la integración de funcionalidades clave, como que el docente pueda ingresar contenido mediante texto enriquecido, encontrar editores en línea que permitan ser embebidos y el progreso de los usuarios dentro del sistema. Estas funcionalidades requirieron un nivel técnico avanzado en la configuración de entornos de desarrollo y la implementación de componentes complejos.

La fase de pruebas evidenció limitaciones en términos de cobertura, especialmente en las pruebas automatizadas para verificar el correcto funcionamiento de la interacción usuario-sistema en diferentes escenarios. Aunque se realizaron pruebas, la falta de tiempo en el TIC impidió desarrollar una cobertura amplia de pruebas. Por lo tanto, es importante destacar que, con un plazo de tiempo más amplio, habría sido posible incrementar significativamente la cobertura de las pruebas unitarias. Si bien alcanzar un 100% de cobertura podría no ser factible debido a la complejidad y extensión del sistema y sus componentes, se habría logrado una ampliación considerable en la realización de dichas pruebas.

Finalmente, la fase de lanzamiento enfrentó retrasos relacionados con la configuración del entorno de despliegue. Estos problemas incluyeron incompatibilidades entre versiones de dependencias y ajustes en el servidor destinado para la implementación, debido a la falta de disponibilidad en ciertos momentos. A pesar de estas complicaciones, se realizaron esfuerzos significativos para adaptar el código a los requerimientos del entorno de producción, asegurando un despliegue funcional del sistema dentro de los laboratorios de la carrera.

7.2 Objetivo 2: Determinar la aceptación tecnológica del software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación, mediante el modelo TAM.

Este objetivo se centró en evaluar la aceptación tecnológica del software de apoyo especializado en el aprendizaje de programación en C, utilizando como referencia el Modelo de Aceptación Tecnológica (TAM). A continuación, se detallan las experiencias y resultados obtenidos durante las etapas clave de este proceso.

Un logro destacado fue la selección del Modelo TAM como marco metodológico para la evaluación. Este modelo resultó ser una herramienta eficaz para analizar la percepción de los usuarios sobre el software de apoyo, ya que permite evaluar la aceptación tecnológica de manera estructurada a través de cuatro variables fundamentales: Utilidad percibida, Facilidad de uso percibida, Actitud hacia el uso e Intención de uso.

El diseño de las encuestas para medir estas variables fue otro punto fuerte del proceso, las preguntas fueron desarrolladas con base en investigaciones previas y ajustadas a las necesidades específicas del software de apoyo, lo que aseguró que la recopilación de datos fuera precisa y relevante. Además, la escala de Likert utilizada permitió cuantificar la percepción de los usuarios de manera efectiva, proporcionando resultados claros y comprensibles para el análisis posterior que se compartieron en el presente TIC.

El proceso de recolección de datos, aunque desafiante, fue exitoso. Se logró demostrar el software a estudiantes de la carrera en escenarios reales, lo que permitió obtener retroalimentación directa y aplicar encuestas inmediatamente después de su interacción con el sistema. Esto garantizó que las respuestas fueran representativas de la experiencia de los usuarios y no de percepciones teóricas o preconcebidas.

En la fase de elaboración, el principal desafío fue garantizar que las preguntas reflejaran fielmente los cuatro constructos evaluados por el Modelo TAM. Aunque la falta de guías específicas en el contexto local representó un obstáculo inicial, este fue superado mediante una revisión exhaustiva de trabajos similares y consultas con expertos en el área.

La fase de medición fue especialmente compleja debido a los retos logísticos. Conseguir permisos para realizar las demostraciones del software y aplicar las encuestas dentro de horarios académicos requirió un esfuerzo adicional de coordinación. Asimismo, la disponibilidad de los estudiantes fue limitada, lo que restringió el número de participantes y añadió presión al cronograma. A pesar de estas dificultades, se logró cumplir con la muestra planificada, asegurando resultados representativos.

En la fase de análisis, el volumen de datos recopilados presentó un reto significativo. Analizar e interpretar las respuestas obtenidas de las encuestas, de manera que reflejaran adecuadamente las percepciones de los usuarios, demandó más tiempo y esfuerzo de lo esperado. Sin embargo, con herramientas de análisis estadístico y la aplicación de técnicas de interpretación cualitativa y cuantitativa, fue posible extraer conclusiones sólidas y bien fundamentadas.

Los resultados obtenidos en cada una de las variables del Modelo TAM indican niveles medios a altos de aceptación tecnológica del software. Los usuarios percibieron el sistema como útil y fácil de usar, con una actitud positiva hacia su implementación y una intención clara de utilizarlo en el futuro. Estos hallazgos respaldan la viabilidad del software como herramienta de apoyo para el aprendizaje de programación en C y confirman que cumple con los requisitos de aceptación tecnológica esperados.

Por lo tanto, para responder a la pregunta de investigación formulada, la cual es: “**¿Qué aceptación tecnológica se logrará obtener al utilizar el software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja, mediante el método de aceptación tecnológica TAM?**”. Los resultados permiten afirmar que el software ha alcanzado un nivel de aceptación tecnológica medio a alto, evidenciando que es una herramienta bien recibida por los estudiantes y con un potencial significativo para ser utilizada como recurso de aprendizaje en el entorno académico.

8 Conclusiones

Tras finalizar el presente TIC, se ha llegado a las siguientes conclusiones:

- El desarrollo del software especializado para el aprendizaje de programación en C, diseñado según el plan de estudios del primer ciclo de la carrera de Computación de la Universidad Nacional de Loja, cumplió satisfactoriamente su propósito. Esta herramienta ofrece a los estudiantes un entorno tanto teórico como práctico que refuerza su aprendizaje, facilitando la comprensión de conceptos fundamentales del lenguaje de programación C. Su diseño intuitivo y la facilidad de uso permitieron una adopción sin dificultades, dado que no requiere conocimientos técnicos avanzados para su manejo, lo que asegura su integración efectiva tanto dentro como fuera del aula.
- La aplicación de la metodología XP en el desarrollo del software de apoyo ha demostrado ser efectiva al permitir una adaptación ágil a los cambios en los requisitos del proyecto, fomentando la colaboración estrecha entre el desarrollador y los usuarios finales, y priorizando la satisfacción del cliente. Esto ha resultado en la entrega de un producto que cumple con las necesidades específicas requeridas.
- La evaluación del software de apoyo especializado para el aprendizaje de programación en C, utilizando el Modelo de Aceptación Tecnológica (TAM), permitió obtener información clave sobre la percepción de los usuarios en dimensiones como la utilidad percibida, la facilidad de uso percibida, la actitud hacia el uso y la intención de uso. Este modelo proporcionó una estructura clara y objetiva para analizar la aceptación tecnológica, facilitando la identificación de áreas de mejora y promoviendo estrategias para optimizar la experiencia de los usuarios. Los resultados reflejan una aceptación positiva, fundamentada en datos cuantitativos obtenidos mediante la escala de Likert, lo que respalda no solo la viabilidad del software como herramienta de apoyo, sino también su potencial para fomentar una adopción tecnológica más amplia y satisfactoria en el entorno académico.
- El desarrollo del software de apoyo especializado para el aprendizaje de programación en C ha demostrado ser una solución eficiente y aceptada por los usuarios, según los resultados obtenidos mediante el Modelo de Aceptación Tecnológica (TAM). Estos resultados reflejan niveles de aceptación tecnológica entre medios y altos, evidenciando percepciones positivas en términos de

utilidad percibida, facilidad de uso percibida, actitud hacia el uso e intención de continuar utilizando la herramienta. Este análisis confirma que el software cumple con las expectativas de los usuarios en el contexto educativo, apoyando eficazmente el proceso de aprendizaje en los laboratorios de la carrera de Computación.

9 Recomendaciones

Al finalizar el presente TIC, se enlistan las siguientes recomendaciones:

- Se recomienda implementar un proceso de evaluación continua del software de apoyo especializado en programación en C, recopilando retroalimentación periódica de docentes y estudiantes. Este enfoque permitirá identificar áreas de mejora, incorporar nuevas funcionalidades y ajustar el software a posibles cambios en el plan de estudios. De esta manera, se garantizará que la herramienta se mantenga actualizada, relevante y alineada con las necesidades educativas, enriqueciendo de forma constante la experiencia de aprendizaje.
- Se recomienda explorar la posibilidad de ampliar la aplicación del software a otras áreas de estudio dentro de la carrera de Computación o incluso a disciplinas relacionadas, con el fin de maximizar su impacto educativo y promover un enfoque interdisciplinario en el aprendizaje de programación. Esta expansión podría requerir ajustes y personalizaciones específicas para cada contexto de uso, así como la colaboración con expertos en las respectivas áreas temáticas.
- Se recomienda considerar la realización de estudios adicionales que profundicen en la evaluación de la aceptación tecnológica del software de apoyo mediante el modelo TAM, con el objetivo de seguir refinando la experiencia de usuario, identificar posibles barreras de adopción y diseñar estrategias efectivas para promover una mayor utilización de la herramienta en los entornos educativos.

9.1 Trabajos futuros

Para futuros trabajos relacionados con el software de apoyo especializado en programación en C para los laboratorios de la carrera de Computación, se recomienda explorar la integración de tecnologías emergentes, como la inteligencia artificial y el aprendizaje automático, para enriquecer las funcionalidades y capacidades del software. Asimismo, se podría investigar la posibilidad de desarrollar versiones adaptadas del software para diferentes plataformas y dispositivos móviles, con el fin de brindar a los usuarios una experiencia de aprendizaje más flexible y accesible. Además, sería relevante profundizar en la evaluación del impacto pedagógico del software en el proceso de enseñanza y aprendizaje de la programación, mediante estudios longitudinales que analicen el rendimiento académico de los estudiantes y su desarrollo de habilidades informáticas.

10 Bibliografía

- [1]. M. J. Bezanilla, "Bases técnico-pedagógicas para la elaboración de software educativo," Universidad Nacional de Educación a Distancia, pp. 164-167, 2010.
- [2]. M. Díaz-Antón, M. Pérez, A. Grimmán, y L. Mendoza, "Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica," 2003. [En línea]. Disponible en: <https://www.redalyc.org/pdf/737/73712305.pdf>
- [3]. A. C. Vilarinho y J. A. T. Machado, "Gestão de Projectos de Software: Uma Abordagem Prática," Revista Ibérica de Sistemas e Tecnologias de Informação (RIST), vol. 41, pp. 50-62, 2023.
- [4]. A. García Rodríguez, "Enseñanza de la programación a través de Scratch para el desarrollo del pensamiento computacional en educación básica secundaria," Academia y Virtualidad, vol. 15, no. 1, pp. 161-182, 2022, doi: 10.18359/ravi.5883.
- [5]. Educaixa, "R(r)evolución educativa es aplicar el Aprendizaje Basado en Proyectos como modelo para reformar la escuela. Qué dice la evidencia sobre el ABP," 2019. [En línea]. Disponible en: <https://educaixa.org/documents/10180/0/Beneficios+del+ABP.pdf> de <https://educaixa.org/documents/10180/0/Beneficios+del+ABP.pdf/ad5642ce-d7ab-923e-3660-e13503cbe070?t=1575268766985>
- [6]. S. D. Brookfield, "Self-Directed Learning," en Encyclopedia of Distributed Learning, A. DiStefano, K. E. Rudestam, y R. Silverman, Eds. Los Angeles, CA, USA: SAGE Publications, pp. 397-399, 2004.
- [7]. J. Biggs y C. Tang, Teaching for Quality Learning at University, 4th ed. Maidenhead, UK: Open University Press, 2011.
- [8]. Universidad Politécnica de Madrid (UPM), "Guía Docente: Aprendizaje Basado en Problemas," Facultad de Educación, 2016. [En línea]. Disponible en: https://innovacioneducativa.upm.es/sites/default/files/guias/Aprendizaje_basado_en_problemas.pdf
- [9]. I. Dussel y M. Caruso, "La educación y la escuela en tiempos de pandemia: entre la emergencia y la oportunidad," Revista Iberoamericana de Educación, vol. 77, no. 1, pp. 13-34, 2020. [En línea]. Disponible en: <https://www.redalyc.org/pdf/666/66619992009.pdf>
- [10]. I. Martínez-Ortiz y F. J. García-Peñalvo, "Propuesta de didáctica de la Programación en Educación Primaria basada en la gamificación usando videojuegos educativos," Revista de Educación a Distancia (RED), vol. 54, pp. 1-22, 2020. [11]. J.

- M. de la Fuente y A. García-Valcárcel, "Enseñanza de la programación en educación primaria: una revisión de la literatura," *Revista de Educación*, vol. 364, pp. 147-167, 2014. [En línea]. Disponible en: <https://core.ac.uk/download/pdf/296347797.pdf>
- [12]. J. P. Díaz Quilla, G. Z. Carbonell Alta, y D. J. Picho Durand, "Los sistemas de gestión de aprendizaje (LMS) en la educación universitaria," *Revista del Centro de Investigación e Innovación Educativa*, vol. 50, no. 2, pp. 95-112, 2021. [En línea]. Disponible en: <https://revista.grupocieg.org/wp-content/uploads/2021/06/Ed.5087-95-Diaz-Carbonel-Picho.pdf>
 - [13]. M. A. Galaz-Fontes y C. Martínez-Garrido, "Evaluación de las competencias informacionales en estudiantes universitarios: una experiencia con rúbricas," *Formación Universitaria*, vol. 13, no. 1, pp. 13-26, 2020. [En línea]. Disponible en: <https://www.scielo.cl/pdf/formuniv/v13n1/0718-5006-formuniv-13-01-13.pdf>
 - [14]. Universität Oldenburg, "Reader-Seminar WS2016," Departamento de Informática, 2016. [En línea]. Disponible en: <https://uol.de/f/2/dept/informatik/ag/svs/download/reader/reader-seminar-ws2016.pdf>
 - [15]. V. Chinnakotla, *MySQL 8 Administrator's Guide: Effective Guide to Administering High-Performance MySQL 8 Solutions*, 1st ed. Birmingham, UK: Packt Publishing, 2018.
 - [16]. M. A. Alvarez, "¿Qué es una SPA?," *desarrolloweb.com*, 2016. [En línea]. Disponible en: <https://desarrolloweb.com/articulos/quees-una-spa.html>
 - [17]. J. Smith, "Un nuevo algoritmo para la clasificación de documentos," *CiteSeerX*, 2023. [En línea]. Disponible en: <https://citeseerx.ist.psu.edu/>
 - [18]. A. M. Prado Salvador y V. G. Moreta, "Aplicación de la Metodología de Desarrollo de Ingeniería de Software Extreme Programming (XP) para un Sistema de Gestión de Prácticas Pre-Profesionales para los estudiantes de la Facultad de Ingeniería en la Pontificia Universidad Católica del Ecuador," *Pontificia Universidad Católica del Ecuador*, 2016.
 - [19]. D. A. Camacho Velíz, "Solución Informática para la administración de procedimientos odontológicos," *Universidad Nacional de Loja*, 2019.
 - [20]. K. Rungta, *Learn NodeJS in 1 Day: Complete Node JS Guide with Examples*, 2016.
 - [21]. ISTQB, "Probador Certificado del ISTQB®," 2018.
 - [22]. R. S. Pressman, *Ingeniería del software: un enfoque práctico*, 7ma ed. 2013. [En línea]. Disponible en:

<http://gen.lib.rus.ec/book/index.php?md5=927F3EBCBEEAD67C38E08FCE46279565>

- [23]. P. J. García-Sánchez y M. López-Coronado, "Las pruebas de integración como proceso de la calidad del software en el ámbito de las telecomunicaciones," Universidad Carlos III de Madrid, 2010.
- [24]. D. Rodríguez et al., Pruebas de software y JUnit: un análisis en profundidad y ejemplos prácticos. Madrid, España: Pearson Educación, 2008. [En línea]. Disponible en:
<http://gen.lib.rus.ec/book/index.php?md5=F86C4A0971C3A4B1A2C680B7605A4C0D>
- [25]. R. S. Pressman, Ingeniería del software: un enfoque práctico, 7ma ed. 2013. [En línea]. Disponible en:
<http://gen.lib.rus.ec/book/index.php?md5=927F3EBCBEEAD67C38E08FCE46279565>
- [26]. P. Puello, V. del Campo, y F. J. Scholborgh, "Modelo de Aceptación Tecnológica (TAM) en el Laboratorio de Física III basado en Internet de las Cosas en el Programa de Ingeniería de Sistemas de la Universidad de Cartagena, Colombia," Revista ESPACIOS, vol. 798, p. 1015, 2020.
- [27]. A. León-Mejía y B. Garzón-García, "Modelo de aceptación tecnológica (TAM): un estudio de la influencia de la cultura nacional y del perfil del usuario en el uso de las TIC," Revista de Ciencias Sociales, vol. 20, no. 1, pp. 15-30, 2013.
- [28]. L. A. Reyes, "RPCS_V3-1_a02," RINvestigación, 2023.
- [29]. P. López-Roldán y R. Hernández-Sampieri, "Escala de Likert: Una alternativa para elaborar e interpretar un instrumento de percepción social," ResearchGate, 2011.
- [30]. R. G. Figueroa Díaz y C. J. Ordoñez Tandazo, "Desarrollo de una aplicación web interactiva para el aprendizaje de conceptos básicos de la asignatura de programación para estudiantes de primer ciclo de la carrera de Ciencias de la Computación," Universidad Nacional de Loja. [En línea]. Disponible en:
<https://dspace.unl.edu.ec/jspui/handle/123456789/29948>
- [31]. E. A. Pisco Méndez, "Desarrollo de un Chatbot como soporte al proceso de aprendizaje de la programación en la UPS," Universidad Politécnica Salesiana, Sede Guayaquil, 2021. [En línea]. Disponible en:
<https://dspace.ups.edu.ec/bitstream/123456789/22288/1/UPS-GT003679.pdf>

- [32]. B. N. Rivadeneira Zambrano, "Aprendizaje de programación usando inteligencia artificial: obtención de retroalimentación del código fuente de un programa escrito con lenguaje C mediante inteligencia artificial," Escuela Politécnica Nacional, Quito, Ecuador, 2024. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/26200>
- [33]. L. Gallego et al., "BD4OPEM H2020 project. The 4+ 1 View Model of Software Architecture for enabling AI-based services in distribution grids," 2021.

11 Anexos

Anexo 1: Entrevistas realizadas a los docentes del área de programación



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Entrevistado:**
 - Ing. Edison Leonardo Coronel Romero, docente encargado de la materia Programación Orientada a Objetos.

Descripción: El presente anexo contiene la transcripción íntegra de la entrevista realizada al Ingeniero Edison Coronel, docente responsable de impartir la asignatura de Programación Orientada a Objetos. Esta entrevista se llevó a cabo con el propósito específico de recopilar información crucial para la definición de los requisitos fundamentales y las funcionalidades del sistema de Trabajo de Integración Curricular (TIC). El objetivo primordial de esta actividad fue garantizar el desarrollo de un sistema que responda de manera efectiva a las necesidades particulares de la carrera de Computación y, en especial, a los requerimientos académicos de los estudiantes del primer ciclo.



UNL

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Entrevista como parte del proceso de recopilación de información para definir los requisitos y funcionalidades del sistema de TIC.

ENTREVISTADO:	Ing. Edison Leonardo Coronel Romero	REALIZADO POR:	Juan Carlos Armijos Sánchez		
FECHA DE LA ENTREVISTA	07/11/2023	HORA DE INICIO DE LA ENTREVISTA	10:45 AM	HORA DE FINALIZACIÓN DE LA ENTREVISTA	11:30 AM

Introducción a la entrevista

Título del TIC: Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja

Tema a tratar: Recopilación de información para definir los requisitos y funcionalidades del sistema de TIC.

Cargo del entrevistado: Docente encargado de la materia Teoría de la programación.

Entrevistador.	Entrevistado.
¿Qué funcionalidades considera esenciales para el software?	En primer lugar, resulta fundamental definir las temáticas a tratar, centrándonos especialmente en los objetivos de aprendizaje que los estudiantes alcanzarán al abordar dichos temas. Asimismo, es crucial comprender cómo evaluar el progreso de los estudiantes dentro del sistema. Además, podría considerarse la incorporación de una sección de recursos adicionales relacionados con las temáticas, brindando a los estudiantes información adicional para enriquecer su comprensión.
¿Cuáles son los principales desafíos que enfrenta al enseñar programación en C en el aula y cómo podría el software ayudar a abordarlos?	El principal desafío radica en la aplicación del pensamiento computacional, especialmente en la transición de una solución clara en mente a un código concreto. Para lograr esto, es necesario traducir los conceptos del mundo real a las diversas sentencias, secuencias y estructuras de control presentes en un lenguaje de programación. Por ejemplo, al enfrentarse a un proceso de toma de decisiones, surge la pregunta de cómo vincularlo con los elementos temáticos en C, que en este caso estarían representados por las estructuras condicionales. Este proceso de traducción, de pasar de la conceptualización en el mundo real al lenguaje específico de programación, constituye el desafío fundamental en este contexto.
¿Qué tipo de recursos didácticos o ejemplos de código le gustaría ver incluidos en el software?	En el ámbito de la programación, es crucial proporcionar ejemplos con diversas sintaxis y, además, presentar alternativas en la generación de ejemplos de código. Tomando el caso específico del cálculo del número mayor entre dos, es posible crear representaciones equivalentes empleando distintas estructuras. Por ejemplo, se podría utilizar una sentencia de control tradicional, optar por un enfoque con un switch, o incluso modularizar el proceso mediante funciones, entre otras posibilidades. La idea es ofrecer variedad en las formas de abordar el mismo concepto, brindando a los aprendices múltiples enfoques para comprender y aplicar estas funciones.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

<p>¿Cómo imagina la integración de este software en su plan de estudios o en sus actividades de enseñanza?</p>	<p>En este contexto, las actividades prácticas y experimentales, que son esenciales para el aprendizaje de los jóvenes, junto con el trabajo autónomo, representan componentes clave. Las actividades autónomas, en particular, desempeñan un papel fundamental en el proceso educativo. Aunque las interacciones directas con el docente pueden abordar otros aspectos, es imperativo integrar ejemplos y ejercicios variados en estas actividades prácticas y experimentales, así como en el trabajo autónomo. La integración de estos elementos enriquecerá significativamente la experiencia de aprendizaje, permitiendo a los estudiantes aplicar y consolidar sus conocimientos de programación de manera más efectiva.</p>
<p>¿Existen características o funcionalidades específicas que considere cruciales para evaluar el progreso de los estudiantes en programación en C?</p>	<p>La comprensión de la sintaxis, el pensamiento computacional y la lógica de programación constituyen elementos esenciales en el aprendizaje y dominio de la programación y que deben estar presentes en el software de apoyo.</p>
<p>¿Qué enfoque pedagógico o metodología de enseñanza prefiere y cómo podría el software adaptarse a ello?</p>	<p>Considero que sería beneficioso implementar preguntas breves y repetitivas en intervalos regulares. Por ejemplo, podríamos preguntar ahora sobre la sintaxis de una estructura condicional, y luego, después de un periodo, volver a plantear la misma pregunta. A través de esta repetición, los estudiantes tendrían la oportunidad de reforzar y consolidar su aprendizaje de manera gradual. Este enfoque favorecería un proceso de aprendizaje más efectivo y duradero.</p>

Entrevistado	FIRMA	FECHA
<p>Ing. Edison Leonardo Coronel Romero</p>		<p>07/11/2023</p>



UNL

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Entrevistado:**
 - Ing. María del Cisne Ruilova, docente encargado de la materia Diseño de Software.

Descripción: El presente anexo contiene la transcripción íntegra de la entrevista realizada a la Ingeniera María del Cisne Ruilova, docente responsable de impartir la asignatura de Diseño de Software. Esta entrevista se llevó a cabo con el propósito específico de recopilar información crucial para la definición de los requisitos fundamentales y las funcionalidades del sistema de Trabajo de Integración Curricular (TIC). El objetivo primordial de esta actividad fue garantizar el desarrollo de un sistema que responda de manera efectiva a las necesidades particulares de la carrera de Computación y, en especial, a los requerimientos académicos de los estudiantes del primer ciclo.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Entrevista como parte del proceso de recopilación de información para definir los requisitos y funcionalidades del sistema de TIC.

ENTREVISTADO:	Ing. María del Cisne Rulova		REALIZADO POR:	Juan Carlos Armijos Sánchez	
FECHA DE LA ENTREVISTA	10/11/2023	HORA DE INICIO DE LA ENTREVISTA	10:30 AM	HORA DE FINALIZACIÓN DE LA ENTREVISTA	10:50 AM
Introducción a la entrevista					
Título del TIC: Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja					
Tema a tratar: Recopilación de información para definir los requisitos y funcionalidades del sistema de TIC.					
Cargo del entrevistado: Docente encargado de la materia Teoría de la programación.					

Entrevistador.	Entrevistado.
¿Qué funcionalidades considera esenciales para el software?	Es fundamental que el sistema pueda notificar al estudiante sobre nuevas actividades disponibles, como ejercicios recién añadidos por el docente. Además, sería beneficioso poder agregar enlaces a recursos adicionales, más allá de lo ya proporcionado, permitiendo al docente compartir enlaces a videos, plataformas u otros materiales relevantes en una sección designada. Otra funcionalidad que se podría agregar sería la capacidad de asignar los ejercicios existentes como tareas específicas para los estudiantes, notificándoles de estas asignaciones.
¿Cuáles son los principales desafíos que enfrenta al enseñar programación en C en el aula y cómo podría el software ayudar a abordarlos?	El principal desafío radica en que los estudiantes a menudo optan por soluciones que ya existen en la web en lugar de realizar un análisis propio de los ejercicios planteados en clase. Este comportamiento representa una dificultad significativa, ya que la estructura del aprendizaje se centra en comprender la metodología y la manipulación de datos, pero en la práctica de estos conocimientos es lo que se queda pendiente.
¿Qué tipo de recursos didácticos o ejemplos de código le gustaría ver incluidos en el software?	Sería de gran importancia contar con una gran variedad de recursos y ejemplos relacionados con las estructuras de control. Es fundamental que estos recursos ofrezcan una explicación clara y concisa sobre el propósito de cada estructura, cómo funcionan y proporcionen ejemplos. Personalmente, encuentro sumamente valiosas las pruebas de escritorio como recurso didáctico al enseñar programación ya que estas pruebas muestran la implementación de una estructura y permiten realizar pruebas para visualizar su correcto funcionamiento. Considero que este tipo de recursos didácticos, que muestran la estructura, su funcionalidad y su aplicación a través de pruebas de escritorio, resultan siendo los más útiles.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

<p>¿Cómo imagina la integración de este software en su plan de estudios o en sus actividades de enseñanza?</p>	<p>Considero que este recurso sería sumamente valioso, ya que a menudo los estudiantes pueden olvidar lo que se cubrió en clase una vez que termina la sesión. Incluso después de recibir dos horas de instrucción, es común que olviden una gran parte del contenido, es por ello que considero que esta herramienta podría ser una solución eficaz. Les permitiría practicar y acceder a recursos adicionales, sirviendo como un punto centralizado para la carrera, donde podrían revisar y consolidar lo que se brindó en clase.</p>
<p>¿Existen características o funcionalidades específicas que considere cruciales para evaluar el progreso de los estudiantes en programación en C?</p>	<p>Para evaluar el progreso de los estudiantes en programación, considero crucial la implementación de funcionalidades que fomenten la práctica a través de la resolución de ejercicios. Es esencial presentarles problemas que desafíen a los estudiantes, que se les exija no solo la solución final, sino también un análisis profundo, el diseño detallado de la solución y su implementación utilizando las estructuras del lenguaje de programación</p>
<p>¿Qué enfoque pedagógico o metodología de enseñanza prefiere y cómo podría el software adaptarse a ello?</p>	<p>Mi preferencia pedagógica para la enseñanza de programación comienza en clase con las sesiones magistrales que proporcionen una base teórica sólida sobre el funcionamiento de lo esencial, que son los conceptos. Sin embargo, considero vital complementar esta base teórica con una aplicación práctica de forma inmediata. La comprensión de los fundamentos teóricos debe ir de la mano con su aplicación práctica. Esta combinación es esencial: si falta la comprensión teórica, la aplicación se vuelve casi imposible de comprender, y si solo se tiene conocimiento teórico sin nada nada práctico, carece de utilidad. Por ende, mi enfoque se centra en integrar la teoría y la aplicación práctica de manera complementaria para lograr una comprensión profunda y útil para los estudiantes, un enfoque que intento reflejar en el diseño de mis clases.</p>

Entrevistado	FIRMA	FECHA
Ing. María del Cisne Rullova		07/11/2023



UNL

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Entrevistado:**
 - Ing. Wilman Patricio Chamba Zaragocín, docente encargado de la materia Teoría de la Programación.

Descripción: El presente anexo contiene la transcripción íntegra de la entrevista realizada al Ingeniero Wilman Chamba, docente responsable de impartir la asignatura de Teoría de la Programación. Esta entrevista se llevó a cabo con el propósito específico de recopilar información crucial para la definición de los requisitos fundamentales y las funcionalidades del sistema de Trabajo de Integración Curricular (TIC). El objetivo primordial de esta actividad fue garantizar el desarrollo de un sistema que responda de manera efectiva a las necesidades particulares de la carrera de Computación y, en especial, a los requerimientos académicos de los estudiantes del primer ciclo.



Entrevista como parte del proceso de recopilación de información para definir los requisitos y funcionalidades del sistema de TIC.

ENTREVISTADO:	Ing. Wilman Patricio Chamba Zaragocín		REALIZADO POR:	Juan Carlos Armijos Sánchez	
FECHA DE LA ENTREVISTA	24/10/2023	HORA DE INICIO DE LA ENTREVISTA	8:30 AM	HORA DE FINALIZACIÓN DE LA ENTREVISTA	8:50 AM

Introducción a la entrevista

Título del TIC: Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja

Tema a tratar: Recopilación de información para definir los requisitos y funcionalidades del sistema de TIC.

Cargo del entrevistado: Docente encargado de la materia Teoría de la programación.

Entrevistador.	Entrevistado.
¿Qué funcionalidades considera esenciales para el software?	Considero fundamental permitir que los estudiantes se registren en la aplicación únicamente durante el ciclo académico. De esta manera, si un estudiante reprueba una materia, tendrá la oportunidad de utilizar el sistema desde 0 en el siguiente ciclo para practicar y mejorar sus habilidades. Además, es esencial que los docentes tengan la capacidad de gestionar el contenido de la aplicación, permitiéndoles crear nuevos temas, subtemas y ejercicios, junto con sus soluciones, así como proporcionar ejemplos de ejecución de los ejercicios.
¿Cuáles son los principales desafíos que enfrenta al enseñar programación en C en el aula y cómo podría el software ayudar a abordarlos?	Considero que el enfoque principal radica en la lógica de razonamiento que se aborda. En el caso específico de C, es crucial dominar el uso de las palabras reservadas dentro de su estructura. Por ejemplo, es esencial saber identificar las secciones correspondientes a los encabezados o cabeceras para incluir las librerías necesarias, así como reconocer las áreas destinadas a las funciones y a la declaración de variables dentro del código en C y de igual manera que algo que quiero que tomes en cuenta es las buenas prácticas de programación al momento de mostrar las soluciones de los ejercicios que brinda la aplicación y de también que en el transcurso de aprendizaje dentro del sistema le aparezcan tips de programación limpia o agregar un temario de programación limpia en el sistema.
¿Qué tipo de recursos didácticos o ejemplos de código le gustaría ver incluidos en el software?	Desde mi punto de vista, resulta crucial brindar a los estudiantes ejemplos sencillos que les permitan no solo observar la solución, sino también ver una ejecución práctica del código. Esto les proporcionará la oportunidad de verificar si, al utilizar los mismos datos de entrada que se muestran en el ejemplo, obtienen un resultado similar.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

<p>¿Cómo imagina la integración de este software en su plan de estudios o en sus actividades de enseñanza?</p>	<p>En mi opinión, considero que estos ejercicios podrían ser incorporados tanto en talleres como en actividades de trabajo autónomo. Es especialmente valioso tenerlos disponibles como parte de una exposición guiada por el profesor, donde se pueden presentar demostraciones para que los estudiantes puedan seguir practicando y profundizando en el tema. A lo largo de su práctica, sería beneficioso aumentar progresivamente la complejidad de los ejercicios y permitir a los estudiantes comparar sus soluciones con las del sistema, que, como mencionamos previamente, seguirá buenas prácticas de programación para fomentar el desarrollo de las habilidades de los estudiantes.</p>
<p>¿Existen características o funcionalidades específicas que considere cruciales para evaluar el progreso de los estudiantes en programación en C?</p>	<p>La única manera de medir el progreso de los estudiantes es mediante la realización de controles de lectura a medida que avanzan a través de cada subtema.</p>
<p>¿Qué enfoque pedagógico o metodología de enseñanza prefiere y cómo podría el software adaptarse a ello?</p>	<p>Mi preferencia en cuanto a la metodología de enseñanza se centra en la realización de ejercicios, resolución de problemas y demostración práctica. Dado que el aprendizaje en su mayoría se logra a través de la práctica, mi enfoque se basa en un proceso que incluye el análisis de los problemas, la creación de algoritmos para resolverlos y la posterior implementación de código para aplicar esas soluciones.</p>

Entrevistado	FIRMA	FECHA
Ing. Wilman Patricio Chamba Zaragocín		24/10/2023

Anexo 2: Requerimientos esenciales del sistema



UNL

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Revisado y aprobado por:**
 - Ing. José Oswaldo Guamán Quinche, director del Trabajo de Integración Curricular.

Descripción: El presente anexo recopila los requerimientos fundamentales del sistema, los cuales han sido identificados en colaboración con los usuarios directos. Dichos requerimientos constituyen una descripción general de las funcionalidades y características que se implementaron en el software, sirviendo como base para las historias de usuario.



UNL

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

La siguiente tabla expone el conjunto de requisitos esenciales del sistema, organizados en orden ascendente y clasificados mediante parámetros clave, como código, requisito, descripción y nivel de prioridad. Esta estructura facilita su interpretación y gestión.

Tabla 1. Requisitos esenciales del sistema

Código	Requisito	Descripción	Prioridad
RF001	Crear usuario	El sistema permitirá al usuario registrarse en el software de apoyo proporcionando nombres, apellidos, correo electrónico y contraseña	Media
RF002	Editar usuarios	El sistema permitirá a los usuarios registrados modificar su información personal, nombres, apellidos, correo electrónico y contraseña	Media
RF003	Listar temas	El sistema permitirá al docente listar todos los temas activos e inactivos.	Media
RF004	Registrar temas	El sistema permitirá al docente crear nuevos temas al proporcionar: título, objetivos de aprendizaje, descripción y recursos adicionales.	Alta
RF005	Editar temas	El sistema permitirá al docente la capacidad de modificar información relacionada con temas, tanto activos como inactivos, lo que incluirá la posibilidad de editar el título, los objetivos de aprendizaje, descripción y recursos adicionales.	Media
RF006	Historial de cambios por tema	El sistema permitirá al docente visualizar hora y fecha de que se realizó algún cambio en el tema seleccionado.	Media
RF007	Activar/Desactivar temas	El sistema permitirá al docente activar y desactivar temas existentes.	Media
RF008	Buscar temas	El sistema permitirá al usuario buscar temas por criterios específicos (Título del tema).	Baja
RF009	Visualizar temas	El sistema permitirá al usuario visualizar todos los temas activos en el sistema.	Media



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

RF010	Listar subtemas	El sistema permitirá al docente listar todos los subtemas activos e inactivos.	Media
RF011	Registrar subtemas	El sistema permitirá al docente crear nuevos subtemas derivados de temas existentes al ingresar los siguientes parámetros: título, objetivos de aprendizaje, descripción, ejemplo de código y recursos adicionales.	Alta
RF012	Editar subtemas	El sistema permitirá al docente editar información de subtemas existentes, el usuario podrá editar siguientes parámetros: título, objetivos de aprendizaje, descripción, ejemplo de código y recursos adicionales.	Media
RF013	Historial de cambios por subtema	El sistema permitirá al docente visualizar hora y fecha de que se realizó algún cambio en el subtema seleccionado.	Media
RF014	Activar/Desactivar subtemas	El sistema permitirá al docente activar y desactivar subtemas existentes.	Media
RF015	Buscar subtemas	El sistema permitirá al usuario buscar subtemas por criterios específicos (Título del subtema).	Baja
RF016	Visualizar subtemas	El sistema permitirá al usuario visualizar todos los subtemas activos en el sistema.	Media
RF017	Listar ejercicios	El sistema permitirá al docente listar todos los ejercicios activos e inactivos.	Media
RF018	Registrar ejercicios	El sistema permitirá al docente crear nuevos ejercicios derivados de subtemas existentes al ingresar los siguientes parámetros: título, instrucciones, restricciones, solución del ejercicio y retroalimentación.	Alta
RF019	Editar ejercicios	El sistema permitirá al docente editar información de ejercicios existentes, el usuario podrá editar los siguientes parámetros: título, instrucciones, restricciones, solución del ejercicio y retroalimentación.	Media



UNL

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

RF020	Historial de cambios por ejercicio	El sistema permitirá al docente visualizar hora y fecha de que se realizó algún cambio en el ejercicio seleccionado.	Media
RF021	Activar/Desactivar ejercicios	El sistema permitirá al docente activar y desactivar ejercicios existentes.	Media
RF022	Buscar ejercicios	El sistema permitirá al usuario buscar ejercicios por criterios específicos (Titulo del ejercicio).	Baja
RF023	Visualizar ejercicios	El sistema permitirá al usuario visualizar todos los ejercicios activos en el sistema.	Media
RF024	Proporcionar retroalimentación de ejercicios	El sistema permitirá al estudiante acceder a la solución codificada del ejercicio y una retroalimentación del mismo	Alta
RF025	Listar preguntas de control	El sistema permitirá al docente listar todas las preguntas de control activas e inactivas.	Media
RF026	Registrar pregunta de control	El sistema permitirá al docente crear nuevas preguntas de control derivados de ejercicios existentes al ingresar los siguientes parámetros: enunciado, opciones, respuesta correcta y justificación de la respuesta.	Alta
RF027	Editar pregunta de control	El sistema permitirá al docente editar información de preguntas de control existentes, el usuario podrá editar los siguientes parámetros: enunciado, opciones, respuesta correcta y justificación de la respuesta.	Media
RF028	Historial de cambios por pregunta de control	El sistema permitirá al docente visualizar hora y fecha de que se realizó algún cambio en la pregunta de control seleccionada.	Media
RF029	Activar/Desactivar pregunta de control	El sistema permitirá al docente activar y desactivar preguntas de control existentes.	Media
RF030	Buscar preguntas de control	El sistema permitirá al usuario buscar preguntas de control por criterios específicos (Titulo de la pregunta).	Baja
RF031	Visualizar preguntas de control	El sistema permitirá al estudiante visualizar la	Alta



UNL

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

		pregunta de control de conocimientos de cada ejercicio.	
RF032	Responder preguntas de control	El sistema permitirá al estudiante responder preguntas de control de conocimientos en cada ejercicio, y al hacerlo, el sistema incrementará el progreso del usuario.	Alta
RF033	Listar comentarios	El sistema permitirá al usuario listar todos los comentarios.	Media
RF034	Registrar comentarios	El sistema permitirá al usuario crear nuevos comentarios derivados de ejercicios existentes al ingresar su descripción	Alta
RF035	Editar comentarios	El sistema permitirá al usuario editar información de comentarios existentes, el usuario podrá editar la descripción del comentario.	Media
RF036	Visualizar comentarios	El sistema permitirá al usuario ver todos los comentarios de cada ejercicio proporcionado por el software de apoyo.	Media
RF037	Visualizar progreso	El sistema permitirá al estudiante visualizar su progreso de aprendizaje.	Media
RF038	Cambiar periodo académico	El sistema permitirá al administrador cambiar el periodo académico, lo que habilitará al sistema para reiniciar el progreso de todos los usuarios en el sistema.	Media
RF039	Temas destacados	El sistema permitirá a los usuarios evaluar los temas, con el propósito de identificar y destacar los cinco contenidos mejor valorados.	Baja
RF040	Notificaciones por email de nuevo contenido	El sistema enviará notificaciones a todos los usuarios cada vez que un docente publique nuevo contenido en el software.	Media



UNL

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Ing. José Oswaldo Guamán Quinche, Mg. Sc.



Firmado electrónicamente por:
JOSE OSWALDO GUAMAN
QUINCHE

Anexo 3: Historias de usuario



UNIVERSIDAD NACIONAL DE LOJA



UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Tema de TIC: Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

Fecha: 13 de noviembre del 2023

Participantes:

- Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- Ing. Wilman Chamba Mg. Sc. Docente de la materia “Teoría de la programación”.
- Ing. Edison Coronel Mg. Sc. Docente de la materia “Programación Orientada a Objetos”.
- Ing. René Guamán Mg. Sc. Docente de la materia “Programación Paralela”.

Descripción: Durante la fase de planificación de la metodología Extreme Programming (XP) aplicada en este Trabajo de Integración Curricular (TIC), se recopilieron las siguientes historias de usuario. Estas historias se basan en los requisitos esenciales identificados por los clientes directos del sistema y representan una extensión más detallada y específica de dichos requerimientos.

Primera iteración

Historia de usuario	
Número: 001	Nombre: Crear usuario
Usuario: Administrador, Docente, Estudiante	
Iteración asignada: 1	Referencia de requisito: RF001
Prioridad en el negocio (Alta/Media/Baja): Alta	Fecha de inicio: 20/11/2023
Riesgo en el desarrollo (Alta/Media/Baja): Baja	Fecha de entrega: 12/12/2023
Descripción:	
<p>COMO usuario no registrado QUIERO crear una cuenta de usuario PARA acceder al software de apoyo y sus funcionalidades</p>	
Observaciones:	
<p>La funcionalidad de crear usuario estará disponible para cualquier persona que acceda al sistema y no tenga una cuenta existente. El proceso de registro requerirá información básica del usuario para garantizar la autenticidad de las cuentas y permitir una mejor experiencia dentro del software de apoyo.</p>	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Mostrar un formulario de registro accesible desde la página principal o de inicio de sesión. • Solicitar los siguientes datos obligatorios: nombres, apellidos, correo electrónico y contraseña. • Validar que todos los campos obligatorios estén completos antes de procesar el registro. • Verificar que el correo electrónico ingresado no esté asociado a una cuenta existente. • Mostrar mensajes de error claros si alguna validación falla (campos incompletos, correo duplicado). • Crear la cuenta de usuario en el sistema al dar clic en el botón "Registrar" si todos los datos son válidos. • Mostrar un mensaje de confirmación indicando que la cuenta ha sido creada exitosamente. • Redirigir al usuario al dashboard del software de apoyo tras el registro exitoso. 	



Historia de usuario

Número: 002	Nombre: Editar usuario
Usuario: Administrador, Docente, Estudiante	
Iteración asignada: 1	Referencia de requisito: RF002
Prioridad en el negocio (Alta/Media/Baja): Alta	Fecha de inicio: 20/11/2023
Riesgo en el desarrollo (Alta/Media/Baja): Baja	Fecha de entrega: 12/12/2023

Descripción:

COMO usuario registrado
QUIERO editar mi información de perfil
PARA mantener actualizados mis datos en el software de apoyo

Observaciones:

La funcionalidad de editar usuario estará disponible para todos los usuarios autenticados en el sistema. Esta función permitirá a los usuarios actualizar su información personal, manteniendo la integridad y actualidad de los datos en el software de apoyo.

Criterios de aceptación:

- Mostrar la opción "Editar perfil" en el menú de usuario.
 - Presentar un formulario con los datos actuales del usuario: nombres, apellidos, correo electrónico y contraseña.
 - Permitir la modificación de los campos: nombres, apellidos, correo electrónico y contraseña.
 - Validar que los campos modificados no estén vacíos antes de guardar los cambios.
 - Mostrar mensajes de error claros si alguna validación falla (campos vacíos).
 - Actualizar la información del usuario en el sistema al hacer clic en el botón "Guardar cambios".
 - Mostrar un mensaje de confirmación indicando que los datos han sido actualizados exitosamente.
 - Reflejar inmediatamente los cambios en la interfaz del usuario tras la actualización exitosa.
-



Historia de usuario

Número: 003

Nombre: Listar temas

Usuario: Docente

Iteración asignada: 1

Referencia de requisito: RF003

Prioridad en el negocio
(Alta/Media/Baja): Media

Fecha de inicio: 20/11/2023

Riesgo en el desarrollo
(Alta/Media/Baja): Media

Fecha de entrega: 12/12/2023

Descripción:

COMO docente

QUIERO listar temas

PARA conocer todos los temas existentes en el software de apoyo

Observaciones:

La funcionalidad de listar todos los temas existentes estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Estos usuarios tendrán acceso a una vista completa que incluirá todos los temas almacenados en el sistema, independientemente de su estado (Activo o Inactivo), permitiendo así una gestión integral del contenido académico.

Criterios de aceptación:

- Mostrar la página de "Gestionar Contenido".
 - Presentar todos los temas existentes por su título en la tabla "Temas".
 - Diferenciar visualmente los temas inactivos mediante celdas con un color rojo distintivo, facilitando su identificación respecto a los temas activos.
 - Habilitar la selección de cualquier tema dentro de la tabla "Temas" mediante interacción del docente.
 - Resaltar el tema seleccionado por el docente con un fondo de color gris, proporcionando una indicación visual clara de la opción elegida.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 004	Nombre: Registrar temas
Usuario: Docente	
Iteración asignada: 1	Referencia de requisito: RF004
Prioridad en el negocio (Alta/Media/Baja): Alta	Fecha de inicio: 20/11/2023
Riesgo en el desarrollo (Alta/Media/Baja): Alta	Fecha de entrega: 12/12/2023

Descripción:

COMO docente
QUIERO registrar temas
PARA agregar nuevos temas al software de apoyo

Observaciones:

La capacidad de registrar nuevos temas en el sistema estará exclusivamente reservada para usuarios autenticados con el rol de "Docente", garantizando así el control y la calidad del contenido académico ingresado.

Criterios de aceptación:

- Desplegar el formulario de registro al seleccionar la opción "Registrar tema".
 - Validar que todos los campos obligatorios (título, objetivos de aprendizaje, descripción y recursos adicionales) estén completos antes de procesar el registro.
 - Verificar la unicidad del título ingresado en la base de datos para evitar duplicados.
 - Mostrar un mensaje de alerta claro si se detecta un título duplicado, informando al usuario que el tema ya existe.
 - Registrar el tema en la base de datos al hacer clic en el botón "Guardar", almacenando toda la información ingresada.
 - Mostrar un mensaje de confirmación indicando que el tema ha sido creado con éxito.
 - Redirigir automáticamente al usuario a la vista de "Gestionar contenido" tras el registro exitoso.
-



Historia de usuario

Número: 005

Nombre: Editar temas

Usuario: Docente

Iteración Asignada: 1

Referencia de requisito: RF005

Prioridad en el negocio **Fecha de inicio:** 20/11/2023

(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 12/12/2023

(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO poder editar los temas existentes

PARA actualizar y mantener al día la información en el software de apoyo

Observaciones:

La función de edición de temas existentes estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Estos usuarios podrán modificar cualquier tema almacenado en el sistema, independientemente de su estado actual (activo o inactivo), permitiendo así una gestión flexible y completa del contenido académico.

Criterios de aceptación:

- Permitir la selección de un tema existente desde la tabla "Temas".
 - Habilitar el botón "Editar tema" una vez seleccionado un tema.
 - Desplegar el formulario de edición con los datos del tema seleccionado.
 - Habilitar la modificación de los campos: título, objetivos de aprendizaje, descripción y recursos adicionales.
 - Validar que todos los campos requeridos estén completos antes de guardar.
 - Verificar que el título no exceda los 255 caracteres.
 - Comprobar que el título modificado no exista ya en la base de datos.
 - Mostrar un mensaje de alerta si se detecta un título duplicado.
 - Actualizar la información del tema al presionar el botón "Guardar".
 - Mostrar un mensaje de confirmación indicando que el tema ha sido modificado con éxito.
 - Redirigir automáticamente al usuario a la vista de "Gestionar contenido" tras la edición exitosa.
-



Historia de usuario

Número: 006

Nombre de Historia de Usuario: Historial de cambios por tema

Usuario: Docente

Iteración Asignada: 1

Referencia de requisito: RF006

Prioridad en el negocio **Fecha de inicio:** 20/11/2023

(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 12/12/2023

(Alta/Media/Baja): Baja

Descripción:

COMO docente

QUIERO visualizar el historial de cambios de un tema

PARA conocer cuándo y qué modificaciones se han realizado en el contenido

Observaciones:

La funcionalidad de ver el historial de cambios estará disponible exclusivamente para usuarios autenticados con el rol de "Docente". Esta función permitirá un seguimiento detallado de las modificaciones realizadas en cada tema, proporcionando transparencia y trazabilidad en la gestión del contenido académico.

Criterios de aceptación:

- Mostrar la opción "Ver historial" junto a cada a la tabla "Temas"
 - Al seleccionar "Ver historial", desplegar una lista cronológica de los cambios realizados en el tema.
 - Presentar para cada entrada del historial: Fecha y hora del cambio realizado.
 - Ordenar el historial de cambios del más reciente al más antiguo.
 - Asegurar que el historial se actualice automáticamente cuando se realicen nuevos cambios en el tema.
 - Mostrar un mensaje indicativo si el tema no tiene cambios registrados.
-



Historia de usuario

Número: 007

Nombre de Historia de Usuario:
Activar/Desactivar temas

Usuario: Docente

Iteración Asignada: 1

Referencia de requisito: RF007

Prioridad en el negocio **Fecha de inicio:** 20/11/2023

(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 12/12/2023

(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO activar/desactivar temas

PARA activar y desactivar temas existentes en el software de apoyo

Observaciones:

La facultad de activar o desactivar temas existentes en el sistema estará reservada exclusivamente para usuarios autenticados con el rol de "Docente". Esta restricción asegura un control preciso sobre la visibilidad y disponibilidad del contenido académico en el software de apoyo.

Criterios de aceptación:

- Listar todos los temas existentes en la tabla "Temas".
 - Diferenciar visualmente los temas inactivos con celdas de color rojo.
 - Permitir la selección de cualquier tema de la tabla "Temas".
 - Resaltar el tema seleccionado con un fondo gris.
 - Alternar el estado del tema (activo/inactivo) al presionar el botón "Activar/Desactivar".
 - Actualizar inmediatamente la visualización del tema en la tabla tras cambiar su estado.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 008

Nombre de Historia de Usuario: Buscar temas

Usuario: Docente

Iteración Asignada: 1

Referencia de requisito: RF008

Prioridad en el negocio **Fecha de inicio:** 20/11/2023

(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 12/12/2023

(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO buscar temas

PARA buscar temas existentes en el software de apoyo independiente de su estado (Activo/Inactivo)

Observaciones:

La funcionalidad de búsqueda de temas estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Esta medida garantiza un acceso controlado a la información académica, permitiendo a los docentes localizar eficientemente contenidos específicos dentro del sistema.

Criterios de aceptación:

- Presentar todos los temas existentes en la tabla "Temas".
 - Proporcionar un campo de búsqueda donde el usuario pueda ingresar el título del tema deseado.
 - Realizar la búsqueda automáticamente mientras el usuario escribe el título- del tema.
 - Mostrar un mensaje informativo si no se encuentra ningún tema con el título buscado.
 - Mantener visible la lista completa de temas, con el resultado de la búsqueda destacado.
 - Permitir borrar el texto del campo de búsqueda y reiniciar la visualización de todos los temas.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 009

Nombre de Historia de Usuario: Visualizar temas

Usuario: Administrador, Docente, Estudiante

Iteración Asignada: 1

Referencia de requisito: RF009

Prioridad en el negocio **Fecha de inicio:** 20/11/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 12/12/2023
(Alta/Media/Baja): Media

Descripción:

COMO usuario del sistema (docente, administrador o estudiante)

QUIERO acceder al contenido completo de los temas activos

PARA utilizar el material educativo disponible en el software de apoyo

Observaciones:

Esta funcionalidad estará disponible para todos los usuarios autenticados en el sistema, independientemente de su rol. Proporcionará acceso al contenido detallado de los temas activos, permitiendo a los usuarios explorar y utilizar el material educativo. A diferencia de la función de listar temas, esta opción mostrará el contenido completo y listo para su uso, facilitando el proceso de aprendizaje y enseñanza.

Criterios de aceptación:

- Mostrar la sección "Temas" en el menú principal del software.
 - Presentar una lista de todos los temas activos, mostrando sus títulos.
 - Permitir al usuario seleccionar un tema específico para ver su contenido detallado.
 - Al seleccionar un tema, mostrar su contenido completo, incluyendo: Título, Objetivos de aprendizaje, Descripción, Recursos adicionales.
 - Proporcionar opciones de búsqueda para facilitar la localización de temas específicos.
 - Incluir indicadores visuales de progreso mostrando el avance del usuario en los temas.
-



Segunda iteración

Historia de usuario	
Número: 0010	Nombre: Listar subtemas
Usuario: Docente	
Iteración asignada: 2	Referencia de requisito: RF010
Prioridad en el negocio (Alta/Media/Baja): Media	Fecha de inicio: 30/11/2023
Riesgo en el desarrollo (Alta/Media/Baja): Media	Fecha de entrega: 20/12/2023
Descripción: COMO docente QUIERO listar subtemas PARA conocer todos los subtemas existentes en el software de apoyo	
Observaciones: La funcionalidad de listar todos los subtemas existentes estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Estos usuarios tendrán acceso a una vista completa que incluirá todos los subtemas almacenados en el sistema, independientemente de su estado (Activo o Inactivo), permitiendo así una gestión integral del contenido académico.	
Criterios de aceptación: <ul style="list-style-type: none">• Mostrar la página de "Gestionar Contenido".• Presentar todos los subtemas existentes por su título en la tabla "Subtemas".• Diferenciar visualmente los subtemas inactivos mediante celdas con un color rojo distintivo, facilitando su identificación respecto a los subtemas activos.• Habilitar la selección de cualquier subtema dentro de la tabla "Subtemas" mediante interacción del docente.• Resaltar el subtema seleccionado por el docente con un fondo de color gris, proporcionando una indicación visual clara de la opción elegida.	



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0011

Nombre: Registrar subtemas

Usuario: Docente

Iteración asignada: 2

Referencia de requisito: RF011

Prioridad en el negocio
(Alta/Media/Baja): Alta

Fecha de inicio: 30/11/2023

Riesgo en el desarrollo
(Alta/Media/Baja): Alta

Fecha de entrega: 20/12/2023

Descripción:

COMO docente

QUIERO registrar subtemas

PARA agregar nuevos subtemas al software de apoyo

Observaciones:

La capacidad de registrar nuevos subtemas en el sistema estará exclusivamente reservada para usuarios autenticados con el rol de "Docente", garantizando así el control y la calidad del contenido académico ingresado.

Criterios de aceptación:

- Desplegar el formulario de registro al seleccionar la opción "Registrar subtema".
 - Validar que todos los campos obligatorios (título, objetivos de aprendizaje, descripción, ejemplo de código, recursos adicionales y retroalimentación) estén completos antes de procesar el registro.
 - Verificar la unicidad del título ingresado en la base de datos para evitar duplicados.
 - Mostrar un mensaje de alerta claro si se detecta un título duplicado, informando al usuario que el subtema ya existe.
 - Registrar el subtema en la base de datos al hacer clic en el botón "Guardar", almacenando toda la información ingresada.
 - Mostrar un mensaje de confirmación indicando que el subtema ha sido creado con éxito.
 - Redirigir automáticamente al usuario a la vista de "Gestionar contenido" tras el registro exitoso.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0012

Nombre: Editar subtemas

Usuario: Docente

Iteración Asignada: 2

Referencia de requisito: RF012

Prioridad en el negocio **Fecha de inicio:** 30/11/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 20/12/2023
(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO poder editar los subtemas existentes

PARA actualizar y mantener al día la información en el software de apoyo

Observaciones:

La función de edición de subtemas existentes estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Estos usuarios podrán modificar cualquier subtema almacenado en el sistema, independientemente de su estado actual (activo o inactivo), permitiendo así una gestión flexible y completa del contenido académico.

Criterios de aceptación:

- Permitir la selección de un subtema existente desde la tabla "Subtemas".
 - Habilitar el botón "Editar subtema" una vez seleccionado un subtema.
 - Desplegar el formulario de edición con los datos del subtema seleccionado.
 - Habilitar la modificación de los campos: título, objetivos de aprendizaje, descripción, ejemplo de código, recursos adicionales y retroalimentación.
 - Validar que todos los campos requeridos estén completos antes de guardar.
 - Verificar que el título no exceda los 255 caracteres.
 - Comprobar que el título modificado no exista ya en la base de datos.
 - Mostrar un mensaje de alerta si se detecta un título duplicado.
 - Actualizar la información del subtema al presionar el botón "Guardar".
 - Mostrar un mensaje de confirmación indicando que el subtema ha sido modificado con éxito.
 - Redirigir automáticamente al usuario a la vista de "Gestionar contenido" tras la edición exitosa.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0013

Nombre de Historia de Usuario: Historial de cambios por subtema

Usuario: Docente

Iteración Asignada: 2

Referencia de requisito: RF013

Prioridad en el negocio **Fecha de inicio:** 30/11/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 20/12/2023
(Alta/Media/Baja): Baja

Descripción:

COMO docente

QUIERO visualizar el historial de cambios de un subtema

PARA conocer cuándo y qué modificaciones se han realizado en el contenido

Observaciones:

La funcionalidad de ver el historial de cambios estará disponible exclusivamente para usuarios autenticados con el rol de "Docente". Esta función permitirá un seguimiento detallado de las modificaciones realizadas en cada subtema, proporcionando transparencia y trazabilidad en la gestión del contenido académico.

Criterios de aceptación:

- Mostrar la opción "Ver historial" junto a cada a la tabla "Subtemas"
 - Al seleccionar "Ver historial", desplegar una lista cronológica de los cambios realizados en el subtema.
 - Presentar para cada entrada del historial: Fecha y hora del cambio realizado.
 - Ordenar el historial de cambios del más reciente al más antiguo.
 - Asegurar que el historial se actualice automáticamente cuando se realicen nuevos cambios en el subtema.
 - Mostrar un mensaje indicativo si el subtema no tiene cambios registrados.
-



Historia de usuario

Número: 0014

Nombre de Historia de Usuario:

Activar/Desactivar subtemas

Usuario: Docente

Iteración Asignada: 2

Referencia de requisito: RF014

Prioridad en el negocio **Fecha de inicio:** 30/11/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 20/12/2023
(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO activar/desactivar subtemas

PARA activar y desactivar subtemas existentes en el software de apoyo

Observaciones:

La facultad de activar o desactivar subtemas existentes en el sistema estará reservada exclusivamente para usuarios autenticados con el rol de "Docente". Esta restricción asegura un control preciso sobre la visibilidad y disponibilidad del contenido académico en el software de apoyo.

Criterios de aceptación:

- Listar todos los subtemas existentes en la tabla "Subtemas".
 - Diferenciar visualmente los subtemas inactivos con celdas de color rojo.
 - Permitir la selección de cualquier subtema de la tabla "Subtemas".
 - Resaltar el subtema seleccionado con un fondo gris.
 - Alternar el estado del subtema (activo/inactivo) al presionar el botón "Activar/Desactivar".
 - Actualizar inmediatamente la visualización del subtema en la tabla tras cambiar su estado.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0015

Nombre de Historia de Usuario: Buscar subtemas

Usuario: Docente

Iteración Asignada: 2

Referencia de requisito: RF015

Prioridad en el negocio **Fecha de inicio:** 30/11/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 20/12/2023
(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO buscar subtemas

PARA buscar subtemas existentes en el software de apoyo independiente de su estado (Activo/Inactivo)

Observaciones:

La funcionalidad de búsqueda de subtemas estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Esta medida garantiza un acceso controlado a la información académica, permitiendo a los docentes localizar eficientemente contenidos específicos dentro del sistema.

Criterios de aceptación:

- Presentar todos los subtemas existentes en la tabla "Subtemas".
 - Proporcionar un campo de búsqueda donde el usuario pueda ingresar el título del subtema deseado.
 - Realizar la búsqueda automáticamente mientras el usuario escribe el título del subtema.
 - Mostrar un mensaje informativo si no se encuentra ningún subtema con el título buscado.
 - Mantener visible la lista completa de subtemas, con el resultado de la búsqueda destacado.
 - Permitir borrar el texto del campo de búsqueda y reiniciar la visualización de todos los subtemas.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0016

Nombre de Historia de Usuario: Visualizar subtemas

Usuario: Administrador, Docente, Estudiante

Iteración Asignada: 2

Referencia de requisito: RF016

Prioridad en el negocio **Fecha de inicio:** 30/11/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 20/12/2023
(Alta/Media/Baja): Media

Descripción:

COMO usuario del sistema (docente, administrador o estudiante)

QUIERO acceder al contenido completo de los subtemas activos

PARA profundizar en aspectos específicos del material educativo disponible en el software de apoyo

Observaciones:

Esta funcionalidad estará disponible para todos los usuarios autenticados en el sistema, independientemente de su rol. Proporcionará acceso al contenido detallado de los subtemas activos dentro de cada tema principal, permitiendo a los usuarios explorar y utilizar material educativo más específico y detallado. Esta opción mostrará el contenido completo de los subtemas, facilitando un aprendizaje más profundo y estructurado.

Criterios de aceptación:

- Mostrar la sección "Subtemas" en el menú principal del software.
 - Presentar una lista de todos los subtemas activos, mostrando sus títulos.
 - Permitir al usuario seleccionar un subtema específico para ver su contenido detallado.
 - Al seleccionar un subtema, mostrar su contenido completo, incluyendo: Título, Objetivos de aprendizaje, Descripción, Ejemplo código, Recursos adicionales y Retroalimentación.
 - Proporcionar opciones de búsqueda para facilitar la localización de subtemas específicos.
 - Incluir indicadores visuales de progreso mostrando el avance del usuario en los subtemas.
-



Tercera iteración

Historia de usuario

Número: 0017	Nombre: Listar ejercicios
Usuario: Docente	
Iteración asignada: 3	Referencia de requisito: RF017
Prioridad en el negocio (Alta/Media/Baja): Media	Fecha de inicio: 06/12/2023
Riesgo en el desarrollo (Alta/Media/Baja): Media	Fecha de entrega: 24/12/2023

Descripción:

COMO docente

QUIERO listar ejercicios

PARA conocer todos los ejercicios existentes en el software de apoyo

Observaciones:

La funcionalidad de listar todos los ejercicios existentes estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Estos usuarios tendrán acceso a una vista completa que incluirá todos los ejercicios almacenados en el sistema, independientemente de su estado (Activo o Inactivo), permitiendo así una gestión integral del contenido académico.

Criterios de aceptación:

- Mostrar la página de "Gestionar Contenido".
 - Presentar todos los ejercicios existentes por su título en la tabla "Ejercicios".
 - Diferenciar visualmente los ejercicios inactivos mediante celdas con un color rojo distintivo, facilitando su identificación respecto a los ejercicios activos.
 - Habilitar la selección de cualquier ejercicio dentro de la tabla "Ejercicios" mediante interacción del docente.
 - Resaltar el ejercicio seleccionado por el docente con un fondo de color gris, proporcionando una indicación visual clara de la opción elegida.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0018

Nombre: Registrar ejercicios

Usuario: Docente

Iteración asignada: 3

Referencia de requisito: RF018

Prioridad en el negocio
(Alta/Media/Baja): Alta

Fecha de inicio: 06/12/2023

Riesgo en el desarrollo
(Alta/Media/Baja): Alta

Fecha de entrega: 24/12/2023

Descripción:

COMO docente

QUIERO registrar ejercicios

PARA agregar nuevos ejercicios al software de apoyo

Observaciones:

La capacidad de registrar nuevos ejercicios en el sistema estará exclusivamente reservada para usuarios autenticados con el rol de "Docente", garantizando así el control y la calidad del contenido académico ingresado.

Criterios de aceptación:

- Desplegar el formulario de registro al seleccionar la opción "Registrar ejercicio".
 - Validar que todos los campos obligatorios (título, instrucciones, restricciones, solución del ejercicio ejercicios y retroalimentación) estén completos antes de procesar el registro.
 - Verificar la unicidad del título ingresado en la base de datos para evitar duplicados.
 - Mostrar un mensaje de alerta claro si se detecta un título duplicado, informando al usuario que el ejercicio ya existe.
 - Registrar el ejercicio en la base de datos al hacer clic en el botón "Guardar", almacenando toda la información ingresada.
 - Mostrar un mensaje de confirmación indicando que el ejercicio ha sido creado con éxito.
 - Redirigir automáticamente al usuario a la vista de "Gestionar contenido" tras el registro exitoso.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0019

Nombre: Editar ejercicios

Usuario: Docente

Iteración Asignada: 3

Referencia de requisito: RF019

Prioridad en el negocio **Fecha de inicio:** 06/12/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 24/12/2023
(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO poder editar los ejercicios existentes

PARA actualizar y mantener al día la información en el software de apoyo

Observaciones:

La función de edición de ejercicios existentes estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Estos usuarios podrán modificar cualquier ejercicio almacenado en el sistema, independientemente de su estado actual (activo o inactivo), permitiendo así una gestión flexible y completa del contenido académico.

Criterios de aceptación:

- Permitir la selección de un ejercicio existente desde la tabla "Ejercicios".
 - Habilitar el botón "Editar ejercicio" una vez seleccionado un ejercicio.
 - Desplegar el formulario de edición con los datos del ejercicio seleccionado.
 - Habilitar la modificación de los campos: título, instrucciones, restricciones, solución del ejercicio ejercicios y retroalimentación.
 - Validar que todos los campos requeridos estén completos antes de guardar.
 - Verificar que el título no exceda los 255 caracteres.
 - Comprobar que el título modificado no exista ya en la base de datos.
 - Mostrar un mensaje de alerta si se detecta un título duplicado.
 - Actualizar la información del ejercicio al presionar el botón "Guardar".
 - Mostrar un mensaje de confirmación indicando que el ejercicio ha sido modificado con éxito.
 - Redirigir automáticamente al usuario a la vista de "Gestionar contenido" tras la edición exitosa.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0020

Nombre de Historia de Usuario: Historial de cambios por ejercicio

Usuario: Docente

Iteración Asignada: 3

Referencia de requisito: RF020

Prioridad en el negocio **Fecha de inicio:** 06/12/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 24/12/2023
(Alta/Media/Baja): Baja

Descripción:

COMO docente

QUIERO visualizar el historial de cambios de un ejercicio

PARA conocer cuándo y qué modificaciones se han realizado en el contenido

Observaciones:

La funcionalidad de ver el historial de cambios estará disponible exclusivamente para usuarios autenticados con el rol de "Docente". Esta función permitirá un seguimiento detallado de las modificaciones realizadas en cada ejercicio, proporcionando transparencia y trazabilidad en la gestión del contenido académico.

Criterios de aceptación:

- Mostrar la opción "Ver historial" junto a cada a la tabla "Ejercicios"
 - Al seleccionar "Ver historial", desplegar una lista cronológica de los cambios realizados en el ejercicio.
 - Presentar para cada entrada del historial: Fecha y hora del cambio realizado.
 - Ordenar el historial de cambios del más reciente al más antiguo.
 - Asegurar que el historial se actualice automáticamente cuando se realicen nuevos cambios en el ejercicio.
 - Mostrar un mensaje indicativo si el ejercicio no tiene cambios registrados.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0021 **Nombre de Historia de Usuario:**
Activar/Desactivar ejercicios

Usuario: Docente

Iteración Asignada: 3 **Referencia de requisito:** RF021

Prioridad en el negocio **Fecha de inicio:** 06/12/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 24/12/2023
(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO activar/desactivar ejercicios

PARA activar y desactivar ejercicios existentes en el software de apoyo

Observaciones:

La facultad de activar o desactivar ejercicios existentes en el sistema estará reservada exclusivamente para usuarios autenticados con el rol de "Docente". Esta restricción asegura un control preciso sobre la visibilidad y disponibilidad del contenido académico en el software de apoyo.

Criterios de aceptación:

- Listar todos los ejercicios existentes en la tabla "Ejercicios".
 - Diferenciar visualmente los ejercicios inactivos con celdas de color rojo.
 - Permitir la selección de cualquier ejercicio de la tabla "Ejercicios".
 - Resaltar el ejercicio seleccionado con un fondo gris.
 - Alternar el estado del ejercicio (activo/inactivo) al presionar el botón "Activar/Desactivar".
 - Actualizar inmediatamente la visualización del ejercicio en la tabla tras cambiar su estado.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0022

Nombre de Historia de Usuario: Buscar ejercicios

Usuario: Docente

Iteración Asignada: 3

Referencia de requisito: RF022

Prioridad en el negocio **Fecha de inicio:** 06/12/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 24/12/2023
(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO buscar ejercicios

PARA buscar ejercicios existentes en el software de apoyo independiente de su estado (Activo/Inactivo)

Observaciones:

La funcionalidad de búsqueda de ejercicios estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Esta medida garantiza un acceso controlado a la información académica, permitiendo a los docentes localizar eficientemente contenidos específicos dentro del sistema.

Criterios de aceptación:

- Presentar todos los ejercicios existentes en la tabla "Ejercicios".
 - Proporcionar un campo de búsqueda donde el usuario pueda ingresar el título del ejercicio deseado.
 - Realizar la búsqueda automáticamente mientras el usuario escribe el título del ejercicio.
 - Mostrar un mensaje informativo si no se encuentra ningún ejercicio con el título buscado.
 - Mantener visible la lista completa de ejercicios, con el resultado de la búsqueda destacado.
 - Permitir borrar el texto del campo de búsqueda y reiniciar la visualización de todos los ejercicios.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0023

Nombre de Historia de Usuario: Visualizar ejercicios

Usuario: Administrador, Docente, Estudiante

Iteración Asignada: 3

Referencia de requisito: RF023

Prioridad en el negocio **Fecha de inicio:** 06/12/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 24/12/2023
(Alta/Media/Baja): Media

Descripción:

COMO usuario del sistema (docente, administrador o estudiante)

QUIERO acceder al contenido completo de los ejercicios activos

PARA profundizar en aspectos específicos del material educativo disponible en el software de apoyo

Observaciones:

Esta funcionalidad estará disponible para todos los usuarios autenticados en el sistema, independientemente de su rol. Proporcionará acceso al contenido detallado de los ejercicios activos dentro de cada subtema, permitiendo a los usuarios explorar y utilizar material educativo más específico y detallado. Esta opción mostrará el contenido completo de los ejercicios, facilitando un aprendizaje más profundo y estructurado.

Criterios de aceptación:

- Mostrar la sección " Ejercicios" debajo del contenido del subtema seleccionado.
 - Presentar una lista de todos los ejercicios activos, mostrando sus títulos.
 - Permitir al usuario seleccionar un ejercicio específico para ver su contenido detallado.
 - Al seleccionar un ejercicio, mostrar su contenido completo, incluyendo: título, instrucciones, restricciones, solución del ejercicio ejercicios y retroalimentación.
 - Incluir indicadores visuales de progreso mostrando el avance del usuario en los subtemas.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0024

Nombre de Historia de Usuario:

Proporcionar retroalimentación de ejercicios

Usuario: Estudiante

Iteración Asignada: 3

Referencia de requisito: RF024

Prioridad en el negocio **Fecha de inicio:** 06/12/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 24/12/2023
(Alta/Media/Baja): Media

Descripción:

COMO estudiante

QUIERO acceder a la solución codificada de los ejercicios y su retroalimentación

PARA comprender mejor la resolución y mejorar mi aprendizaje en programación

Observaciones:

Esta funcionalidad estará disponible exclusivamente para usuarios autenticados con el rol de "Estudiante". Permitirá a los estudiantes revisar las soluciones correctas de los ejercicios de programación, junto con explicaciones detalladas, facilitando un aprendizaje más profundo y autónomo.

Criterios de aceptación:

- Proporcionar un entorno dentro del software de apoyo donde el estudiante pueda desarrollar el ejercicio.
 - Mostrar un botón "Mostrar Solución" junto a cada ejercicio, disponible en todo momento.
 - Al presionar "Mostrar Solución", al estudiante se le mostrará una solución codificada en un IDE integrado en el software y a la vez una retroalimentación del ejercicio.
 - Permitir al estudiante ejecutar y validar el código dentro del IDE.
 - Mostrar una explicación detallada de la solución y conceptos clave del ejercicio.
-



Cuarta iteración

Historia de usuario	
Número: 0025	Nombre: Listar preguntas de control
Usuario: Docente	
Iteración asignada: 4	Referencia de requisito: RF025
Prioridad en el negocio (Alta/Media/Baja): Media	Fecha de inicio: 19/12/2023
Riesgo en el desarrollo (Alta/Media/Baja): Media	Fecha de entrega: 05/01/2024
Descripción: COMO docente QUIERO listar preguntas de control PARA conocer todas las preguntas de control existentes en el software de apoyo	
Observaciones: La funcionalidad de listar todas las preguntas de control existentes estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Estos usuarios tendrán acceso a una vista completa que incluirá todas las preguntas de control almacenadas en el sistema, independientemente de su estado (Activo o Inactivo), permitiendo así una gestión integral del contenido académico.	
Criterios de aceptación: <ul style="list-style-type: none">• Mostrar la página de "Gestionar Contenido".• Presentar todas las preguntas de control existentes por su título en la tabla "Preguntas de control".• Diferenciar visualmente las preguntas de control inactivas mediante celdas con un color rojo distintivo, facilitando su identificación respecto a las preguntas de control activas.• Habilitar la selección de cualquier pregunta de control dentro de la tabla "Preguntas de control " mediante interacción del docente.• Resaltar la pregunta de control seleccionada por el docente con un fondo de color gris, proporcionando una indicación visual clara de la opción elegida.	



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0026 **Nombre:** Registrar preguntas de control

Usuario: Docente

Iteración asignada: 4

Referencia de requisito: RF026

Prioridad en el negocio
(Alta/Media/Baja): Alta

Fecha de inicio: 19/12/2023

Riesgo en el desarrollo
(Alta/Media/Baja): Alta

Fecha de entrega: 05/01/2024

Descripción:

COMO docente

QUIERO registrar preguntas de control

PARA agregar nuevas preguntas de control al software de apoyo

Observaciones:

La capacidad de registrar nuevas preguntas de control en el sistema estará exclusivamente reservada para usuarios autenticados con el rol de "Docente", garantizando así el control y la calidad del contenido académico ingresado.

Criterios de aceptación:

- Desplegar el formulario de registro al seleccionar la opción "Registrar pregunta de control".
 - Validar que todos los campos obligatorios (enunciado, opciones, respuesta correcta y justificación de la respuesta) estén completos antes de procesar el registro.
 - Verificar la unicidad del título ingresado en la base de datos para evitar duplicados.
 - Mostrar un mensaje de alerta claro si se detecta un título duplicado, informando al usuario que la pregunta de control ya existe.
 - Registrar la pregunta de control en la base de datos al hacer clic en el botón "Guardar", almacenando toda la información ingresada.
 - Mostrar un mensaje de confirmación indicando que la pregunta de control ha sido creada con éxito.
 - Redirigir automáticamente al usuario a la vista de "Gestionar contenido" tras el registro exitoso.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0027

Nombre: Editar preguntas de control

Usuario: Docente

Iteración Asignada: 4

Referencia de requisito: RF027

Prioridad en el negocio **Fecha de inicio:** 19/12/2023

(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 05/01/2024

(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO poder editar las preguntas de control existentes

PARA actualizar y mantener al día la información en el software de apoyo

Observaciones:

La función de edición de preguntas de control existentes estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Estos usuarios podrán modificar cualquier pregunta de control almacenada en el sistema, independientemente de su estado actual (activo o inactivo), permitiendo así una gestión flexible y completa del contenido académico.

Criterios de aceptación:

- Permitir la selección de una pregunta de control existente desde la tabla "Preguntas de control".
 - Habilitar el botón "Editar pregunta de control " una vez seleccionado una pregunta de control.
 - Desplegar el formulario de edición con los datos de la pregunta de control seleccionada.
 - Habilitar la modificación de los campos: enunciado, opciones, respuesta correcta y justificación de la respuesta
 - Validar que todos los campos requeridos estén completos antes de guardar.
 - Comprobar que el enunciado modificado no exista ya en la base de datos.
 - Mostrar un mensaje de alerta si se detecta un enunciado duplicado.
 - Actualizar la información de la pregunta de control al presionar el botón "Guardar".
 - Mostrar un mensaje de confirmación indicando que la pregunta de control ha sido modificada con éxito.
 - Redirigir automáticamente al usuario a la vista de "Gestionar contenido" tras la edición exitosa.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0028

Nombre de Historia de Usuario: Historial de cambios por pregunta de control

Usuario: Docente

Iteración Asignada: 4

Referencia de requisito: RF028

Prioridad en el negocio **Fecha de inicio:** 19/12/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 05/01/2024
(Alta/Media/Baja): Baja

Descripción:

COMO docente

QUIERO visualizar el historial de cambios de una pregunta de control

PARA conocer cuándo y qué modificaciones se han realizado en el contenido

Observaciones:

La funcionalidad de ver el historial de cambios estará disponible exclusivamente para usuarios autenticados con el rol de "Docente". Esta función permitirá un seguimiento detallado de las modificaciones realizadas en cada pregunta de control, proporcionando transparencia y trazabilidad en la gestión del contenido académico.

Criterios de aceptación:

- Mostrar la opción "Ver historial" junto a cada a la tabla "Preguntas de control"
 - Al seleccionar "Ver historial", desplegar una lista cronológica de los cambios realizados en la pregunta de control.
 - Presentar para cada entrada del historial: Fecha y hora del cambio realizado.
 - Ordenar el historial de cambios del más reciente al más antiguo.
 - Asegurar que el historial se actualice automáticamente cuando se realicen nuevos cambios en la pregunta de control.
 - Mostrar un mensaje indicativo si la pregunta de control no tiene cambios registrados.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0029

Nombre de Historia de Usuario:

Activar/Desactivar preguntas de control

Usuario: Docente

Iteración Asignada: 4

Referencia de requisito: RF029

Prioridad en el negocio **Fecha de inicio:** 19/12/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 05/01/2024
(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO activar/desactivar preguntas de control

PARA activar y desactivar preguntas de control existentes en el software de apoyo

Observaciones:

La facultad de activar o desactivar preguntas de control existentes en el sistema estará reservada exclusivamente para usuarios autenticados con el rol de "Docente". Esta restricción asegura un control preciso sobre la visibilidad y disponibilidad del contenido académico en el software de apoyo.

Criterios de aceptación:

- Listar todas las preguntas de control existentes en la tabla "Preguntas de control".
 - Diferenciar visualmente las preguntas de control inactivas con celdas de color rojo.
 - Permitir la selección de cualquier pregunta de control de la tabla "Preguntas de control".
 - Resaltar la pregunta de control seleccionada con un fondo gris.
 - Alternar el estado de la pregunta de control (activo/inactivo) al presionar el botón "Activar/Desactivar".
 - Actualizar inmediatamente la visualización de la pregunta de control en la tabla tras cambiar su estado.
-



Historia de usuario

Número: 0030

Nombre de Historia de Usuario: Buscar preguntas de control

Usuario: Docente

Iteración Asignada: 4

Referencia de requisito: RF030

Prioridad en el negocio **Fecha de inicio:** 19/12/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 05/01/2024
(Alta/Media/Baja): Media

Descripción:

COMO docente

QUIERO buscar preguntas de control

PARA buscar preguntas de control existentes en el software de apoyo independiente de su estado (Activo/Inactivo)

Observaciones:

La funcionalidad de búsqueda de preguntas de control estará restringida exclusivamente a usuarios autenticados con el rol de "Docente". Esta medida garantiza un acceso controlado a la información académica, permitiendo a los docentes localizar eficientemente contenidos específicos dentro del sistema.

Criterios de aceptación:

- Presentar todas las preguntas de control existentes en la tabla "Preguntas de control".
 - Proporcionar un campo de búsqueda donde el usuario pueda ingresar el enunciado de la pregunta de control deseada.
 - Realizar la búsqueda automáticamente mientras el usuario escribe el enunciado de la pregunta de control.
 - Mostrar un mensaje informativo si no se encuentra ninguna pregunta de control con el enunciado buscado.
 - Mantener visible la lista completa de preguntas de control, con el resultado de la búsqueda destacado.
 - Permitir borrar el texto del campo de búsqueda y reiniciar la visualización de todas las preguntas de control.
-



Historia de usuario

Número: 0031	Nombre de Historia de Usuario: Visualizar preguntas de control
Usuario: Administrador, Docente, Estudiante	
Iteración Asignada: 4	Referencia de requisito: RF031
Prioridad en el negocio (Alta/Media/Baja): Media	Fecha de inicio: 19/12/2023
Riesgo en el desarrollo (Alta/Media/Baja): Media	Fecha de entrega: 05/01/2024

Descripción:

COMO usuario del sistema (docente, administrador o estudiante)
QUIERO acceder al contenido completo de las preguntas de control activas
PARA profundizar en aspectos específicos del material educativo disponible en el software de apoyo

Observaciones:

Esta funcionalidad estará disponible para todos los usuarios autenticados en el sistema, independientemente de su rol. Proporcionará acceso al contenido detallado de las preguntas de control activas dentro de cada ejercicio, permitiendo a los usuarios explorar y utilizar material educativo más específico y detallado. Esta opción mostrará el contenido completo de las preguntas de control, facilitando un aprendizaje más profundo y estructurado.

Criterios de aceptación:

- Mostrar la sección " Preguntas de control" debajo del contenido del ejercicio seleccionado.
 - Presentar una lista de todas las preguntas de control activas, mostrando sus enunciados.
 - Permitir al usuario seleccionar una pregunta de control específica para ver su contenido detallado.
 - Al seleccionar una pregunta de control, mostrar su contenido completo, incluyendo: enunciado, opciones, respuesta correcta y justificación de la respuesta
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0032

Nombre de Historia de Usuario: Responder pregunta de control

Usuario: Estudiante

Iteración Asignada: 4

Referencia de requisito: RF032

Prioridad en el negocio **Fecha de inicio:** 19/12/2023
(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 05/01/2024
(Alta/Media/Baja): Media

Descripción:

COMO estudiante

QUIERO responder preguntas de control de conocimientos en cada ejercicio

PARA reforzar mi aprendizaje y ver mi progreso incrementarse

Observaciones:

Esta funcionalidad estará disponible para usuarios autenticados con el rol de "Estudiante". Permitirá a los estudiantes evaluar su comprensión de los conceptos relacionados con cada ejercicio mediante preguntas de control, y al mismo tiempo, visualizar cómo su progreso aumenta al responder correctamente.

Criterios de aceptación:

- Presentar un conjunto de preguntas de control de conocimientos después de cada ejercicio práctico.
 - Mostrar las preguntas de control en un formato claro y fácil de entender (opción múltiple).
 - Proporcionar retroalimentación inmediata después de enviar la respuesta correcta de las preguntas de control.
 - Incrementar visiblemente el indicador de progreso del ejercicio basado en el porcentaje de respuestas correctas.
 - Asegurar que las preguntas de control estén directamente relacionadas con los conceptos abordados en el ejercicio correspondiente.
-



Quinta iteración

Historia de usuario

Número: 0033

Nombre: Listar comentarios

Usuario: Administrador, Docente, Estudiante

Iteración asignada: 5

Referencia de requisito: RF033

Prioridad en el negocio

Fecha de inicio: 15/01/2024

(Alta/Media/Baja): Media

Riesgo en el desarrollo

Fecha de entrega: 06/02/2024

(Alta/Media/Baja): Media

Descripción:

COMO usuario del sistema (docente, administrador o estudiante)

QUIERO listar comentarios

PARA revisar las discusiones y opiniones relacionadas con los ejercicios

Observaciones:

Esta funcionalidad estará disponible para todos los usuarios autenticados en el sistema.

Permitirá a los usuarios ver una lista completa de todos los comentarios realizados en los ejercicios, facilitando el seguimiento de discusiones y el acceso a información adicional proporcionada por la comunidad.

Criterios de aceptación:

- Mostrar la página de "Gestionar Contenido".
 - Presentar una lista ordenada de todos los comentarios existentes por ejercicio.
 - Presentar cada comentario con la siguiente información: Nombre del usuario y contenido.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0034

Nombre: Registrar comentarios

Usuario: Administrador, Docente, Estudiante

Iteración asignada: 5

Referencia de requisito: RF034

Prioridad en el negocio
(Alta/Media/Baja): Alta

Fecha de inicio: 15/01/2024

Riesgo en el desarrollo
(Alta/Media/Baja): Alta

Fecha de entrega: 06/02/2024

Descripción:

COMO usuario del sistema (docente, administrador o estudiante)

QUIERO crear nuevos comentarios relacionados con ejercicios existentes

PARA compartir mis ideas, preguntas o reflexiones sobre el contenido

Observaciones:

Esta funcionalidad estará disponible para todos los usuarios autenticados en el sistema. Permitirá a los usuarios añadir comentarios a ejercicios específicos, fomentando la interacción y el intercambio de conocimientos entre la comunidad de aprendizaje.

Criterios de aceptación:

- Mostrar una opción "Agregar comentario" junto a cada ejercicio existente.
 - Al seleccionar "Agregar comentario", desplegar un campo de texto para ingresar la descripción del comentario.
 - Requerir que el campo de descripción del comentario no esté vacío para poder enviarlo.
 - Implementar un botón "Publicar" para enviar el comentario una vez completado.
 - Mostrar una confirmación visual cuando el comentario se haya publicado exitosamente.
 - Agregar el nuevo comentario a la lista de comentarios del ejercicio, mostrándolo inmediatamente sin necesidad de recargar la página.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0035

Nombre: Editar comentarios

Usuario: Administrador, Docente, Estudiante

Iteración Asignada: 5

Referencia de requisito: RF035

Prioridad en el negocio **Fecha de inicio:** 15/01/2024

(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 06/02/2024

(Alta/Media/Baja): Media

Descripción:

COMO usuario del sistema (docente, administrador o estudiante)

QUIERO editar la descripción de mis comentarios existentes

PARA corregir, actualizar o mejorar el contenido de mis aportaciones

Observaciones:

Esta funcionalidad estará disponible para usuarios autenticados que sean autores de los comentarios. Permitirá a los usuarios modificar el contenido de sus comentarios previos, manteniendo la integridad de las discusiones mientras se permite la corrección y mejora de las aportaciones.

Criterios de aceptación:

- Mostrar una opción "Editar" junto a cada comentario creado por el usuario actual.
 - Al seleccionar "Editar", transformar el texto del comentario en un campo editable.
 - Permitir al usuario modificar la descripción del comentario.
 - Validar que no existan campos vacíos.
 - Implementar un botón "Guardar cambios" para aplicar las modificaciones al comentario.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0036

Nombre: Visualizar comentarios

Usuario: Administrador, Docente, Estudiante

Iteración Asignada: 5

Referencia de requisito: RF036

Prioridad en el negocio **Fecha de inicio:** 15/01/2024

(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 06/02/2024

(Alta/Media/Baja): Media

Descripción:

COMO usuario del sistema (docente, administrador o estudiante)

QUIERO ver todos los comentarios asociados a cada ejercicio

PARA acceder a las discusiones, ideas y retroalimentación relacionadas con el contenido específico

Observaciones:

Esta funcionalidad estará disponible para todos los usuarios autenticados en el sistema. Permitirá a los usuarios visualizar todos los comentarios relacionados con un ejercicio en particular, facilitando el acceso a información adicional y promoviendo el aprendizaje colaborativo.

Criterios de aceptación:

- Mostrar una sección de "Comentarios" claramente visible junto a cada ejercicio en el software de apoyo.
 - Listar todos los comentarios asociados al ejercicio específico en orden cronológico, con los más recientes primero.
 - Presentar cada comentario con la siguiente información: Nombre del usuario que la lo creó y contenido del comentario.
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0037

Nombre: Visualizar progreso

Usuario: Estudiante

Iteración Asignada: 5

Referencia de requisito: RF037

Prioridad en el negocio **Fecha de inicio:** 15/01/2024

(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 06/02/2024

(Alta/Media/Baja): Media

Descripción:

COMO estudiante

QUIERO visualizar una vista detallada y actualizada de mi progreso de aprendizaje PARA conocer mi avance en temas, subtemas y ejercicios completados en el software de apoyo

Observaciones:

Esta funcionalidad estará disponible para usuarios autenticados con el rol de "Estudiante". Proporcionará una visión integral del progreso individual, permitiendo a los estudiantes monitorear su aprendizaje e identificar áreas que requieren más atención.

Criterios de aceptación:

- Presentar un resumen general del progreso total del estudiante en porcentaje.
 - Desglosar el progreso por temas principales, mostrando: Nombre del tema, Porcentaje de avance en cada tema
 - Para cada tema, permitir expandir la vista para mostrar el progreso en subtemas: Nombre del subtema, Porcentaje de avance en cada subtema
 - Dentro de cada subtema, listar los ejercicios completados y pendientes: Nombre o número del ejercicio, Porcentaje de avance en cada ejercicio
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0038

Nombre: Cambiar periodo académico

Usuario: Administrador

Iteración Asignada: 5

Referencia de requisito: RF038

Prioridad en el negocio **Fecha de inicio:** 15/01/2024

(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 06/02/2024

(Alta/Media/Baja): Media

Descripción:

COMO administrador del sistema

QUIERO gestionar el cambio de periodo académico

PARA reiniciar el progreso de los usuarios, respaldar datos anteriores y acceder a reportes por periodo

Observaciones:

Esta funcionalidad estará disponible exclusivamente para usuarios con rol de administrador. Facilitará la transición entre periodos académicos, asegurando la preservación de datos históricos y permitiendo un análisis comprensivo del rendimiento de los usuarios a lo largo del tiempo.

Criterios de aceptación:

- Proporcionar una opción clara en el panel de administración para "Cambiar Periodo Académico".
 - Mostrar una advertencia y solicitar confirmación antes de proceder con el cambio de periodo, detallando las consecuencias de la acción.
 - Al confirmar el cambio: a) Reiniciar automáticamente el progreso de todos los usuarios en el sistema. b) Generar una copia de seguridad de los progresos de los usuarios del periodo que finaliza. c) Crear un nuevo registro para el periodo académico que inicia.
 - Generar automáticamente un reporte del periodo académico que finalizo, incluyendo: Estadísticas generales de progreso de los usuarios en temas, subtemas y ejercicios.
 - Implementar una interfaz para acceder a los reportes de periodos académicos anteriores, con opciones de filtrado por periodo académico.
-



UNL

Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0039

Nombre: Temas destacados

Usuario: Administrador, Docente, Estudiante

Iteración Asignada: 5

Referencia de requisito: RF039

Prioridad en el negocio **Fecha de inicio:** 15/01/2024

(Alta/Media/Baja): Media

Riesgo en el desarrollo **Fecha de entrega:** 06/02/2024

(Alta/Media/Baja): Media

Descripción:

COMO usuario del sistema

QUIERO poder valorar positivamente los temas y ver los más populares

PARA interactuar con el contenido y acceder rápidamente a los temas mejor recibidos por la comunidad

Observaciones:

Esta funcionalidad estará disponible para todos los usuarios autenticados, independientemente de su rol. Fomentará la interacción con el contenido y proporcionará una forma rápida de identificar los temas más valorados por la comunidad.

Criterios de aceptación:

- Mostrar un botón de "Me gusta" junto a cada tema en el sistema.
 - Permitir a cualquier usuario autenticado hacer clic en el botón "Me gusta" para valorar positivamente un tema.
 - Actualizar visualmente el estado del botón "Me gusta" inmediatamente después de la interacción del usuario.
 - Mostrar el número total de "Me gusta" para cada tema.
 - Crear un dashboard en la página de inicio del software que muestre los temas más populares.
 - Ordenar los temas en el dashboard basándose en el número de "Me gusta" recibidos.
 - Limitar el dashboard a un número específico de temas (por ejemplo, los 10 más populares).
-



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

Historia de usuario

Número: 0040

Nombre: Notificaciones por email de nuevo contenido

Usuario: Administrador, Docente, Estudiante

Iteración Asignada: 5

Referencia de requisito: RF040

Prioridad en el negocio **Fecha de inicio:** 15/01/2024
(Alta/Media/Baja): Alta

Riesgo en el desarrollo **Fecha de entrega:** 06/02/2024
(Alta/Media/Baja): Media

Descripción:

COMO usuario registrado

QUIERO recibir notificaciones por correo electrónico sobre nuevo contenido

PARA estar informado de las actualizaciones y participar activamente en el software de apoyo

Observaciones:

Esta funcionalidad se activará automáticamente cuando un docente cree nuevo contenido en el sistema. Está diseñada para mantener a todos los usuarios informados y fomentar su participación continua en la plataforma.

Criterios de aceptación:

- Detectar automáticamente la creación de nuevo contenido (tema, subtema o ejercicio) por parte de un docente.
 - Generar una notificación por correo electrónico inmediatamente después de la creación del nuevo contenido.
 - Enviar la notificación a todos los usuarios registrados en el sistema.
 - Incluir en el correo electrónico: Título del nuevo contenido, Tipo de contenido (tema, subtema o ejercicio), Nombre del docente que lo creó
-



Universidad
Nacional
de Loja


UNIVERSIDAD NACIONAL DE LOJA




Carrera de Ingeniería en
Sistemas / Computación

Firmas de los docentes participantes:

Docente	Firma	Fecha
Ing. Wilman Chamba Mg. Sc.	 Firma electrónicamente por: WILMAN PATRICIO CHAMBA ZARAGOCIN	13 de noviembre del 2023

Docente	Firma	Fecha
Ing. Edison Coronel Mg. Sc.	 Firma electrónicamente por: EDISON LEONARDO CORONEL ROMERO	13 de noviembre del 2023

Docente	Firma	Fecha
Ing. René Guamán Mg. Sc.	 Firma electrónicamente por: EDWIN RENE GUAMAN QUINCHE	13 de noviembre del 2023

Anexo 4: Acta de socialización del temario



UNIVERSIDAD NACIONAL DE LOJA
FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES
NO RENOVABLES
CARRERA EN COMPUTACIÓN
ACTA DE SOCIALIZACIÓN DE TEMARIO DEL PROYECTO DE TIC

En la ciudad de Loja, a 24 de enero de 2024, el estudiante JUAN CARLOS ARMIJOS SÁNCHEZ deja constancia de la socialización realizada con los docentes del área de programación de la carrera de Computación. Durante la sesión, se presentó el temario que abarcará el software especializado en el aprendizaje de programación en C. Se detallaron tanto los temas principales como los subtemas correspondientes.

Es importante destacar que los temas socializados fueron extraídos del actual sílabo del primer ciclo del periodo académico Octubre-Marzo 2023. Además, se incorporaron elementos obtenidos de entrevistas realizadas a los propios docentes. Durante la presentación, se informó a los docentes que tendrán la capacidad de CREAR, EDITAR y ACTIVAR/DESACTIVAR el contenido en el software de apoyo de acuerdo con sus necesidades particulares.

Para constancia firma el autor del TIC y los docentes a los que se les socializo el contenido.

Loja, 24 de enero del 2024

Juan Carlos Armijos Sánchez
ESTUDIANTE CIS-UNL



FIRMADO DIGITALMENTE POR:
EDISON LEONARDO
CORONEL ROMERO

Ing. Edison Leonardo Coronel Romero MSc.
DOCENTE CIS-UNL



UNIVERSIDAD NACIONAL DE LOJA
FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES
NO RENOVABLES
CARRERA EN COMPUTACIÓN
ACTA DE SOCIALIZACIÓN DE TEMARIO DEL PROYECTO DE TIC

En la ciudad de Loja, a 24 de enero de 2024, el estudiante JUAN CARLOS ARMIJOS SÁNCHEZ deja constancia de la socialización realizada con los docentes del área de programación de la carrera de Computación. Durante la sesión, se presentó el temario que abarcará el software especializado en el aprendizaje de programación en C. Se detallaron tanto los temas principales como los subtemas correspondientes.

Es importante destacar que los temas socializados fueron extraídos del actual sílabo del primer ciclo del periodo académico Octubre-Marzo 2023. Además, se incorporaron elementos obtenidos de entrevistas realizadas a los propios docentes. Durante la presentación, se informó a los docentes que tendrán la capacidad de CREAR, EDITAR y ACTIVAR/DESACTIVAR el contenido en el software de apoyo de acuerdo con sus necesidades particulares.

Para constancia firma el autor del TIC y los docentes a los que se les socializo el contenido.

Loja, 24 de enero del 2024

Juan Carlos Armijos Sánchez
ESTUDIANTE CIS-UNL



Escaneado el documento por:
EDWIN RENE GUAMAN
QUINCHE

Ing. Edwin René Guamán Quinche MSc.
DOCENTE CIS-UNL



UNIVERSIDAD NACIONAL DE LOJA
FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES
NO RENOVABLES
CARRERA EN COMPUTACIÓN
ACTA DE SOCIALIZACIÓN DE TEMARIO DEL PROYECTO DE TIC

En la ciudad de Loja, a 24 de enero de 2024, el estudiante JUAN CARLOS ARMIJOS SÁNCHEZ deja constancia de la socialización realizada con los docentes del área de programación de la carrera de Computación. Durante la sesión, se presentó el temario que abarcará el software especializado en el aprendizaje de programación en C. Se detallaron tanto los temas principales como los subtemas correspondientes.

Es importante destacar que los temas socializados fueron extraídos del actual sílabo del primer ciclo del periodo académico Octubre-Marzo 2023. Además, se incorporaron elementos obtenidos de entrevistas realizadas a los propios docentes. Durante la presentación, se informó a los docentes que tendrán la capacidad de CREAR, EDITAR y ACTIVAR/DESACTIVAR el contenido en el software de apoyo de acuerdo con sus necesidades particulares.

Para constancia firma el autor del TIC y los docentes a los que se les socializo el contenido.

Loja, 24 de enero del 2024

Juan Carlos Armijos Sánchez
ESTUDIANTE CIS-UNL



Ing. Wilman Patricio Chamba Zaragocin MSc.
DOCENTE CIS-UNL

A continuación se presenta el temario socializado a los docentes:

Temario para el software de apoyo especializado en el aprendizaje de programación en C “C’ amigo”

Desarrollado por: Juan Carlos Armijos

A continuación, se presenta el temario que abarcará el software especializado en el aprendizaje de programación en C, detallando tanto los temas principales como los subtemas correspondientes. Cabe destacar que esta lista son los temas que el autor del TIC vio pertinentes y que los obtuvo mediante el silabo actual del primer ciclo y las entrevistas realizadas a los docentes del área de programación, ya que los docentes tendrán la capacidad de **crear, editar, activar y desactivar** contenido en el software de apoyo según sus necesidades.

Temas:

- **Lógica de programación**
 - Definición
 - Importancia de la lógica en el desarrollo de software
 - Descomposición de problemas y desarrollo de algoritmos
- **Proceso para resolver problemas a través de computadoras y algoritmos**
 - Pasos generales para abordar problemas de programación
 - Identificación y definición clara del problema
 - Dividir el problema en pasos más pequeños
- **Algoritmos**
 - Definición
 - Características
 - Clasificación.
- **Lenguaje de programación.**
 - Paradigma
 - Definición
 - Diferencia entre lenguaje de programación y PseudoCódigo
 - Clasificación.
- **Algoritmos Computacionales.**
 - Dato e información
 - Tipos de datos simples
 - Identificadores, variables y constantes
 - Instrucciones de control secuencial: asignación, lectura y escritura
- **Programación limpia**
 - Definición.
 - Beneficios de escribir código limpio.
 - Nombres significativos.
 - Identación y espaciados consistentes
 - Funciones y métodos claramente definidos
 - Evitar código duplicado
 - Simplicidad y claridad

- **Expresiones**
 - Aritméticas
 - Lógicas
 - Relacionales
 - Literales
- **Herramientas para programación.**
 - Entorno de desarrollo (IDE)
 - Compilador de C
 - Recursos de aprendizaje en línea.
- **Estructuras algorítmicas condicionales.**
 - Estructura condicional Simple
 - Estructura condicional Doble
 - Estructura condicional Múltiple
- **Estructuras algorítmicas repetitivas.**
 - Estructura Para (For)
 - Estructura Mientras (While)
 - Estructura Hacer ... Mientras (Do ... while)
- **Subprogramas**
 - Funciones
 - Procedimientos
 - Parámetros y argumentos
- **Modularidad**
 - Definición
 - Implementación en algoritmos computacionales
- **Arreglos**
 - Arreglos unidimensionales
 - Arreglos multidimensionales
 - Cadenas o Strings

Anexo 5: Implementación de la metodología de desarrollo



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Revisado y aprobado por:**
 - Ing. José Oswaldo Guamán, director del Trabajo de Integración Curricular.

Descripción: El presente anexo técnico describe la implementación sistemática de la metodología XP (Extreme Programming) en el desarrollo del sistema del Trabajo de Integración Curricular. La estructuración del proceso se fundamentó en un plan de iteración metódicamente establecido, el cual segmentó el desarrollo en cinco iteraciones diferenciadas. Cada una de estas iteraciones incorporó las fases fundamentales propias de la metodología XP: planificación estratégica, diseño arquitectónico, codificación estructurada y pruebas de validación.

Es pertinente señalar que la fase de lanzamiento se ejecutó de manera independiente a las iteraciones mencionadas, constituyendo un evento único de implementación en producción que culminó el ciclo de desarrollo. La adopción de este enfoque iterativo e incremental facilitó una adaptación dinámica a los requerimientos evolutivos del sistema, permitiendo así una optimización continua del software fundamentada en los resultados obtenidos en cada fase iterativa.

IMPLEMENTACIÓN DE LA METODOLOGÍA XP

1 Primera Iteración

1.1 Planificación

La **Tabla 1** muestra las actividades específicas a desarrollar en la primera iteración.

Tabla 1. Historias de Usuario a desarrollar en la primera iteración.

HU #	Historia de usuario	Fecha de creación	Fecha de entrega
HU001	Crear usuario	20/11/2023	12/12/2023
HU002	Editar usuario	25/11/2023	12/12/2023
HU003	Listar temas	26/11/2023	12/12/2023
HU004	Registrar temas	28/11/2023	12/12/2023
HU005	Editar temas	30/11/2023	12/12/2023
HU006	Historial de cambios por tema	01/12/2023	12/12/2023
HU007	Activar/Desactivar temas	06/12/2023	12/12/2023
HU008	Buscar temas	10/12/2023	12/12/2023
HU009	Visualizar temas	11/12/2023	12/12/2023

1.2 Diseño

La **Figura 1** muestra las funcionalidades de manera general a desarrollar en la primera iteración.

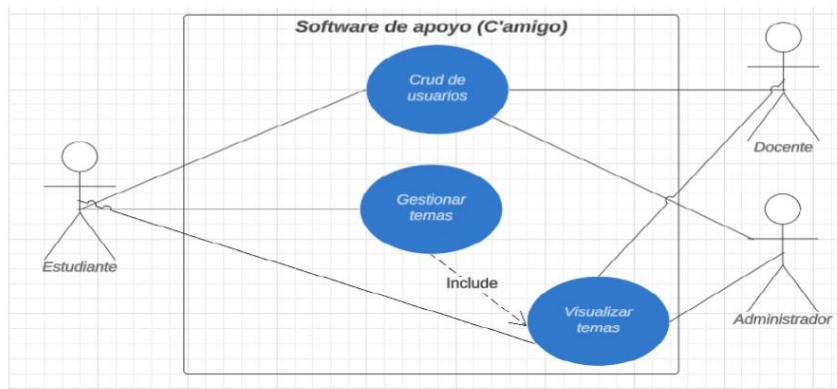


Figura 1. Casos de Uso general para la primera iteración.

1.2.1 Diagramas BPMN

A continuación, se presentan los diagramas BPMN pertenecientes a cada Historia de Usuario a desarrollar en la primera iteración.

1.2.1.1 HU001: Crear usuario

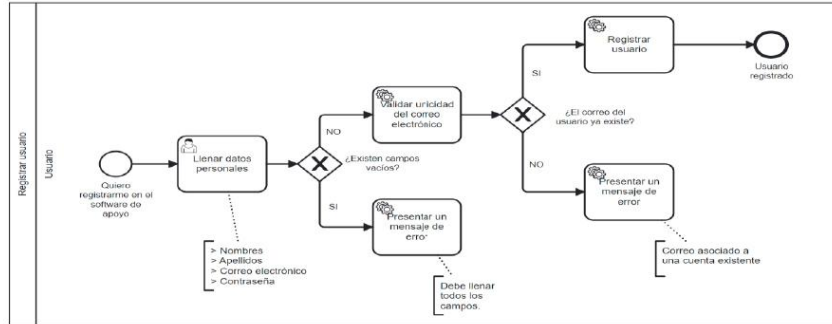


Figura 2. Diagrama BPMN de HU001.

1.2.1.2 HU002: Editar usuario

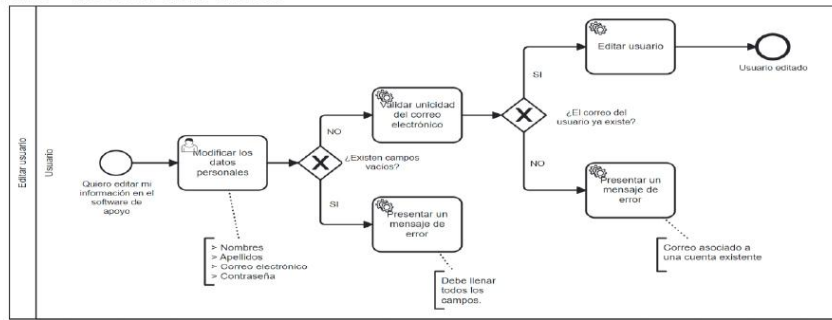


Figura 3. Diagrama BPMN de HU002.

1.2.1.3 HU003: Listar temas

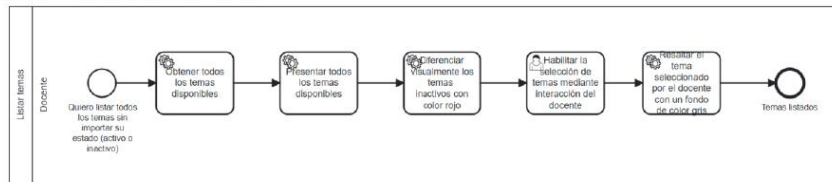


Figura 4. Diagrama BPMN de HU003

1.2.1.4 HU004: Registrar temas

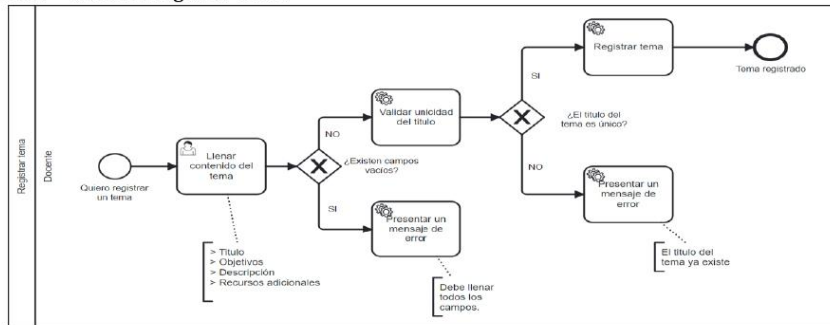


Figura 5. Diagrama BPMN de HU004.

1.2.1.5 HU005: Editar temas

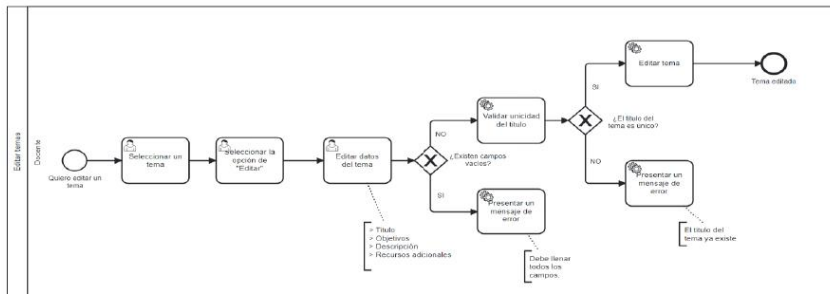


Figura 6. Diagrama BPMN de HU005.

1.2.1.6 HU006: Historial de cambios por tema

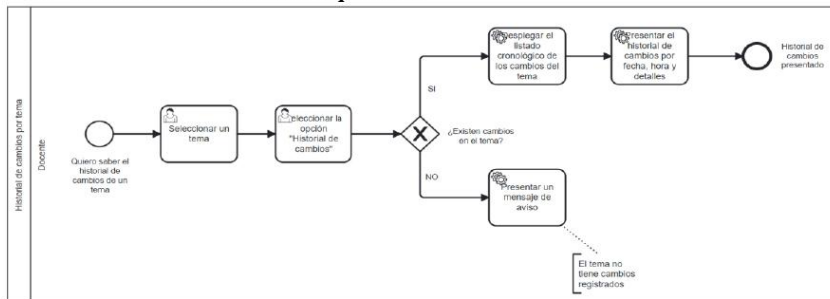


Figura 7. Diagrama BPMN de HU006.

1.2.1.7 HU007: Activar/Desactivar temas

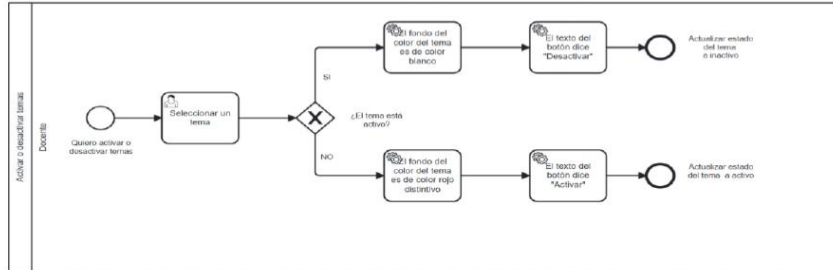


Figura 8. Diagrama BPMN de HU007.

1.2.1.8 HU008: Buscar temas

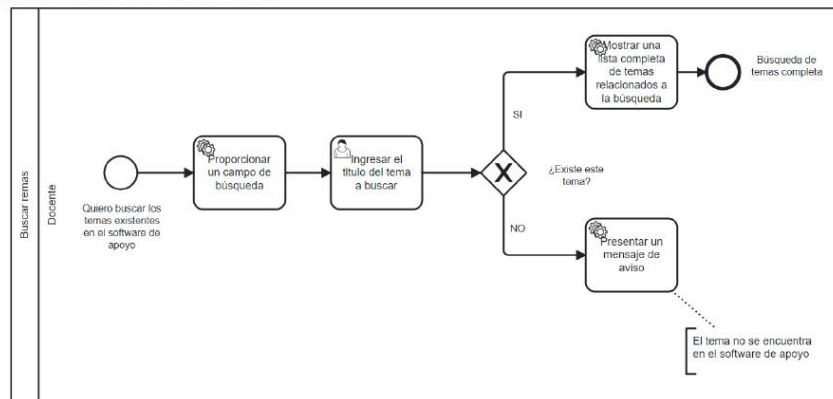


Figura 9. Diagrama BPMN de HU008.

1.2.1.9 HU009: Visualizar temas

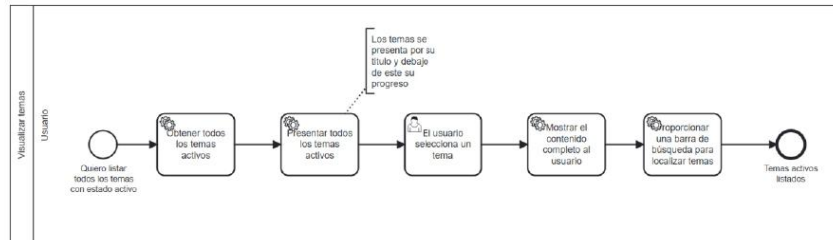


Figura 10. Diagrama BPMN de HU009.

1.2.2 Prototipos

A continuación, se presentan los prototipos pertenecientes a cada Historia de Usuario a desarrollar en la primera iteración.

1.2.2.1 HU001: Crear usuario



Prototipo de la interfaz de usuario para la creación de un usuario. El formulario se muestra en un navegador con la URL "https://". El título del formulario es "Registro de usuario:". El formulario contiene cuatro campos de entrada: "Nombres:", "Apellidos:", "Correo:" y "Contraseña:". Debajo de los campos hay un botón verde que dice "Registrarse".

Figura 11. Prototipo de HU001.

1.2.2.2 HU002: Editar usuario



Prototipo de la interfaz de usuario para editar el perfil de un usuario. El formulario se muestra en un navegador con la URL "https://". El título del formulario es "Editar perfil". El formulario contiene cinco campos de entrada: "Nombres:", "Apellidos:", "Correo:", "Contraseña actual:" y "Nueva contraseña:". Debajo de los campos hay un botón verde que dice "Validar contraseña". Debajo de los campos "Nueva contraseña:" y "Confirma la contraseña:" hay un botón rojo que dice "Cancelar" y un botón verde que dice "Guardar cambios".

Figura 12. Prototipo de HU002.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

1.2.2.3 HU003: Listar temas

Temas

Temas inactivos de rojo

Tema 1
Tema 2
Tema 3
Tema 4

Crear Editar Activar/Desactivar Historial de cambios

Figura 13. Prototipo de HU003.

1.2.2.4 HU004: Registrar temas

C-Compañero

DOCENTE

Docente OCT 2023 - FEB 2024

Crear temas

Título:

Objetivos:

Descripción:

Recursos adicionales:

Cancelar Guardar

Figura 14. Prototipo de HU004.

1.2.2.5 HU005: Editar temas



C-Compañero

DOCENTE

Docente OCT 2023 - FEB 2024

Editar tema

Título: Tema 1

Objetivos: Tema 1

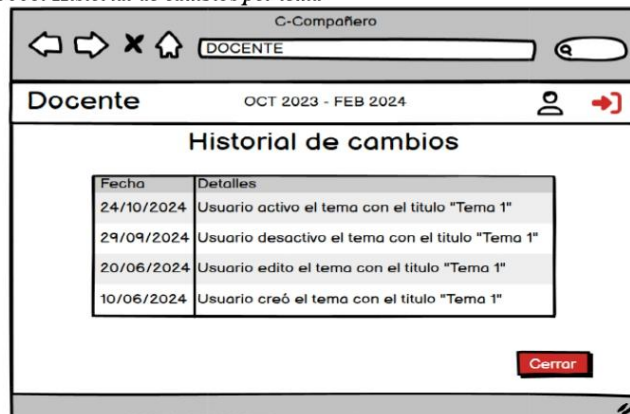
Descripción: Tema 1

Recursos adicionales: Tema 1

Cancelar Guardar

Figura 15. Prototipo de HU005.

1.2.2.6 HU006: Historial de cambios por tema



C-Compañero

DOCENTE

Docente OCT 2023 - FEB 2024

Historial de cambios

Fecha	Detalles
24/10/2024	Usuario activo el tema con el título "Tema 1"
29/09/2024	Usuario desactivo el tema con el título "Tema 1"
20/06/2024	Usuario edito el tema con el título "Tema 1"
10/06/2024	Usuario creó el tema con el título "Tema 1"

Cerrar

Figura 16. Prototipo de HU006.

1.2.2.7 HU007: Activar/Desactivar temas

Temas



Figura 17. Prototipo de HU007

1.2.2.8 HU008: Buscar temas

Temas



Figura 18. Prototipo de HU008.

1.2.2.9 HU009: Visualizar temas

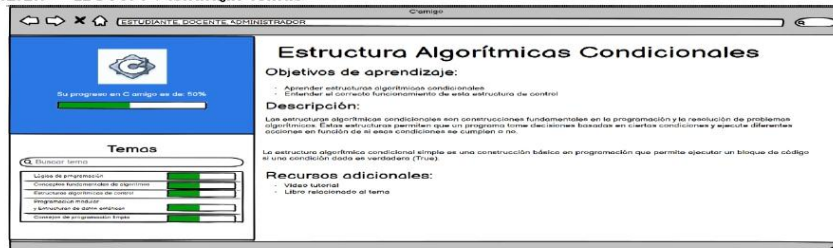


Figura 19. Prototipo de HU009.

1.3 Codificación

A continuación, se presenta los fragmentos principales del código fuente desarrollados en la primera iteración.



UNL

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

```

1  router.post('/registro', (req, res) => {
2    const { nombres, apellidos, email, clave, tipodni } = req.body;
3
4    // Verificar si el email ya está registrado
5    const verificarEmail = "SELECT * FROM usuarios WHERE email = ?";
6    sql.executeSQ(query(verificarEmail), [email], (resultado) => {
7      if (resultado.length > 0) {
8        return res.status(200).send({ msg: '-1', m: "El email ya está registrado" });
9      } else {
10         // Crear una nueva cuenta
11         const claveEncriptada = md5(clave);
12         const crearCuenta = "INSERT INTO usuarios (nombres, apellidos, email, clave) VALUES (?, ?, ?, ?)";
13         sql.executeSQ(query(crearCuenta), [nombres, apellidos, email, claveEncriptada], (resultadoCuenta) => {
14           // Obtener el ID de la cuenta recién creada
15           const idCuenta = resultadoCuenta.insertId;
16
17           // Crear un nuevo registro en la tabla persona
18           const crearPersona = "INSERT INTO persona (nombres, apellidos) VALUES (?, ?)";
19           sql.executeSQ(query(crearPersona), [nombres, apellidos], (resultadoPersona) => {
20             // Obtener el ID de la persona recién creada
21             const idPersona = resultadoPersona.insertId;
22
23             // Obtener el ID del rol basado en el tipo de rol proporcionado
24             const obtenerRol = "SELECT id FROM roles WHERE tipo = ?";
25             sql.executeSQ(query(obtenerRol), [tipodni], (resultadoRol) => {
26               if (resultadoRol.length > 0) {
27                 const idRol = resultadoRol[0].id;
28
29                 // Crear un nuevo usuario
30                 const crearUsuario = "INSERT INTO usuarios (nombre, id_usuario, id_usuario, id_usuario) VALUES (?, ?, ?, ?)";
31                 sql.executeSQ(query(crearUsuario), [nombres, idPersona, idRol, idCuenta], (resultadoUsuario) => {
32                   if (resultadoUsuario.affectedRows > 0) {
33                     const idUsuarioCreado = resultadoUsuario.insertId;
34                     // Segundo consulta para insertar en usuarios_tema
35
36                     const insertarTemasUsuario =
37                       "INSERT INTO usuarios_tema (id_usuario, idtema) SELECT ?, id FROM temas";
38                     sql.executeSQ(
39                       insertarTemasUsuario,
40                       [idUsuarioCreado],
41                       (resultadoTemasUsuario) => {
42                         // No verificamos afectados aquí, continuamos con la siguiente inserción
43                         const insertarSubtemasUsuario =
44                           "INSERT INTO usuarios_subtema (id_usuario, idsubtema) SELECT ?, id FROM subtema";
45                         sql.executeSQ(
46                           insertarSubtemasUsuario,
47                           [idUsuarioCreado],
48                           (resultadoSubtemasUsuario) => {
49                             // No verificamos afectados aquí, continuamos con la siguiente inserción
50                             const insertarServiciosUsuario =
51                               "INSERT INTO usuarios_servicio (id_usuario, idservicio) SELECT ?, id FROM servicios";
52                             sql.executeSQ(
53                               insertarServiciosUsuario,
54                               [idUsuarioCreado],
55                               (resultadoServiciosUsuario) => {
56                                 // No verificamos afectados aquí, continuamos con la última inserción
57                                 const insertarPreguntasUsuario =
58                                   "INSERT INTO usuarios_pregunta (id_usuario, idpregunta) SELECT ?, id FROM preguntas";
59                                 sql.executeSQ(
60                                   insertarPreguntasUsuario,
61                                   [idUsuarioCreado],
62                                   (resultadoPreguntasUsuario) => {
63                                     // Aquí hacemos verificación de afectados
64                                     return res.status(200).send({
65                                       msg: '1',
66                                       m: "Se registró correctamente el usuario",
67                                       id_usuario: idUsuarioCreado,
68                                     });
69                                   });
70                                 });
71                               });
72                             });
73                           });
74                         });
75                       });
76                     });
77                   } else {
78                     return res.status(200).send({ msg: '-1', m: "No se pudo crear el usuario" });
79                   }
80                 });
81               }
82             });
83           });
84         });
85       }
86     });
87   });
88 }
89
90 return res.status(200).send({ msg: '-1', m: "Tipo de rol no válido" });
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }

```

Figura 20. Codificación HU001.

```
1 router.post('/editarUsuario', (req, res) => {
2   const { userId, nombres, apellidos, email, clave } = req.body;
3   const usuarioEditado = { userId, nombres, apellidos, email, clave }
4
5   // Verificar si el usuario existe
6   const verificarUsuario = "SELECT * FROM usuario WHERE id = ?;";
7
8   sql.ejecutarResSQL(verificarUsuario, [userId], (resultadoUsuario) => {
9     if (resultadoUsuario.length === 0) {
10      return res.status(404).send({ en: -1, m: "Usuario no encontrado" });
11    }
12
13    // Actualizar la cuenta asociada al usuario, incluida la clave si se proporciona
14    let actualizarCuenta;
15    let parametrosCuenta;
16
17    if (clave) {
18      // Si se proporciona una nueva clave, actualizar también la clave
19      actualizarCuenta = "UPDATE cuenta SET email = ?, clave = ? WHERE id = ?;";
20      const claveEncriptada = md5(clave);
21      parametrosCuenta = [email, claveEncriptada, resultadoUsuario[0].cuenta_id];
22    } else {
23      // Si no se proporciona una nueva clave, solo actualizar el email
24      actualizarCuenta = "UPDATE cuenta SET email = ? WHERE id = ?;";
25      parametrosCuenta = [email, resultadoUsuario[0].cuenta_id];
26    }
27
28    sql.ejecutarResSQL(actualizarCuenta, parametrosCuenta, (resultadoCuenta) => {
29      // Actualizar la persona asociada al usuario
30      const actualizarPersona = "UPDATE persona SET nombres = ?, apellidos = ? WHERE id = ?;";
31      sql.ejecutarResSQL(actualizarPersona, [nombres, apellidos, resultadoUsuario[0].persona_id], (resultadoPersona) => {
32        return res.status(200).send({ en: 1, m: "Usuario actualizado exitosamente", usuarioEditado });
33      });
34    });
35  });
36 });
```

Figura 21. Codificación HU002.

```
1 router.post("/listarTemas", (req, res) => {
2   let obtenerTemas;
3   let idUsuario = req.body.idUsuario;
4   if (req.body.mensaje === "temasActivos") {
5     obtenerTemas =
6     "SELECT usuario_tema.idTema, usuario_tema.progreso, tema.titulo, tema.objetivos, tema.descripcion, tema.recursos, tema.estado " +
7     "FROM S(8D).usuario_tema " +
8     "INNER JOIN S(8D).tema ON usuario_tema.idTema = tema.id " +
9     "WHERE usuario_tema.idUsuario = ? AND tema.estado = 1; ";
10  } else {
11    obtenerTemas =
12    "SELECT usuario_tema.idTema, usuario_tema.progreso, tema.titulo, tema.objetivos, tema.descripcion, tema.recursos, tema.estado " +
13    "FROM S(8D).usuario_tema " +
14    "INNER JOIN S(8D).tema ON usuario_tema.idTema = tema.id " +
15    "WHERE usuario_tema.idUsuario = ?; ";
16  }
17  sql.ejecutarResSQL(obtenerTemas, [idUsuario], (resultado) => {
18    if (resultado.length > 0) {
19      return res
20        .status(200)
21        .send({ en: 1, m: "Temas obtenidos", temas: resultado });
22    } else {
23      return res.status(200).send({ en: -1, m: "No se encontraron temas" });
24    }
25  });
26 });
27
```

Figura 22. Codificación HU003.

```
1 router.post("/registrarTema", [], (req, res) => {
2   let titulo = req.body.titulo;
3   let objetivos = req.body.objetivos;
4   let descripcion = req.body.descripcion;
5   let recursos = req.body.recursos;
6   let estado = 1;
7
8   const registrarTema =
9     "INSERT INTO tema (titulo, objetivos, descripcion, recursos, estado) VALUES (?, ?, ?, ?, ?)";
10
11   sql.ejecutarResSQL(
12     registrarTema,
13     [titulo, objetivos, descripcion, recursos, estado],
14     (resultado) => {
15       if (resultado["affectedRows"] > 0) {
16         const idtemaInsertado = resultado["insertId"];
17
18         // Segunda consulta para insertar en usuario_tema
19         const insertarUsuarioTema =
20           "INSERT INTO usuario_tema (idusuario, idtema) SELECT id, ? FROM usuarios";
21
22         sql.ejecutarResSQL(
23           insertarUsuarioTema,
24           [idtemaInsertado],
25           (resultadoUsuarioTema) => {
26             if (resultadoUsuarioTema["affectedRows"] > 0) {
27               // Obtener correo de la tabla cuenta
28               const obtenerCorreos = "SELECT email FROM cuenta";
29
30               sql.ejecutarResSQL(
31                 obtenerCorreos,
32                 [],
33                 (resultadoCorreos) => {
34                   if (resultadoCorreos.length > 0) {
35                     const asunto = "Nuevo tema creado en C'amigo";
36                     const mensaje =
37                       `<h3>Se ha creado un nuevo tema en C'amigo: ${titulo} </h3>
38                       <p><strong>Objetivos </strong> ${objetivos}</p>
39                       <p>Accede a la plataforma para más detalles. </p>
40                       <p>Carrera de Computación, Universidad Nacional de Loja </p>
41                       <p>¡NO CONTESTAR ESTE CORREO </p>
42                       
43                     `;
44
45                     resultadoCorreos.forEach(cuenta => {
46                       sql.enviarCorreo(cuenta.email, asunto, mensaje);
47                     });
48
49                     return res.status(200).send({
50                       en: 1,
51                       m: "Se registró el tema con éxito y se enviaron correos de notificación",
52                       idtema: idtemaInsertado,
53                     });
54                   } else {
55                     return res.status(200).send({
56                       en: 1,
57                       m: "Se registró el tema con éxito, pero no hay cuentas para notificar",
58                       idtema: idtemaInsertado,
59                     });
60                   }
61                 });
62             } else {
63               return res.status(200).send({ en: -1, m: "No se pudo registrar en usuario_tema porque faltan temas o usuarios" });
64             }
65           }
66         );
67       } else {
68         return res.status(200).send({ en: -1, m: "No se pudo registrar el tema" });
69       }
70     }
71   );
72 });
```

Figura 23. Codificación HU004.

```
1 router.post("/editarTema", [], (req, res) => {
2   let id = req.body.id;
3   let titulo = req.body.titulo;
4   let objetivos = req.body.objetivos;
5   let descripcion = req.body.descripcion;
6   let recursos = req.body.recursos;
7   let estado = req.body.estado;
8   let temaEditadoBackend = { id, titulo, objetivos, descripcion, recursos, estado };
9
10  const editarTema =
11    "UPDATE tema SET titulo = ?, objetivos = ?, descripcion = ?, recursos = ?, estado = ? WHERE id = ?";
12  sql.ejecutarResSQL(
13    editarTema,
14    [titulo, objetivos, descripcion, recursos, estado, id],
15    (resultado) => {
16      if (resultado["affectedRows"] > 0)
17        return res
18          .status(200)
19          .send({ en: 1, m: "Se editó el tema con éxito", idTema: id, temaEditadoBackend });
20      return res.status(200).send({ en: -1, m: "No se pudo editar el tema" });
21    }
22  );
23 });
```

Figura 24. Codificación HU005.

```
1 router.post("/registrarCambio", (req, res) => {
2   let tipoEntidad = req.body.tipoEntidad;
3   let idTema = req.body.idTema;
4   let idSubtema = req.body.idSubtema;
5   let idEjercicio = req.body.idEjercicio;
6   let idPregunta = req.body.idPregunta;
7   let detalles = req.body.detalles;
8   let idUsuario = req.body.idUsuario;
9
10  const registrarCambio =
11    "INSERT INTO historial (tipo_entidad, id_tema, id_subtema, id_ejercicio, id_pregunta, detalles, id_usuario) VALUES (?, ?, ?, ?, ?, ?, ?)";
12
13  sql.ejecutarResSQL(
14    registrarCambio,
15    [tipoEntidad, idTema, idSubtema, idEjercicio, idPregunta, detalles, idUsuario],
16    (resultado) => {
17      if (resultado["affectedRows"] > 0) {
18        return res.status(200).send({
19          en: 1,
20          m: "Se registró el cambio con éxito",
21          idCambio: resultado["insertId"],
22        });
23      } else {
24        return res.status(200).send({ en: -1, m: "No se pudo registrar el cambio" });
25      }
26    }
27  );
28 });
```

Figura 25. Codificación HU006.

```
1 router.post("/activarDesactivarTema", [], (req, res) => {
2   let id = req.body.id;
3   let estado = req.body.estado;
4   const actualizarTema = "UPDATE tema SET estado = ? WHERE id = ?";
5   sql.ejecutarResSQL(actualizarTema, [estado, id], (resultado) => {
6     if (resultado["affectedRows"] > 0)
7       return res
8         .status(200)
9         .send({ en: 1, m: "Cambio el estado del tema", idTema: id });
10    return res
11      .status(200)
12      .send({ en: -1, m: "No se pudo cambiar el estado del tema" });
13  });
14 });
```

Figura 26. Codificación HU007.

```
1 router.post("/buscarTemas", (req, res) => {
2   const busqueda = req.body.busqueda;
3   const idUsuario = req.body.idUsuario;
4
5   // Query para buscar temas que coincidan con el término de búsqueda
6   const buscarTemas =
7     SELECT
8     usuario_tema.idTema,
9     usuario_tema.progreso,
10    tema.titulo,
11    tema.objetivos,
12    tema.descripcion,
13    tema.recursos,
14    tema.estado
15  FROM ${DB} usuario_tema
16  INNER JOIN ${DB}.tema ON usuario_tema.idTema = tema.id
17  WHERE usuario_tema.idUsuario = ?
18  AND (
19    tema.titulo LIKE CONCAT('%', ?, '%')
20    OR tema.descripcion LIKE CONCAT('%', ?, '%')
21    OR tema.objetivos LIKE CONCAT('%', ?, '%')
22  )
23  ORDER BY tema.titulo ASC;
24  ;
25
26  sql.ejecutarResSQL(
27    buscarTemas,
28    [idUsuario, busqueda, busqueda],
29    (resultado) => {
30      if (resultado.length > 0) {
31        return res.status(200).send({
32          en: 1,
33          m: "contenido encontrado",
34          temas: resultado
35        });
36      } else {
37        return res.status(200).send({
38          en: -1,
39          m: "no se encontro contenido que coincida con la búsqueda"
40        });
41      }
42    });
43  });
44 });
```

Figura 27. Codificación HU008.

```

1 router.post("/listarTemas", (req, res) => {
2   let obtenerTemas;
3   let idUsuario = req.body.idUsuario;
4   if (req.body.mensaje === "temasActivos") {
5     obtenerTemas =
6     `SELECT usuario_tema.idTema, usuario_tema.progreso, tema.titulo, tema.objetivos, tema.descripcion, tema.recursos, tema.estado +
7     FROM ${BD}.usuario_tema +
8     INNER JOIN ${BD}.tema ON usuario_tema.idTema = tema.id +
9     WHERE usuario_tema.idUsuario = ? AND tema.estado = 1;`;
10  } else {
11    obtenerTemas =
12    `SELECT usuario_tema.idTema, usuario_tema.progreso, tema.titulo, tema.objetivos, tema.descripcion, tema.recursos, tema.estado +
13    FROM ${BD}.usuario_tema +
14    INNER JOIN ${BD}.tema ON usuario_tema.idTema = tema.id +
15    WHERE usuario_tema.idUsuario = ?;`;
16  }
17  sql.ejecutarResSQL(obtenerTemas, [idUsuario], (resultado) => {
18    if (resultado.length > 0) {
19      return res
20        .status(200)
21        .send({ en: 1, m: "Temas obtenidos", temas: resultado });
22    } else {
23      return res.status(200).send({ en: -1, m: "No se encontraron temas" });
24    }
25  });
26 });
27

```

Figura 28. Codificación HU009.

1.4 Pruebas

La

Tabla 2 muestra las pruebas unitarias a desarrollar en la primera iteración.

Tabla 2. Pruebas unitarias a ejecutar en la primera iteración

ID	Descripción	Datos de Entrada	Salida Esperada	Resultado
PU001	Registrar un nuevo estudiante	<ul style="list-style-type: none"> ➤ nombres: 'Juan', ➤ apellidos: 'Pérez', ➤ email: 'juan.perez@test.com', ➤ clave: '123456', ➤ tipoRol: 'estudiante' 	Un registro exitoso del usuario con el rol tipo estudiante	Aprobada
PU002	Registrar un nuevo docente	<ul style="list-style-type: none"> ➤ nombres: 'María', ➤ apellidos: 'García', ➤ email: 'maria.garcia@test.com', ➤ clave: '123456', ➤ tipoRol: 'docente' 	Un registro exitoso del usuario con el rol tipo docente	Aprobada
PU003	Registrar un nuevo administrador	<ul style="list-style-type: none"> ➤ nombres: 'Carlos', ➤ apellidos: 'López', ➤ email: 'carlos.lopez@test.com', ➤ clave: '123456', ➤ tipoRol: 'administrador' 	Un registro exitoso del usuario con el rol tipo docente	Aprobada



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

PU004	Registro con un email duplicado	<ul style="list-style-type: none"> ➤ nombres: 'Juan', ➤ apellidos: 'Pérez', ➤ email: 'juan.perez@test.com', ➤ clave: '123456', ➤ tipoRol: 'estudiante' 	El email ya está registrado.	Aprobada
PU005	Editar usuario con todos sus campos incluida la clave	<ul style="list-style-type: none"> ➤ userId: 47, ➤ nombres: 'Juan Editado', ➤ apellidos: 'Pérez Editado', ➤ email: 'juan.editado@test.com', ➤ clave: 'nueva123' 	Usuario actualizado exitosamente.	Aprobada
PU006	Editar usuario con todos sus campos sin incluir clave	<ul style="list-style-type: none"> ➤ userId: 47, ➤ nombres: 'Juan Editado', ➤ apellidos: 'Pérez Editado', ➤ email: 'juan.editado@test.com' 	Usuario actualizado exitosamente.	Aprobada
PU007	Edición de usuario inexistente	<ul style="list-style-type: none"> ➤ userId: 999999, ➤ nombres: 'Juan Editado', ➤ apellidos: 'Pérez Editado', ➤ email: 'juan.editado@test.com', ➤ clave: 'nueva123' 	Usuario no encontrado	Aprobada
PU008	Listar temas activos	<ul style="list-style-type: none"> ➤ idUsuario: 47, ➤ mensaje: "temasActivos" 	Temas obtenidos	Aprobada
PU009	Listar temas cuando existe ningún tema	<ul style="list-style-type: none"> ➤ idUsuario 9999 	No se encontraron temas	Aprobada
PU010	Listar temas activos e inactivos.	<ul style="list-style-type: none"> ➤ idUsuario: 47 	Temas obtenidos	Aprobada
PU011	Registrar un nuevo tema	<ul style="list-style-type: none"> ➤ título: 'Nuevo Tema de Prueba', ➤ objetivos: 'Objetivos del tema de prueba', ➤ descripcion: 'Descripción del tema de prueba', ➤ recursos: 'Recursos del tema de prueba' 	Se registró el tema con éxito y se enviaron correos de notificación	Aprobada
PU012	Editar un tema existente	<ul style="list-style-type: none"> ➤ id: 32, ➤ título: 'Título editado', ➤ objetivos: 'Objetivos editados', ➤ descripcion: 'Descripción editada', ➤ recursos: 'Recursos editados', ➤ estado: 1 	Se editó el tema con éxito	Aprobada
PU013	Editar un tema inexistente.	<ul style="list-style-type: none"> ➤ id: 99999, ➤ título: 'Título editado', ➤ objetivos: 'Objetivos editados', ➤ descripcion: 'Descripción editada', ➤ recursos: 'Recursos editados', ➤ estado: 1 	No se pudo editar el tema	Aprobada
PU014	Registrar un nuevo cambio en tema	<ul style="list-style-type: none"> ➤ tipoEntidad: "tema", ➤ idTema: 32, ➤ idSubtema: null, 	Se registró el cambio con éxito	Aprobada

		<ul style="list-style-type: none"> ➤ idEjercicio: null, ➤ idPregunta: null, ➤ detalles: "Se actualizó el contenido del tema", ➤ idUsuario: 47 		
PU015	Activar/Desactivar tema	<ul style="list-style-type: none"> ➤ id: 32, ➤ estado: 1 	Activar o desactivar tema	Aprobada
PU016	Buscar contenido con término existente	<ul style="list-style-type: none"> ➤ idUsuario: 47, ➤ busqueda: "Título editado". 	Contenido encontrado	Aprobada
PU017	Visualizar temas	➤ N/A	Temas activos listados	Aprobada

A continuación, se presentan las pruebas unitarias desarrolladas en la primera iteración.

```

1 describe('Test de registro de usuarios', () => {
2   it('Registrar un nuevo estudiante', async () => {
3     const response = await api
4       .post(rutaSesionUsuario + '/registro')
5       .send(bodyEstudiante);
6     expect(response.statusCode).toBe(200);
7     expect(response.body.en).toEqual(1);
8     expect(response.body.m).toEqual("Se registró correctamente el usuario");
9   });
10
11   it('Registrar un nuevo docente', async () => {
12     const response = await api
13       .post(rutaSesionUsuario + '/registro');
14     .send(bodyDocente);
15     expect(response.statusCode).toBe(200);
16     expect(response.body.en).toEqual(1);
17     expect(response.body.m).toEqual("Se registró correctamente el usuario");
18   });
19
20   it('Registrar un nuevo administrador', async () => {
21     const response = await api
22       .post(rutaSesionUsuario + '/registro')
23       .send(bodyAdministrador);
24     expect(response.statusCode).toBe(200);
25     expect(response.body.en).toEqual(1);
26     expect(response.body.m).toEqual("Se registró correctamente el usuario");
27   });
28
29   it('Intento de registro con email duplicado', async () => {
30     const response = await api
31       .post(rutaSesionUsuario + '/registro')
32       .send(bodyEstudiante);
33     expect(response.statusCode).toBe(200);
34     expect(response.body.en).toEqual(-1);
35     expect(response.body.m).toEqual("El email ya está registrado");
36   });
37 });

```

Figura 29. Pruebas unitarias (PU001, PU002, PU003, PU004), registrar usuario.



```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUno
> camigo@1.0.0 testIteracionUno
> cross-env BD=camigo_prueba jest --verbose --silent tests/sesionUsuario/sesionUsuario.test.js

PASS tests/sesionUsuario/sesionUsuario.test.js
Test de registro de usuarios
  ✓ Registrar un nuevo estudiante (106 ms)
  ✓ Registrar un nuevo docente (40 ms)
  ✓ Registrar un nuevo administrador (29 ms)
  ✓ Intento de registro con email duplicado (4 ms)

Test Suites: 1 passed, 1 total
Tests: 4 passed, 4 total
Snapshots: 0 total
Time: 1.333 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 30. Ejecución de las pruebas unitarias (PU001, PU002, PU003, PU004).

```
1 describe('Test de edición de usuarios', () => {
2   it('Editar usuario exitosamente con todos los campos incluyendo clave', async () => {
3     const response = await api
4       .post(rutaSesionUsuario + '/editarUsuario')
5       .send(bodyEdicionCompleta);
6     expect(response.statusCode).toBe(200);
7     expect(response.body.en).toEqual(1);
8     expect(response.body.m).toEqual("Usuario actualizado exitosamente");
9     expect(response.body).toHaveProperty('usuarioEditado');
10  });
11
12   it('Editar usuario exitosamente sin cambiar la clave', async () => {
13     const response = await api
14       .post(rutaSesionUsuario + '/editarUsuario')
15       .send(bodyEdicionSinClave);
16     expect(response.statusCode).toBe(200);
17     expect(response.body.en).toEqual(1);
18     expect(response.body.m).toEqual("Usuario actualizado exitosamente");
19     expect(response.body).toHaveProperty('usuarioEditado');
20  });
21
22   it('Intento de edición de usuario inexistente', async () => {
23     const bodyUsuarioInexistente = {
24       ...bodyEdicionCompleta,
25       userId: 99999 // ID que no existe
26     };
27     const response = await api
28       .post(rutaSesionUsuario + '/editarUsuario')
29       .send(bodyUsuarioInexistente);
30     expect(response.statusCode).toBe(404);
31     expect(response.body.en).toEqual(-1);
32     expect(response.body.m).toEqual("Usuario no encontrado");
33  });
34 });
```

Figura 31. Pruebas unitarias (PU005, PU006, PU007), editar usuario.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUno
> camigo@1.0.0 testIteracionUno
> cross-env BD=camigo_prueba jest --verbose --silent tests/sesionUsuario/sesionUsuario.test.js

PASS tests/sesionUsuario/sesionUsuario.test.js
  Test de edición de usuarios
    ✓ Editar usuario exitosamente con todos los campos incluyendo clave (96 ms)
    ✓ Editar usuario exitosamente sin cambiar la clave (9 ms)
    ✓ Intento de edición de usuario inexistente (5 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.363 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 32. Ejecución de las pruebas unitarias (PU005, PU006, PU007), editar usuario.

```
1 describe('Test de listado de temas', () => {
2   // Prueba para listar temas activos
3   it('Listar temas activos del usuario', async () => {
4     const response = await api
5       .post(rutaTemas + '/listarTemas')
6       .send(bodyListarActivos);
7
8     expect(response.statusCode).toBe(200);
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Temas obtenidos");
11    expect(Array.isArray(response.body.temas)).toBe(true);
12  });
13
14  // Prueba para el caso cuando no hay temas
15  it('Intento de listar temas cuando no hay ninguno', async () => {
16    const bodySinTemas = {
17      idUsuario: 999 // ID de usuario que no tiene temas asignados
18    };
19
20    const response = await api
21      .post(rutaTemas + '/listarTemas')
22      .send(bodySinTemas);
23
24    expect(response.statusCode).toBe(200);
25    expect(response.body.en).toBe(-1);
26    expect(response.body.m).toBe("No se encontraron temas");
27  });
28
29  // Prueba para listar todos los temas (activos e inactivos)
30  it('Listar todos los temas del usuario (activos e inactivos)', async () => {
31    const bodyTodosTemas = {
32      idUsuario: 47 // Usando el mismo ID que en la primera prueba
33    };
34
35    const response = await api
36      .post(rutaTemas + '/listarTemas')
37      .send(bodyTodosTemas);
38
39    expect(response.statusCode).toBe(200);
40    expect(response.body.en).toBe(1);
41    expect(response.body.m).toBe("Temas obtenidos");
42    expect(Array.isArray(response.body.temas)).toBe(true);
43  });
44 });
45
```

Figura 33. Pruebas unitarias (PU008, PU009, PU010), listar temas.



```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Test de listado de temas
    ✓ Listar temas activos del usuario (86 ms)
    ✓ Intento de listar temas cuando no hay ninguno (8 ms)
    ✓ Listar todos los temas del usuario (activos e inactivos) (6 ms)

Test Suites: 1 passed, 1 total
Tests: 3 passed, 3 total
Snapshots: 0 total
Time: 1.486 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 34. Ejecución pruebas unitarias (PU008, PU009, PU010), listar temas.

```
1 describe('Test de registro de temas', () => {
2   // Prueba para registrar un tema exitosamente
3   it('Registrar un nuevo tema con notificaciones', async () => {
4     const response = await api
5     .post(rutaTemas + '/registrarTema')
6     .send(bodyRegistrar);
7
8     expect(response.statusCode).toBe(200);
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Se registró el tema con éxito y se enviaron correos de notificación");
11    expect(response.body.idTema).toBeDefined();
12    expect(typeof response.body.idTema).toBe('number');
13  });
14 });
15
```

Figura 35. Pruebas unitarias (PU011), registrar temas.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Test de registro de temas
    ✓ Registrar un nuevo tema con notificaciones (117 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 1.129 s, estimated 7 s
Jest did not exit one second after the test run has completed.
```

Figura 36. Ejecución pruebas unitarias (PU011), registrar temas.

```
1 describe('Test de edición de temas', () => {
2   // Prueba para editar un tema exitosamente
3   it('Editar un tema existente', async () => {
4     const response = await api
5       .post(rutaTemas + '/editarTema')
6       .send(bodyEditar);
7
8     expect(response.statusCode).toBe(200);
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Se editó el tema con éxito");
11    expect(response.body.idTema).toBe(bodyEditar.id);
12    expect(response.body.temaEditadoBackend).toEqual(bodyEditar);
13  });
14
15  // Prueba para intentar editar un tema que no existe
16  it('Intentar editar un tema inexistente', async () => {
17    const bodyTemaInexistente = {
18      ...bodyEditar,
19      id: 999 // ID que no existe en la base de datos
20    };
21
22    const response = await api
23      .post(rutaTemas + '/editarTema')
24      .send(bodyTemaInexistente);
25
26    expect(response.statusCode).toBe(200);
27    expect(response.body.en).toBe(-1);
28    expect(response.body.m).toBe("No se pudo editar el tema");
29  });
30 });
```

Figura 37. Pruebas unitarias (PU012, PU013), editar temas.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Test de edición de temas
    ✓ Editar un tema existente (117 ms)
    ✓ Intentar editar un tema inexistente (7 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:       1.506 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 38. Ejecución pruebas unitarias (PU012, PU013), editar temas.

```
1 describe('Test de registro de cambios', () => {
2   // Prueba para registrar un cambio exitosamente
3   it('Registrar un nuevo cambio en tema', async () => {
4     const response = await api
5       .post(rutaHistorial + '/registrarCambio')
6       .send(bodyRegistrarCambio);
7
8     expect(response.statusCode).toBe(200);
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Se registró el cambio con éxito");
11    expect(response.body.idCambio).toBeDefined();
12    expect(typeof response.body.idCambio).toBe('number');
13  });
14 });
```

Figura 39. Pruebas unitarias (PU014), historial de cambios.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Test de registro de cambios
    ✓ Registrar un nuevo cambio en tema (88 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.033 s, estimated 7 s
Jest did not exit one second after the test run has completed.
```

Figura 40. Ejecución de pruebas unitarias (PU014), historial de cambios.

```
1 describe('Pruebas para activar/desactivar tema', () => {
2
3   it('debería activar un tema correctamente', async () => {
4     const response = await api
5       .post(rutaTemas + '/activarDesactivarTema')
6       .send(bodyActivar_Desactivar)
7       .expect(200)
8       .expect('Content-Type', /application\/json/);
9
10    expect(response.body.en).toBe(1);
11  });
12
13  it('debería desactivar un tema correctamente', async () => {
14    bodyActivar_Desactivar.estado = -1;
15
16    const response = await api
17      .post(rutaTemas + '/activarDesactivarTema')
18      .send(bodyActivar_Desactivar)
19      .expect(200)
20      .expect('Content-Type', /application\/json/);
21
22    expect(response.body.en).toBe(1);
23  });
24 });
```

Figura 41. Pruebas unitarias (PU015), activar/desactivar tema

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Pruebas para activar/desactivar tema
    ✓ debería activar un tema correctamente (123 ms)
    ✓ debería desactivar un tema correctamente (16 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.793 s
Jest did not exit one second after the test run has completed.
```

Figura 42. Ejecución pruebas unitarias (PU015), activar/desactivar tema

```
1 describe('Test de búsqueda de temas', () => {
2   it('Buscar contenido con término existente', async () => {
3     const response = await api
4       .post(rutaTemas + '/buscarTemas')
5       .send(bodyBusquedaValida);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Contenido encontrado");
10    expect(Array.isArray(response.body.temas)).toBe(true);
11    expect(response.body.temas.length).toBeGreaterThan(0);
12
13    // Verificar la estructura de cada tema encontrado
14    const primerTema = response.body.temas[0];
15    expect(primerTema).toHaveProperty('idTema');
16    expect(primerTema).toHaveProperty('progreso');
17    expect(primerTema).toHaveProperty('titulo');
18    expect(primerTema).toHaveProperty('objetivos');
19    expect(primerTema).toHaveProperty('descripcion');
20    expect(primerTema).toHaveProperty('recursos');
21    expect(primerTema).toHaveProperty('estado');
22  });
23 });
24
```

Figura 43. Pruebas unitarias (PU016), Buscar temas

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Test de búsqueda de temas
    ✓ Buscar contenido con término existente (86 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.351 s, estimated 3 s
Jest did not exit one second after the test run has completed.
```

Figura 44. Ejecución pruebas unitarias (PU016), Buscar temas


```

1 describe('Test de listado de temas', () => {
2   // Prueba para listar temas activos
3   it('Listar temas activos del usuario', async () => {
4     const response = await api
5       .post(rutaTemas + '/listarTemas')
6       .send(bodyListarActivos);
7
8     expect(response.statusCode).toBe(200);
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Temas obtenidos");
11    expect(Array.isArray(response.body.temas)).toBe(true);
12  });

```

Figura 45. Pruebas unitarias (PU017), Visualizar temas

```

PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Test de listado de temas
    ✓ Listar temas activos del usuario (86 ms)
    ✓ Intento de listar temas cuando no hay ninguno (8 ms)
    ✓ Listar todos los temas del usuario (activos e inactivos) (6 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.486 s, estimated 2 s
Jest did not exit one second after the test run has completed.

```

Figura 46. Ejecución pruebas unitarias (PU017), Visualizar temas

2 Segunda Iteración

2.1 Planificación

La **Tabla 3** muestra las actividades específicas a desarrollar en la segunda iteración.

Tabla 3. Historias de Usuario a desarrollar en la segunda iteración.

HU #	Historia de usuario	Fecha de creación	Fecha de entrega
HU010	Listar subtemas	30/11/2023	20/12/2023
HU011	Registrar subtemas	01/12/2023	20/12/2023
HU012	Editar subtemas	05/12/2023	20/12/2023
HU013	Historial de cambios por subtema	09/12/2023	20/12/2023
HU014	Activar/Desactivar subtemas	12/12/2023	20/12/2023
HU015	Buscar subtemas	15/12/2023	20/12/2023

2.2 Diseño

La **Figura 47** muestra las funcionalidades de manera general a desarrollar en la segunda iteración.

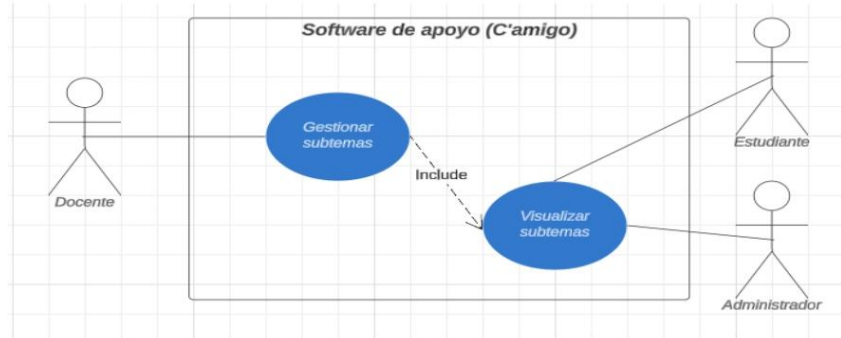


Figura 47. Casos de Uso general para la segunda iteración.

2.2.1 Diagramas BPMN

A continuación, se presentan los diagramas BPMN pertenecientes a cada Historia de Usuario a desarrollar en la segunda iteración.

2.2.1.1 HU010: Listar subtemas

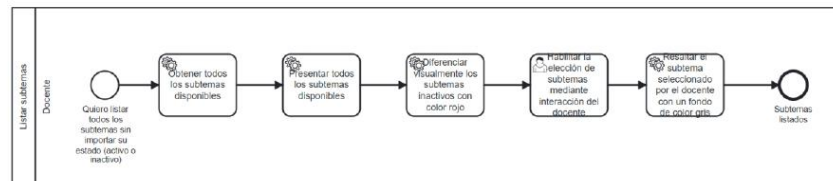


Figura 48. Diagrama BPMN de HU010.

2.2.1.2 HU011: Registrar subtemas

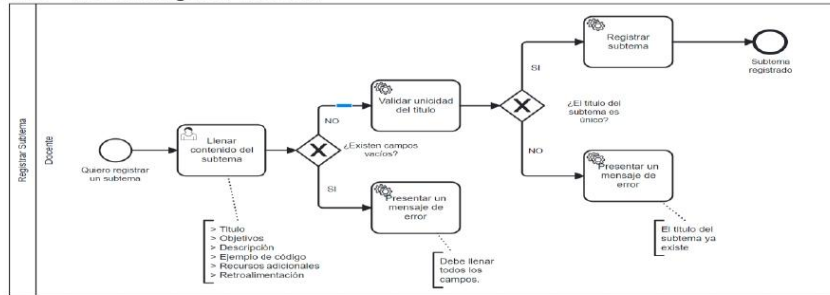


Figura 49. Diagrama BPMN de HU011.

2.2.1.3 HU012: Editar

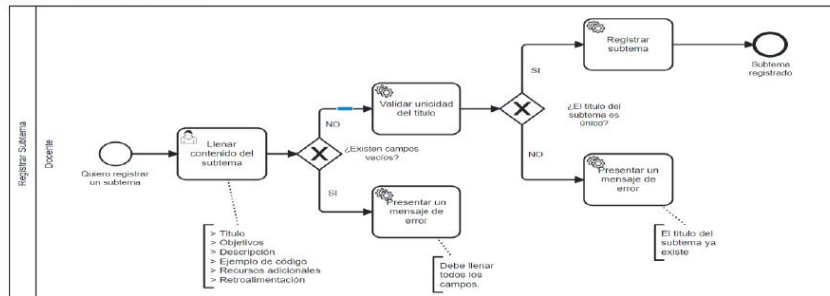


Figura 50. Diagrama BPMN de HU012.

2.2.1.4 HU013: Historial de cambios por subtema

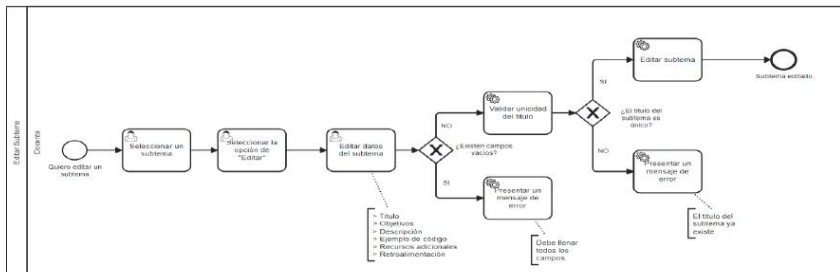


Figura 51. Diagrama BPMN de HU013.

2.2.1.5 HU014: Activar/Desactivar subtemas

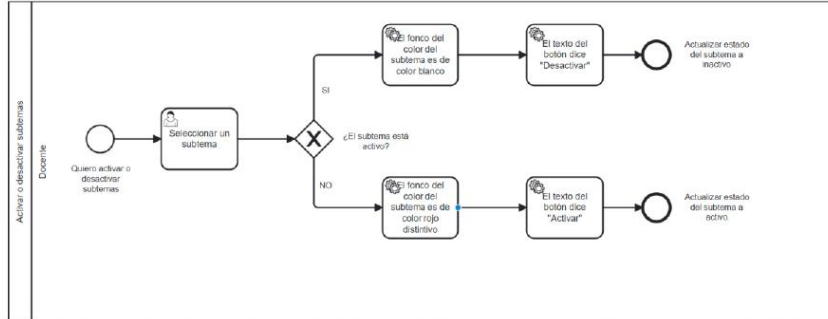


Figura 52. Diagrama BPMN de HU014.

2.2.1.6 HU015: Buscar subtemas

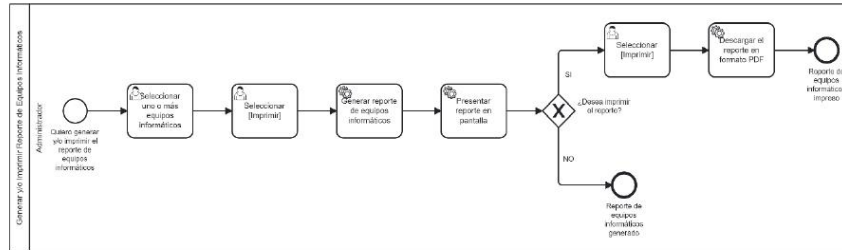


Figura 53. Diagrama BPMN de HU015.

2.2.1.7 HU016: Visualizar subtemas

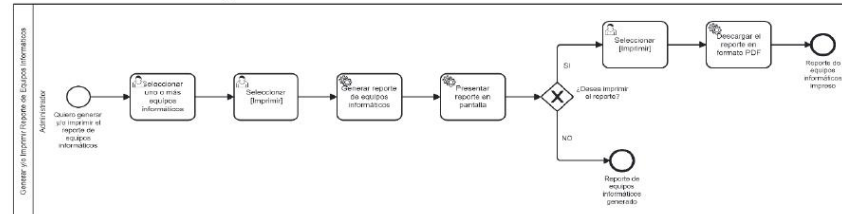


Figura 54. Diagrama BPMN de HU016.

2.2.2 Prototipos

A continuación, se presentan los prototipos pertenecientes a cada Historia de Usuario a desarrollar en la segunda iteración.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

2.2.2.1 HU010: Listar subtemas

Subtemas

Subtemas inactivos de rojo

Buscar tema

Subtema 1
Subtema 2
Subtema 3
Subtema 4

Crear Editar Activar/Desactivar Historial de cambios

Figura 55. Prototipo de HU010.

2.2.2.2 HU011: Registrar subtemas

C-Compañero

DOCENTE

Docente OCT 2023 - FEB 2024

Crear subtemas

Título:

Objetivos:

Descripción:

Ejemplo de código:

Recursos adicionales:

Retroalimentación:

Cancelar Guardar

Figura 56. Prototipo de HU011.

2.2.2.3 HU012: Editar subtemas

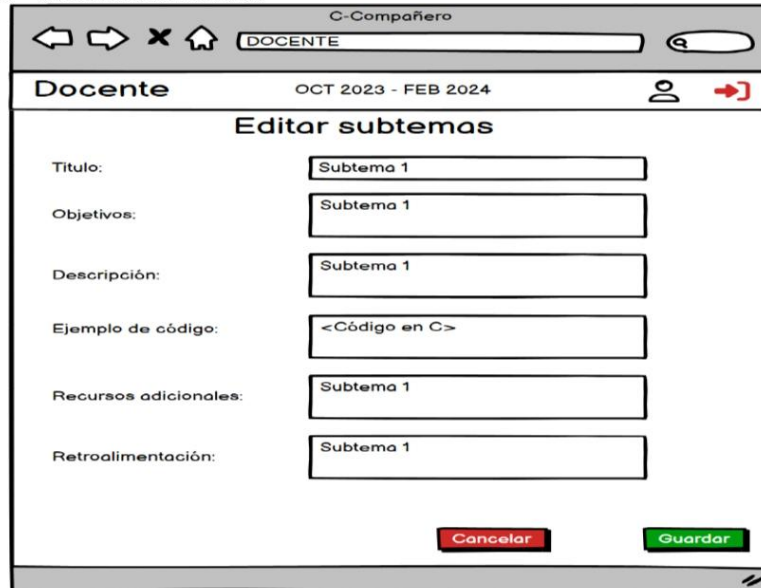
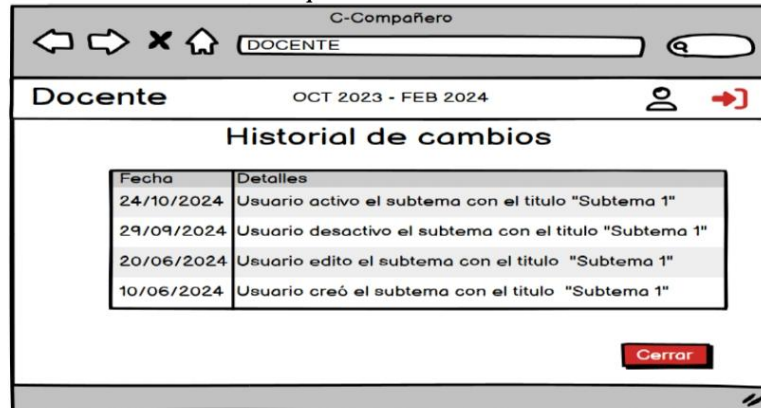


Figura 57. Prototipo de HU012.

2.2.2.4 HU013: Historial de cambios por subtema



Fecha	Detalles
24/10/2024	Usuario activo el subtema con el titulo "Subtema 1"
29/09/2024	Usuario desactivo el subtema con el titulo "Subtema 1"
20/06/2024	Usuario edito el subtema con el titulo "Subtema 1"
10/06/2024	Usuario creó el subtema con el titulo "Subtema 1"

Figura 58. Prototipo de HU013.



2.2.2.5 HU014: Activar/Desactivar subtemas

Subtemas

Subtemas inactivos de rojo

Q Buscar tema

Subtema 1
Subtema 2
Subtema 3
Subtema 4

Crear Editar Activar/Desactivar Historial de cambios

Figura 59. Prototipo de HU014.

2.2.2.6 HU015: Buscar subtemas

Subtemas

Subtemas inactivos de rojo

Q Buscar tema

Subtema 1
Subtema 2
Subtema 3
Subtema 4

Crear Editar Activar/Desactivar Historial de cambios

Figura 60. Prototipo de HU015.

2.2.2.7 HU016: Visualizar subtemas

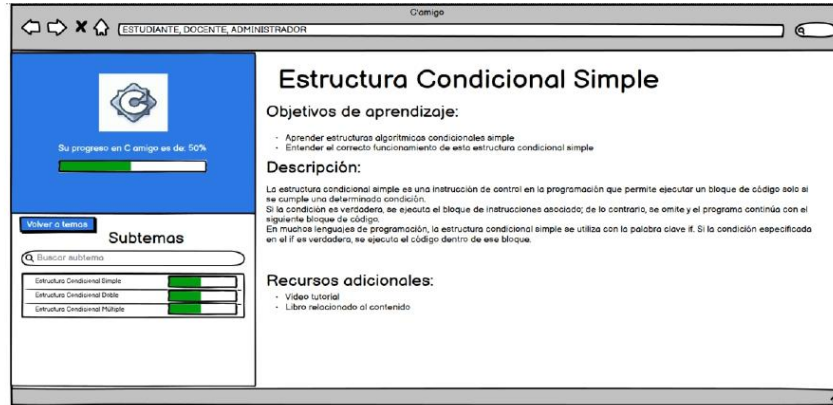


Figura 61. Prototipo de HU016.

2.3 Codificación

A continuación, se presenta los fragmentos principales del código fuente desarrollados en la segunda iteración.

```

1  main:post('/listarSubtemas', req, res) => {
2    [titulos = req.body.titulos;
3     documentos = req.body.documentos;
4     obtenerSubtemas;
5     if (req.body.metodo === 'obtenerSubtemas') {
6       obtenerSubtemas =
7         SELECT tema_sistema_sistemas, tema_sistema.progreso, subtema.titulo, subtema.objetivo, subtema.descripcion, subtema.codigo_codigo, subtema.recurso, subtema.recursos, subtema.recursos, subtema.recursos, subtema.recursos
8         FROM [DB] tema_sistema
9         INNER JOIN [DB] sistema ON tema_sistema.sistema = sistema.id
10        WHERE sistema.status = 'NO' AND sistema = 'NO' AND sistema = '1';
11     } else {
12       obtenerSubtemas =
13         SELECT tema_sistema_sistemas, tema_sistema.progreso, subtema.titulo, subtema.objetivo, subtema.descripcion, subtema.codigo_codigo, subtema.recurso, subtema.recursos, subtema.recursos, subtema.recursos
14         FROM [DB] tema_sistema
15         INNER JOIN [DB] sistema ON tema_sistema.sistema = sistema.id
16         WHERE sistema.status = 'NO' AND sistema = '1';
17     }
18     sql.execute(sql.query(obtenerSubtemas, [titulos, documentos]), (resultado) => {
19       if (resultado.length > 0) {
20         return res;
21         return res.json(resultado);
22         return res.json(resultado);
23       } else {
24         return res.status(200).send({ msg: 'No se encontraron sistemas' });
25       }
26     });
27   };

```

Figura 62. Codificación HU010.



```

1 router.post("/registroSubitem", (req, res) => {
2   let id = req.body.id;
3   let titulo = req.body.titulo;
4   let objetivos = req.body.objetivos;
5   let descripcion = req.body.descripcion;
6   let recursos = req.body.recursos;
7   let retroalimentacion = req.body.retroalimentacion;
8   let estado = req.body.estado;
9   let subitem = { id, titulo, objetivos, descripcion, ejemploCodigo, recursos, retroalimentacion, estado };
10  const registrarSubitem =
11    async () => {
12      await db.collection('subitem').insertOne(subitem);
13    };
14  registrarSubitem()
15    .then(() => {
16      res.status(200).send({
17        message: "Se registró el subitem con éxito y se envió correo de notificación",
18        subitem: subitem
19      });
20    })
21    .catch((err) => {
22      res.status(500).send({
23        message: "No se pudo registrar el subitem",
24      });
25    });
26 });

```

Figura 63. Codificación HU011.

```

1 router.post("/editarSubitem", (req, res) => {
2   let id = req.body.id;
3   let titulo = req.body.titulo;
4   let objetivos = req.body.objetivos;
5   let descripcion = req.body.descripcion;
6   let ejemploCodigo = req.body.ejemploCodigo;
7   let recursos = req.body.recursos;
8   let retroalimentacion = req.body.retroalimentacion;
9   let estado = req.body.estado;
10  let subitemEditadoBackend = { id, titulo, objetivos, descripcion, ejemploCodigo, recursos, retroalimentacion, estado };
11  const editarSubitem =
12    async () => {
13      await db.collection('subitem').updateOne({ _id: id }, { $set: subitemEditadoBackend });
14    };
15  editarSubitem()
16    .then(() => {
17      res.status(200).send({
18        message: "Se editó el subitem con éxito, idSubitem: id, subitemEditadoBackend",
19      });
20    })
21    .catch((err) => {
22      res.status(500).send({
23        message: "No se pudo editar el subitem",
24      });
25    });
26 });

```

Figura 64. Codificación HU012.

```
1 router.post("/registrarCambio", (req, res) => {
2   let tipoEntidad = req.body.tipoEntidad;
3   let idTema = req.body.idTema;
4   let idSubtema = req.body.idSubtema;
5   let idEjercicio = req.body.idEjercicio;
6   let idPregunta = req.body.idPregunta;
7   let detalles = req.body.detalles;
8   let idUsuario = req.body.idUsuario;
9
10  const registrarCambio =
11    "INSERT INTO Historial (tipo_entidad, id_tema, id_subtema, id_ejercicio, id_pregunta, detalles, id_usuario) VALUES (?, ?, ?, ?, ?, ?, ?)";
12
13  sql.ejecutarResSQL(
14    registrarCambio,
15    [tipoEntidad, idTema, idSubtema, idEjercicio, idPregunta, detalles, idUsuario],
16    (resultado) => {
17      if (resultado["affectedRows"] > 0) {
18        return res.status(200).send({
19          en: 1,
20          m: "Se registro el cambio con exito",
21          idCambio: resultado["insertId"],
22        });
23      } else {
24        return res.status(200).send({ en: -1, m: "No se pudo registrar el cambio" });
25      }
26    });
27  });
28  });
```

Figura 65. Codificación HU013.

```
1 router.post("/activarDesactivarSubtema", [], (req, res) => {
2   let id = req.body.id;
3   let estado = req.body.estado;
4   const actualizarSubtema = "UPDATE subtema SET estado = ? WHERE id = ?";
5   sql.ejecutarResSQL(actualizarSubtema, [estado, id], (resultado) => {
6     //return res.status(200).send({result:resultado, titulo:titulo, descripcion:descripcion, id:id})
7     if (resultado["affectedRows"] > 0)
8       return res
9         .status(200)
10        .send({ en: 1, m: "Cambio el estado del subtema", idSubtema: id });
11    return res
12      .status(200)
13      .send({ en: -1, m: "No se pudo cambiar el estado del subtema" });
14  });
15  });
```

Figura 66. Codificación HU014.

```

1 router.post('/buscarTemas', (req, res) => {
2   const busqueda = req.body.busqueda;
3   const idUsuario = req.body.idUsuario;
4
5   // Query para buscar temas que coincidan con el término de búsqueda
6   const buscarTemas = `
7     SELECT
8     usuario_tema.idTema,
9     usuario_tema.progreso,
10    tema.titulo,
11    tema.objetivos,
12    tema.descripcion,
13    tema.recursos,
14    tema.estado
15  FROM $IDB).usuario_tema
16  INNER JOIN $IDB).tema ON usuario_tema.idTema = tema.id
17  WHERE usuario_tema.idUsuario = ?
18  AND (
19    tema.titulo LIKE CONCAT('%', ?, '%')
20    OR tema.descripcion LIKE CONCAT('%', ?, '%')
21    OR tema.objetivos LIKE CONCAT('%', ?, '%')
22  )
23  ORDER BY tema.titulo ASC;
24  `;
25
26  sql.ejecutarResSQL(
27    buscarTemas,
28    [idUsuario, busqueda, busqueda],
29    (resultado) => {
30      if (resultado.length > 0) {
31        return res.status(200).send({
32          msg: 'Contenido encontrado',
33          temas: resultado
34        });
35      } else {
36        return res.status(300).send({
37          msg: 'No se encontró contenido que coincida con la búsqueda'
38        });
39      }
40    });
41  });
42  });
43  });
44  });

```

Figura 67. Codificación HU015.

```

1 router.post('/listarSubtemas', (req, res) => {
2   let idTema = req.body.idTema;
3   let idUsuario = req.body.idUsuario;
4   let obtenerSubtemas = `
5     if (req.body.mensaje === "subtemasActivos") {
6       SELECT tema.subtema_idSubtema, tema.subtema.progreso, subtema.titulo, subtema.objetivos, subtema.descripcion, subtema.empleoCodigo, subtema.recurso, subtema.retroliderazgo, subtema.estado
7     FROM $IDB).tema_subtema
8     INNER JOIN $IDB).subtema ON tema.subtema_idSubtema = subtema.id
9     WHERE subtema.idTema = ? AND idUsuario = ? AND estado = 1;
10    } else {
11      SELECT tema.subtema_idSubtema, tema.subtema.progreso, subtema.titulo, subtema.objetivos, subtema.descripcion, subtema.empleoCodigo, subtema.recurso, subtema.retroliderazgo, subtema.estado
12    FROM $IDB).tema_subtema
13    INNER JOIN $IDB).subtema ON tema.subtema_idSubtema = subtema.id
14    WHERE subtema.idTema = ? AND idUsuario = ?;
15    }
16  `;
17  sql.ejecutarResSQL(obtenerSubtemas, [idTema, idUsuario], (resultado) => {
18    if (resultado.length > 0) {
19      return res
20        .status(200)
21        .send({
22          msg: 'Subtemas obtenidos', subtemas: resultado
23        });
24    } else {
25      return res.status(200).send({ msg: 'No se encontraron subtemas' });
26    }
27  });
28  });

```

Figura 68. Codificación HU016.

2.4 Pruebas

La Tabla 4 muestra las pruebas unitarias a desarrollar en la primera iteración.

Tabla 4. Pruebas unitarias a ejecutar en la segunda iteración

ID	Descripción	Datos de Entrada	Salida Esperada	Resultado
PU018	Listar subtemas activos	<ul style="list-style-type: none"> ➤ idTema: 32, ➤ idUsuario: 47, ➤ mensaje: "subtemasActivos" 	Subtemas obtenidos	Aprobada



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

PU019	Listar subtemas cuando existe ningún tema	<ul style="list-style-type: none"> ➤ idTema 9999 ➤ idUsuario: 47, ➤ mensaje: "subtemasActivos" 	No se encontraron subtemas	Aprobada
PU020	Listar subtemas activos e inactivos.	<ul style="list-style-type: none"> ➤ idTema: 32, ➤ idUsuario: 47 	Subtemas obtenidos	Aprobada
PU021	Registrar un nuevo subtema	<ul style="list-style-type: none"> ➤ idTema: 32, ➤ titulo: 'Nuevo Subtema de Prueba', ➤ objetivos: 'Objetivos del subtema de prueba', ➤ descripcion: 'Descripción del subtema de prueba', ➤ ejemploCodigo: 'console.log("Ejemplo de código");', ➤ recursos: 'Recursos del subtema de prueba', ➤ retroalimentacion: 'Retroalimentación del subtema de prueba' 	Se registró el subtema con éxito y se enviaron correos de notificación	Aprobada
PU022	Editar un subtema existente	<ul style="list-style-type: none"> ➤ id: 37, ➤ titulo: 'Título editado del subtema', ➤ objetivos: 'Objetivos editados del subtema', ➤ descripcion: 'Descripción editada del subtema', ➤ ejemploCodigo: 'console.log("Código editado");', ➤ recursos: 'Recursos editados del subtema', ➤ retroalimentacion: 'Retroalimentación editada del subtema', ➤ estado: 1 	Se editó el subtema con éxito	Aprobada
PU023	Editar un subtema inexistente.	<ul style="list-style-type: none"> ➤ id: 99999, ➤ titulo: 'Título editado del subtema', ➤ objetivos: 'Objetivos editados del subtema', ➤ descripcion: 'Descripción editada del subtema', ➤ ejemploCodigo: 'console.log("Código editado");', ➤ recursos: 'Recursos editados del subtema', ➤ retroalimentacion: 'Retroalimentación editada del subtema', ➤ estado: 1 	No se pudo editar el subtema	Aprobada

PU024	Registrar un nuevo cambio en subtema	<ul style="list-style-type: none"> ➢ tipoEntidad: "subtema", ➢ idTema: 32, ➢ idSubtema: 37, ➢ idEjercicio: null, ➢ idPregunta: null, ➢ detalles: "Se actualizó el contenido del subtema", ➢ idUsuario: 47 	Se registró el cambio con éxito	Aprobada
PU025	Activar/Desactivar subtema	<ul style="list-style-type: none"> ➢ id: 37, ➢ estado: 1 	Activar o desactivar subtema	Aprobada
PU026	Buscar contenido con término existente	<ul style="list-style-type: none"> ➢ idUsuario: 47, ➢ búsqueda: "Título editado". 	Contenido encontrado	Aprobada
PU027	Visualizar subtemas	➢ N/A	Subtemas activos listados	Aprobada

A continuación, se presentan las seis pruebas unitarias desarrolladas en la segunda iteración.

```

1 describe('Test de listado de subtemas', () => {
2   it('listar subtemas del tema del tema', async () => {
3     const response = await api
4       .post('/rutaSubtemas + /listarSubtemas')
5       .send(bodyListaActivos);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toHaveLength(3);
9     expect(response.body.m).toHaveLength(3);
10    expect(Array.isArray(response.body.subtemas)).toBe(true);
11  });
12
13  it('Intento de listar subtemas cuando no hay ninguno', async () => {
14    const bodySinSubtemas = {
15      idTema: 999, // ID de tema que no tiene subtemas
16      idUsuario: 47,
17      mensajes: "subtemasActivos"
18    };
19
20    const response = await api
21      .post('/rutaSubtemas + /listarSubtemas')
22      .send(bodySinSubtemas);
23
24    expect(response.statusCode).toBe(200);
25    expect(response.body.en).toHaveLength(1);
26    expect(response.body.m).toHaveLength(1);
27  });
28
29  it('Listar todos los subtemas del tema (activos e inactivos)', async () => {
30    const bodyTodosSubtemas = {
31      idTema: 32,
32      idUsuario: 47
33    };
34
35    const response = await api
36      .post('/rutaSubtemas + /listarSubtemas')
37      .send(bodyTodosSubtemas);
38
39    expect(response.statusCode).toBe(200);
40    expect(response.body.en).toHaveLength(3);
41    expect(response.body.m).toHaveLength(3);
42    expect(Array.isArray(response.body.subtemas)).toBe(true);
43  });
44  });
45

```

Figura 69. Pruebas unitarias (PU018, PU019, PU020), listar subtemas.



```

PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionDos
> camigo@1.0.0 testIteracionDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/subtema/subtema.test.js

PASS tests/subtema/subtema.test.js
  Test de listado de subtemas
    ✓ Listar subtemas activos del tema (81 ms)
    ✓ Intento de listar subtemas cuando no hay ninguno (10 ms)
    ✓ Listar todos los subtemas del tema (activos e inactivos) (6 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.199 s, estimated 2 s
Jest did not exit one second after the test run has completed.

```

Figura 70. Ejecución pruebas unitarias (PU018, PU019, PU020), listar subtemas.

```

1 describe('Test de registro de subtemas', () => {
2   it('Registrar un nuevo subtema con notificaciones', async () => {
3     const response = await api
4       .post(rutaSubtemas + '/registrarSubtema')
5       .send(bodyRegistrar);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Se registró el subtema con éxito y se enviaron correos de notificación");
10    expect(response.body.idSubtema).toBeDefined();
11    expect(typeof response.body.idSubtema).toBe('number');
12  });
13 });

```

Figura 71. Pruebas unitarias (PU021), registrar subtemas.

```

PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionDos
> camigo@1.0.0 testIteracionDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/subtema/subtema.test.js

PASS tests/subtema/subtema.test.js
  Test de registro de subtemas
    ✓ Registrar un nuevo subtema con notificaciones (134 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.305 s, estimated 7 s
Jest did not exit one second after the test run has completed.

```

Figura 72. Ejecución pruebas unitarias (PU021), registrar subtemas.

```
1 describe('Test de edición de subtemas', () => {
2   it('Editar un subtema existente', async () => {
3     const response = await api
4       .post(rutaSubtemas + '/editarSubtema')
5       .send(bodyEditar);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Se editó el subtema con éxito");
10    expect(response.body.idSubtema).toBe(bodyEditar.id);
11    expect(response.body.subtemaEditadoBackend).toEqual(bodyEditar);
12  });
13
14  it('Intentar editar un subtema inexistente', async () => {
15    const bodySubtemaInexistente = {
16      ...bodyEditar,
17      id: 999 // ID que no existe en la base de datos
18    };
19
20    const response = await api
21      .post(rutaSubtemas + '/editarSubtema')
22      .send(bodySubtemaInexistente);
23
24    expect(response.statusCode).toBe(200);
25    expect(response.body.en).toBe(-1);
26    expect(response.body.m).toBe("No se pudo editar el subtema");
27  });
28 });
```

Figura 73. Pruebas unitarias (PU022, PU023), editar subtemas.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionDos
> camigo@1.0.0 testIteracionDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/subtema/subtema.test.js
PASS tests/subtema/subtema.test.js
  Test de edición de subtemas
    ✓ Editar un subtema existente (106 ms)
    ✓ Intentar editar un subtema inexistente (10 ms)
Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 1.563 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 74. Ejecución pruebas unitarias (PU022, PU023), editar subtemas.

```
1 describe('Test de registro de cambios', () => {
2   // Prueba para registrar un cambio exitosamente
3   it('Registrar un nuevo cambio en tema', async () => {
4     const response = await api
5       .post(rutaHistorial + '/registrarCambio')
6       .send(bodyRegistrarCambio);
7
8     expect(response.statusCode).toBe(200);
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Se registró el cambio con éxito");
11    expect(response.body.idCambio).toBeDefined();
12    expect(typeof response.body.idCambio).toBe('number');
13  });
14 });
```

Figura 75. Pruebas unitarias (PU024), historial de cambios en subtemas.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionDos
> camigo@1.0.0 testIteracionDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/subtema/subtema.test.js

PASS tests/subtema/subtema.test.js
  Test de registro de cambios en subtema
    ✓ Registrar un nuevo cambio en subtema (111 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.558 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 76. Ejecución pruebas unitarias (PU024), historial de cambios en subtemas.


```

1 describe('Pruebas para activar/desactivar subtema', () => {
2   it('deberia activar un subtema correctamente', async () => {
3     const response = await api
4       .post(rutaSubtemas + '/activarDesactivarSubtema')
5       .send(bodyActivar_Desactivar)
6       .expect(200)
7       .expect('Content-Type', /application\/json/);
8
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Cambio el estado del subtema");
11    expect(response.body.idSubtema).toBe(bodyActivar_Desactivar.id);
12  });
13
14  it('deberia desactivar un subtema correctamente', async () => {
15    bodyActivar_Desactivar.estado = -1;
16
17    const response = await api
18      .post(rutaSubtemas + '/activarDesactivarSubtema')
19      .send(bodyActivar_Desactivar)
20      .expect(200)
21      .expect('Content-Type', /application\/json/);
22
23    expect(response.body.en).toBe(1);
24    expect(response.body.m).toBe("Cambio el estado del subtema");
25    expect(response.body.idSubtema).toBe(bodyActivar_Desactivar.id);
26  });
27
28  it('deberia manejar el intento de activar/desactivar un subtema inexistente', async () => {
29    const bodySubtemaInexistente = {
30      ...bodyActivar_Desactivar,
31      id: 999 // ID que no existe
32    };
33
34    const response = await api
35      .post(rutaSubtemas + '/activarDesactivarSubtema')
36      .send(bodySubtemaInexistente)
37      .expect(200)
38      .expect('Content-Type', /application\/json/);
39
40    expect(response.body.en).toBe(-1);
41    expect(response.body.m).toBe("No se pudo cambiar el estado del subtema");
42  });
43 });

```

Figura 77. Pruebas unitarias (PU025), activar/desactivar subtemas.

```

PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionDos
> camigo@1.0.0 testIteracionDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/subtema/subtema.test.js
PASS tests/subtema/subtema.test.js
  Pruebas para activar/desactivar subtema
    ✓ deberia activar un subtema correctamente (91 ms)
    ✓ deberia desactivar un subtema correctamente (17 ms)
    ✓ deberia manejar el intento de activar/desactivar un subtema inexistente (6 ms)
Test Suites: 1 passed, 1 total
Tests: 3 passed, 3 total
Snapshots: 0 total
Time: 1.265 s, estimated 2 s
Jest did not exit one second after the test run has completed.

```

Figura 78. Ejecución pruebas unitarias (PU025), activar/desactivar subtema

```

1 describe('Test de búsqueda de temas', () => {
2   it('Buscar contenido con término existente', async () => {
3     const response = await api
4       .post(rutaTemas + '/buscarTemas')
5       .send(bodyBusquedaValida);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe('Contenido encontrado');
10    expect(Array.isArray(response.body.temas)).toBe(true);
11    expect(response.body.temas.length).toBeGreaterThan(0);
12
13    // Verificar la estructura de cada tema encontrado
14    const primerTema = response.body.temas[0];
15    expect(primerTema).toHaveProperty('idTema');
16    expect(primerTema).toHaveProperty('progreso');
17    expect(primerTema).toHaveProperty('titulo');
18    expect(primerTema).toHaveProperty('objetivos');
19    expect(primerTema).toHaveProperty('descripcion');
20    expect(primerTema).toHaveProperty('recursos');
21    expect(primerTema).toHaveProperty('estado');
22  });
23 });
24

```

Figura 79. Pruebas unitarias (PU026), buscar subtemas.

```

PS C:\Users\juane\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Test de búsqueda de temas
    / Buscar contenido con término existente (86 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.351 s, estimated 3 s
Jest did not exit one second after the test run has completed.

```

Figura 80. Ejecución pruebas unitarias (PU026), buscar subtemas.

```

1 describe('Test de listado de subtemas', () => {
2   it('Listar subtemas activos del tema', async () => {
3     const response = await api
4       .post(rutaSubtemas + '/listarSubtemas')
5       .send(bodyListarActivos);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe('subtemas obtenidos');
10    expect(Array.isArray(response.body.subtemas)).toBe(true);
11  });
12
13   it('Intento de listar subtemas cuando no hay ninguno', async () => {
14     const bodySinSubtemas = {
15       idTema: 999, // id de tema que no tiene subtemas
16       idUsuario: 47,
17       mensaje: "subtemasActivos"
18     };
19
20     const response = await api
21       .post(rutaSubtemas + '/listarSubtemas')
22       .send(bodySinSubtemas);
23
24     expect(response.statusCode).toBe(200);
25     expect(response.body.en).toBe(-1);
26     expect(response.body.m).toBe("no se encontraron subtemas");
27   });
28
29   it('Listar todos los subtemas del tema (activos e inactivos)', async () => {
30     const bodyTodosSubtemas = {
31       idTema: 32,
32       idUsuario: 47
33     };
34
35     const response = await api
36       .post(rutaSubtemas + '/listarSubtemas')
37       .send(bodyTodosSubtemas);
38
39     expect(response.statusCode).toBe(200);
40     expect(response.body.en).toBe(1);
41     expect(response.body.m).toBe('subtemas obtenidos');
42     expect(Array.isArray(response.body.subtemas)).toBe(true);
43   });
44 });
45

```

Figura 81. Pruebas unitarias (PU027), visualizar subtemas.



```

PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionDos
> camigo@1.0.0 testIteracionDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/subtema/subtema.test.js

PASS tests/subtema/subtema.test.js
  Test de listado de subtemas
    ✓ Listar subtemas activos del tema (81 ms)
    ✓ Intento de listar subtemas cuando no hay ninguno (10 ms)
    ✓ Listar todos los subtemas del tema (activos e inactivos) (6 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.199 s, estimated 2 s
Jest did not exit one second after the test run has completed.

```

Figura 82. Ejecución pruebas unitarias (PU027), visualizar subtemas.

3 Tercera Iteración

3.1 Planificación

La **Tabla 5** muestra las actividades específicas a desarrollar en la tercera iteración.

Tabla 5. Historias de Usuario a desarrollar en la tercera iteración.

HU #	Historia de usuario	Fecha de creación	Fecha de entrega
HU017	Listar ejercicios	06/11/2023	24/12/2023
HU018	Registrar ejercicios	10/12/2023	24/12/2023
HU019	Editar ejercicios	12/12/2023	24/12/2023
HU020	Historial de cambios por ejercicio	15/12/2023	24/12/2023
HU021	Activar/Desactivar ejercicios	17/12/2023	24/12/2023
HU022	Buscar ejercicios	19/12/2023	24/12/2023
HU023	Visualizar ejercicios	22/12/2023	24/12/2023
HU024	Proporcionar retroalimentación de ejercicios	23/12/2023	24/12/2023

3.2 Diseño

La **Figura 83** muestra las funcionalidades de manera general a desarrollar en la tercera iteración.

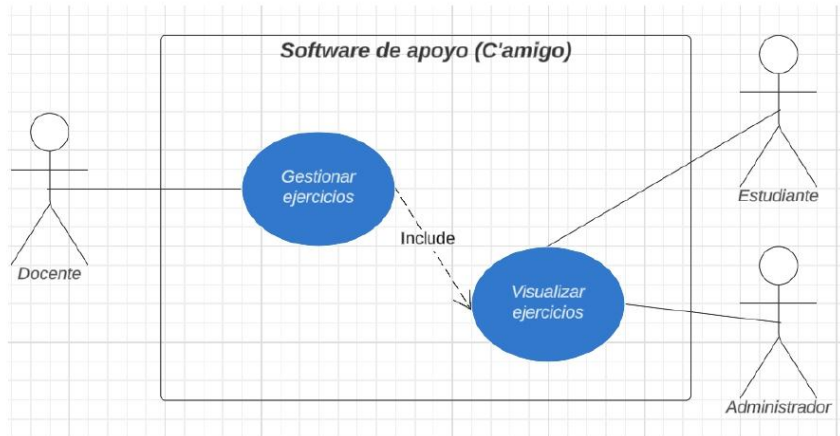


Figura 83. Casos de Uso general para la tercera iteración.

3.2.1 Diagramas BPMN

A continuación, se presentan los diagramas BPMN pertenecientes a cada Historia de Usuario a desarrollar en la tercera iteración.

3.2.1.1 HU017: Listar ejercicios

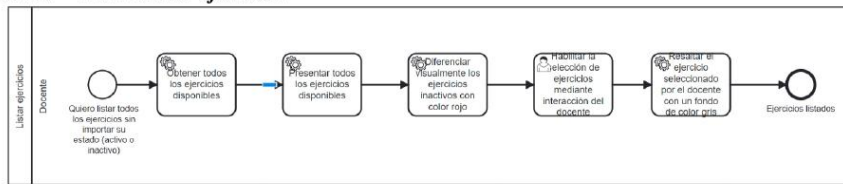


Figura 84. Diagrama BPMN de HU017.

3.2.1.2 HU018: Registrar ejercicios

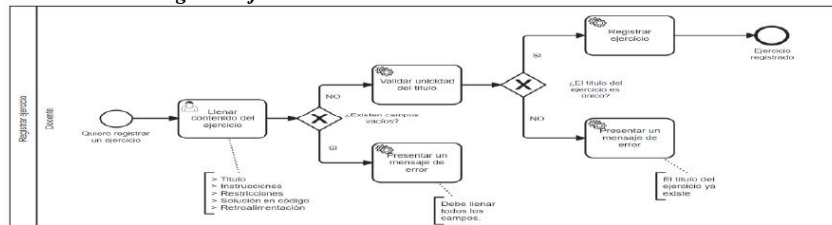


Figura 85. Diagrama BPMN de HU018

3.2.1.3 HU019: Editar ejercicios

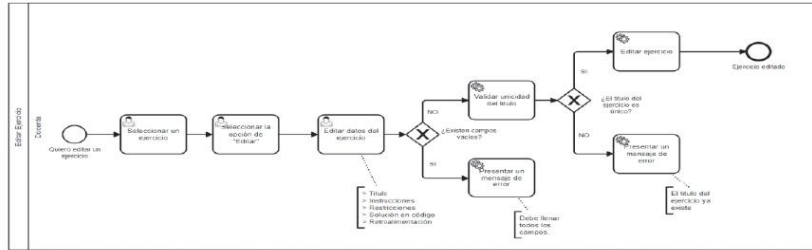


Figura 86. Diagrama BPMN de HU019.

3.2.1.4 HU020: Historial de cambios por ejercicio

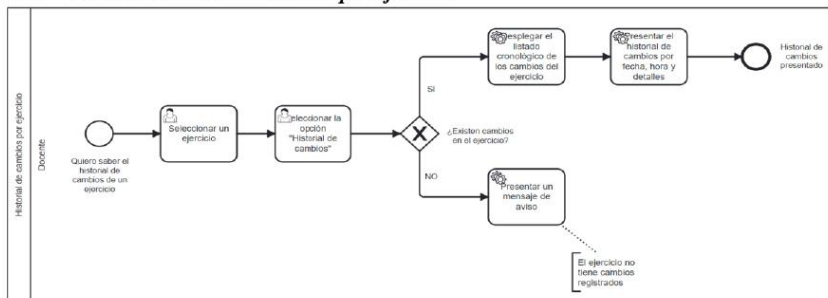


Figura 87. Diagrama BPMN de HU020.

3.2.1.5 HU021: Activar/Desactivar ejercicios

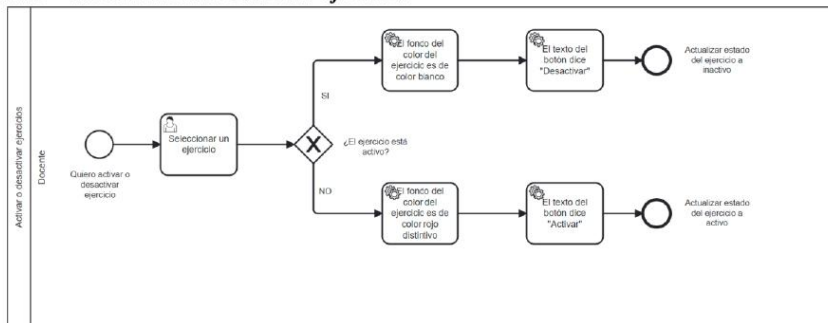


Figura 88. Diagrama BPMN de HU021.

3.2.1.6 HU022: Buscar ejercicios

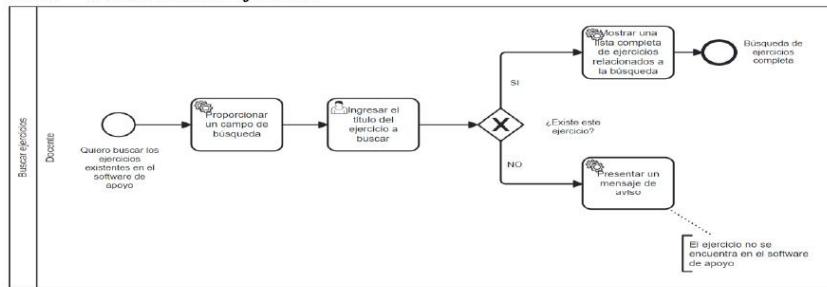


Figura 89. Diagrama BPMN de HU022.

3.2.1.7 HU023: Visualizar ejercicios

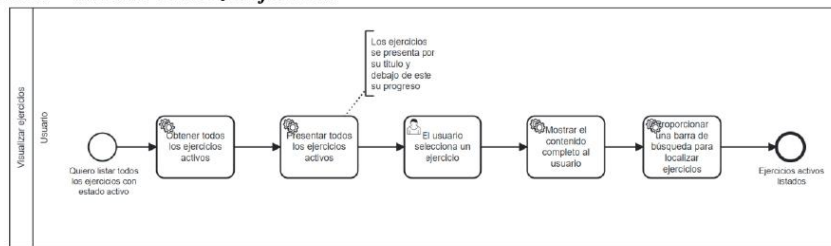


Figura 90. Diagrama BPMN de HU023.

3.2.1.8 HU024: Proporcionar retroalimentación de ejercicios

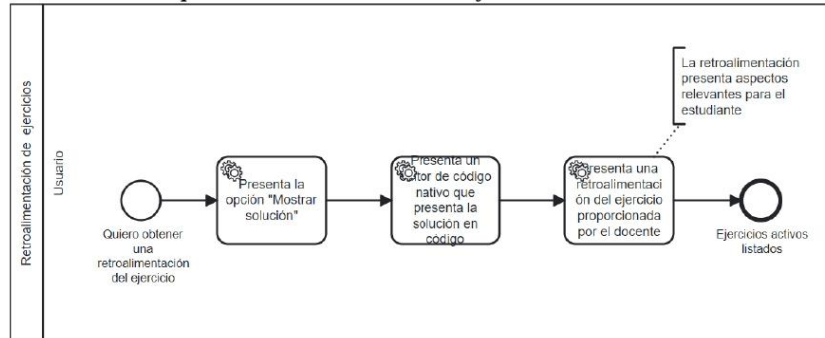


Figura 91. Diagrama BPMN de HU024.

3.2.2 Prototipos

A continuación, se presentan los prototipos pertenecientes a cada Historia de Usuario a desarrollar en la tercera iteración.

3.2.2.1 HU017: Listar ejercicios

Ejercicios

Ejercicios inactivos de rojo

Ejercicio 1
Ejercicio 2
Ejercicio 3
Ejercicio 4

Figura 92. Prototipo HU017.

3.2.2.2 HU018: Registrar ejercicios

C-Compañero

DOCENTE

Docente OCT 2023 - FEB 2024

Crear ejercicios

Título:

Instrucciones:

Restricciones:

Solución en código:

Retroalimentación:

Figura 93. Prototipo HU018.

3.2.2.3 HU019: Editar ejercicios

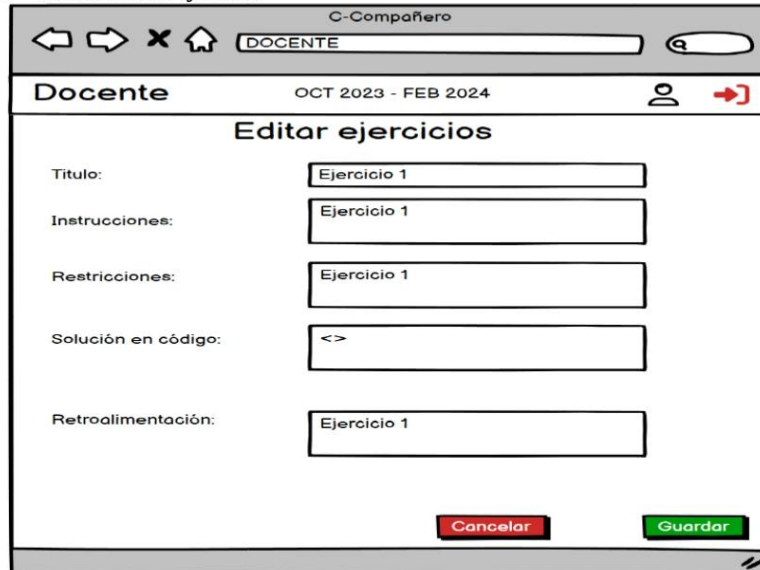
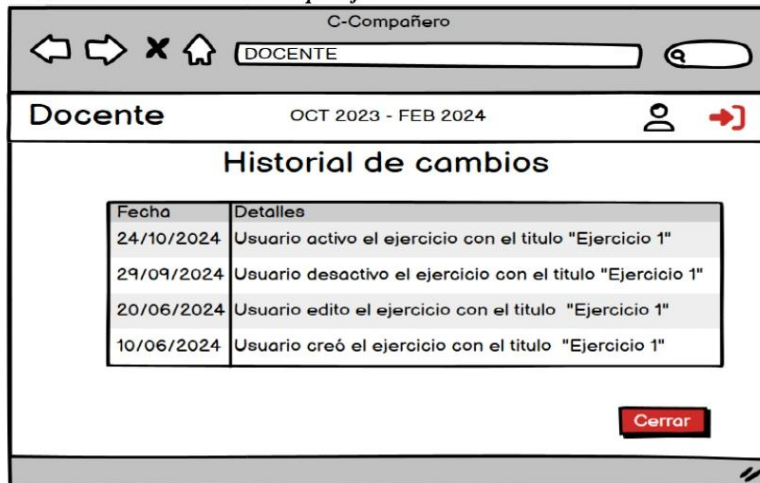


Figura 94. Prototipo HU019.

3.2.2.4 HU020: Historial de cambios por ejercicio.



Fecha	Detalles
24/10/2024	Usuario activo el ejercicio con el titulo "Ejercicio 1"
29/09/2024	Usuario desactivo el ejercicio con el titulo "Ejercicio 1"
20/06/2024	Usuario edito el ejercicio con el titulo "Ejercicio 1"
10/06/2024	Usuario creó el ejercicio con el titulo "Ejercicio 1"

Figura 95. Prototipo HU020



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

3.2.2.5 HU021: Activar/Desactivar ejercicios

Ejercicios

Ejercicios inactivos de rojo

Q Buscar tema

Ejercicio 1
Ejercicio 2
Ejercicio 3
Ejercicio 4

Crear Editar Activar/Desactivar Historial de cambios

Figura 96. Prototipo HU021.

3.2.2.6 HU022: Buscar ejercicios

Ejercicios

Ejercicios inactivos de rojo

Q Buscar tema

Ejercicio 1
Ejercicio 2
Ejercicio 3
Ejercicio 4

Crear Editar Activar/Desactivar Historial de cambios

Figura 97. Prototipo HU022.



3.2.2.7 HU023: Visualizar ejercicios

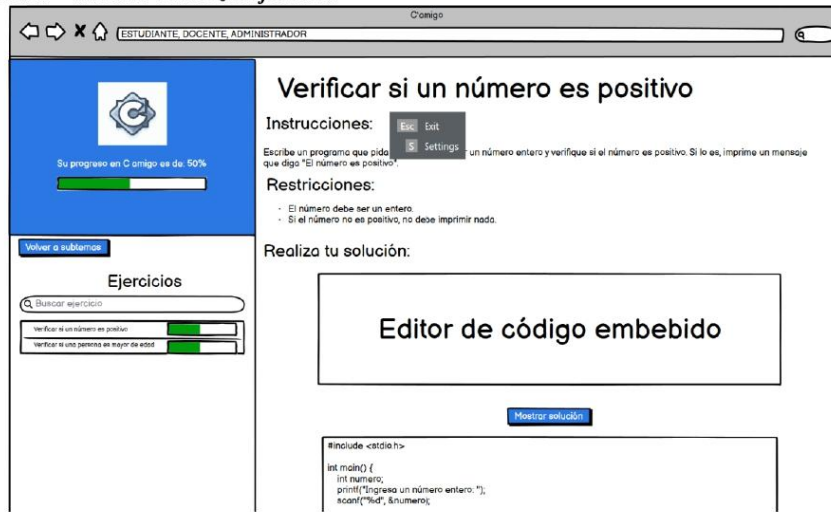


Figura 98. Prototipo HU023.

3.2.2.8 HU024: Proporcionar retroalimentación de ejercicios

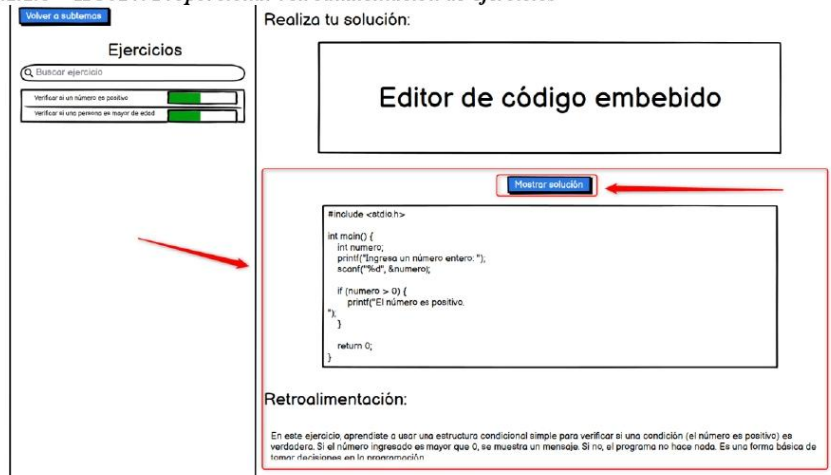


Figura 99. Prototipo HU024.

3.3 Codificación

A continuación, se presenta los fragmentos principales del código fuente desarrollados en la tercera iteración.

```
1 router_post('/listarEjercicios', (req, res) => {
2   let idSubtema = req.body.idSubtema;
3   let idUsuario = req.body.idUsuario;
4
5   let obtenerEjercicios;
6   if (req.body.metodo === "EjerciciosActivos") {
7     obtenerEjercicios =
8     "SELECT subtema_idEjercicio, subtema_idEjercicio_progreso, ejercicio_estado, ejercicio_instrucciones, ejercicio_restricciones, ejercicio_solucion, ejercicio_retroalimentacion, ejercicio_estado " +
9     "FROM subtema_ejercicio ";
10    "WHERE 2028 (180) ejercicio_idEjercicio = ejercicio_id +
11    "WHERE ejercicio_idSubtema = ? AND idUsuario = ? ";
12  } else {
13    obtenerEjercicios =
14    "SELECT subtema_idEjercicio, subtema_idEjercicio_progreso, ejercicio_estado, ejercicio_instrucciones, ejercicio_restricciones, ejercicio_solucion, ejercicio_retroalimentacion, ejercicio_estado " +
15    "FROM subtema_ejercicio ";
16    "WHERE 2028 (180) ejercicio_idEjercicio = ejercicio_id +
17    "WHERE ejercicio_idSubtema = ? AND idUsuario = ? ";
18  }
19  sql.ejecutarQuery(obtenerEjercicios, [idSubtema, idUsuario], (resultado) => {
20    if (resultado.length > 0) {
21      return req;
22    } else {
23      .status(404)
24      .send({ en: 1, m: "Ejercicios obtenidos, ejercicio: resultado });
25    } else {
26      return res.status(200).send({ en: -1, m: "No se encontraron ejercicios" });
27    }
28  });
29 });
```

Figura 100. Codificación HU017.

```
1 router_post('/registroEjercicio', [], (req, res) => {
2   let titulo = req.body.titulo;
3   let instrucciones = req.body.instrucciones;
4   let restricciones = req.body.restricciones;
5   let solucion = req.body.solucion;
6   let retroalimentacion = req.body.retroalimentacion;
7   let estado = 1;
8   let idSubtema = req.body.idSubtema;
9
10  const registrarEjercicio =
11  "INSERT INTO ejercicio (titulo, instrucciones, restricciones, solucion, retroalimentacion, estado, idSubtema) VALUES (?, ?, ?, ?, ?, ?, ?)";
12
13  sql.ejecutarQuerySQL(
14    registrarEjercicio,
15    [titulo, instrucciones, restricciones, solucion, retroalimentacion, estado, idSubtema],
16    (resultado) => {
17      if (resultado["affectedRows"] > 0) {
18        const idEjercicioInsertado = resultado["insertId"];
19
20        // segunda consulta para insertar en subtema_ejercicio por cada usuario
21        "INSERT INTO subtema_ejercicio (idUsuario, idEjercicio) SELECT id, ? FROM usuario";
22
23        sql.ejecutarQuerySQL(
24          obtenerEjercicio,
25          [idEjercicioInsertado],
26          (resultadoObtenerEjercicio) => {
27            if (resultadoObtenerEjercicio["affectedRows"] > 0) {
28              const obtenerCorreo = "SELECT email FROM usuario";
29
30              sql.ejecutarQuerySQL(
31                obtenerCorreo,
32                [],
33                (resultadoObtenerCorreo) => {
34                  if (resultadoObtenerCorreo.length > 0) {
35                    const mensaje = "Nuevo ejercicio creado en Codigo";
36
37                    // Para no crear un nuevo ejercicio en Codigo, el estado es 1 (no
38                    // disponible a la plataforma para practicar y mejorar los habilidades)
39                    // cuando se crea un nuevo ejercicio en la plataforma de Codigo
40                    // como comentario este correo es "SELECT email FROM usuario";
41                    // img src="http://descripcion.unl.edu.ec/images/logo_unl.png" alt="Logo Universidad Nacional de Loja" style="width:100px; height:auto;"/>
42
43                    resultadoObtenerCorreo.forEach(cuenta => {
44                      sql.ejecutarQuerySQL(
45                        "INSERT INTO correo (idUsuario, idEjercicio, asunto, mensaje)",
46                        [cuenta.id, idEjercicioInsertado, mensaje, mensaje],
47                        (resultadoInsertarCorreo) => {
48                          if (resultadoInsertarCorreo["affectedRows"] > 0) {
49                            return res.status(200).send({
50                              en: 1,
51                              m: "Se registro el ejercicio con éxito y se enviaron correos de notificación",
52                              idEjercicio: idEjercicioInsertado,
53                            });
54                          } else {
55                            return res.status(200).send({
56                              en: 1,
57                              m: "No se registro el ejercicio con éxito, pero no hay cuentas para notificar",
58                              idEjercicio: idEjercicioInsertado,
59                            });
60                          }
61                        });
62                      } else {
63                        return res.status(200).send({ en: -1, m: "No se pudo registrar en subtema_ejercicio porque faltan ejercicios o usuarios" });
64                      }
65                    });
66                  } else {
67                    return res.status(200).send({ en: 1, m: "No se pudo registrar el ejercicio" });
68                  }
69                });
70              } else {
71                return res.status(200).send({ en: 1, m: "No se pudo registrar el ejercicio" });
72              }
73            }
74          });
75    });
```

Figura 101. Codificación HU018.

```
1 router.post("/editarEjercicio", [], (req, res) => {
2   let id = req.body.id;
3   let titulo = req.body.titulo;
4   let instrucciones = req.body.instrucciones;
5   let restricciones = req.body.restricciones;
6   let solucion = req.body.solucion;
7   let retroalimentacion = req.body.retroalimentacion;
8   let estado = req.body.estado;
9   let ejercicioEditadoBackend = { id, titulo, instrucciones, restricciones, solucion, retroalimentacion, estado };
10
11  const editarEjercicio =
12    "UPDATE ejercicio SET titulo = ?, instrucciones = ?, restricciones = ?, solucion = ?, retroalimentacion = ?, estado = ? WHERE id = ?";
13  sql.ejecutarResSQL(
14    editarEjercicio,
15    [titulo, instrucciones, restricciones, solucion, retroalimentacion, estado, id],
16    (resultado) => {
17      //return res.status(200).send({result:resultado, titulo:titulo, descripcion:descripcion, id:id});
18      if (resultado["affectedRows"] > 0)
19        return res
20          .status(200)
21          .send({ en: 1, m: "Se editó el ejercicio con éxito", idEjercicio: id, ejercicioEditadoBackend });
22      return res.status(200).send({ en: -1, m: "No se pudo editar el ejercicio" });
23    }
24  );
25 });
26
```

Figura 102. Codificación HU019.

```
1 router.post("/registrarCambio", (req, res) => {
2   let tipoEntidad = req.body.tipoEntidad;
3   let idCrea = req.body.idCrea;
4   let idSubtema = req.body.idSubtema;
5   let idEjercicio = req.body.idEjercicio;
6   let idPregunta = req.body.idPregunta;
7   let detalles = req.body.detalles;
8   let idUsuario = req.body.idUsuario;
9
10  const registrarCambio =
11    "INSERT INTO historial (tipo_entidad, id_tema, id_subtema, id_ejercicio, id_pregunta, detalles, id_usuario) VALUES (?, ?, ?, ?, ?, ?, ?)";
12
13  sql.ejecutarResSQL(
14    registrarCambio,
15    [tipoEntidad, idCrea, idSubtema, idEjercicio, idPregunta, detalles, idUsuario],
16    (resultado) => {
17      if (resultado["affectedRows"] > 0) {
18        return res.status(200).send({
19          en: 1,
20          m: "Se registró el cambio con éxito",
21          idCambio: resultado["insertId"],
22        });
23      } else {
24        return res.status(200).send({ en: -1, m: "No se pudo registrar el cambio" });
25      }
26    }
27  );
28 });
```

Figura 103. Codificación HU020.

```
1 router.post("/activarDesactivarEjercicio", [], (req, res) => {
2   let id = req.body.id;
3   let estado = req.body.estado;
4   const actualizarEjercicio = "UPDATE ejercicio SET estado = ? WHERE id = ?";
5   sql.ejecutarResSQL(actualizarEjercicio, [estado, id], (resultado) => {
6     //return res.status(200).send({result:resultado, titulo:titulo, descripcion:descripcion, id:id})
7     if [resultado["affectedRows"] > 0]
8       return res
9         .status(200)
10        .send({ en: 1, m: "Cambio el estado del ejercicio", idEjercicio: id });
11    return res
12      .status(200)
13      .send({ en: -1, m: "No se pudo cambiar el estado del ejercicio" });
14  });
15 });
```

Figura 104. Codificación HU021.

```
1 router.post("/buscarTemas", (req, res) => {
2   const busqueda = req.body.busqueda;
3   const idUsuario = req.body.idUsuario;
4
5   // Query para buscar temas que coincidan con el término de búsqueda
6   const buscarTemas = `
7   SELECT
8     usuario_tema.idTema,
9     usuario_tema.progreso,
10    tema.titulo,
11    tema.objetivos,
12    tema.descripcion,
13    tema.recursos,
14    tema.estado
15  FROM ${DB}.usuario_tema
16  INNER JOIN ${DB}.tema ON usuario_tema.idTema = tema.id
17  WHERE usuario_tema.idUsuario = ?
18  AND (
19    tema.titulo LIKE CONCAT('%', ?, '%')
20    OR tema.descripcion LIKE CONCAT('%', ?, '%')
21    OR tema.objetivos LIKE CONCAT('%', ?, '%')
22  )
23  ORDER BY tema.titulo ASC;
24  `;
25
26  sql.ejecutarResSQL(
27    buscarTemas,
28    [idUsuario, busqueda, busqueda, busqueda],
29    (resultado) => {
30      if (resultado.length > 0) {
31        return res.status(200).send({
32          en: 1,
33          m: "Contenido encontrado",
34          temas: resultado
35        });
36      } else {
37        return res.status(200).send({
38          en: -1,
39          m: "No se encontro contenido que coincida con la búsqueda"
40        });
41      }
42    });
43 });
44 });
```

Figura 105. Codificación HU022.

```

1 router.post('/listarEjercicios', (req, res) => {
2   let idSubtema = req.body.idSubtema;
3   let idUsuario = req.body.idUsuario;
4
5   let obtenerEjercicios =
6     if (req.body.mensaje === "EjerciciosActivos") {
7       obtenerEjercicios =
8         SELECT subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio
9         FROM EJERCICIO
10        WHERE subtema_idEjercicio = subtema_idEjercicio +
11        WHERE ejercicio_idSubtema = ? AND idUsuario = ? AND estado = 1;
12      } else {
13        obtenerEjercicios =
14          SELECT subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio
15          FROM EJERCICIO
16          WHERE subtema_idEjercicio = subtema_idEjercicio +
17          WHERE ejercicio_idSubtema = ? AND idUsuario = ?;
18        }
19      }
20      let ejecutarRequiere = (obtenerEjercicios, [idSubtema, idUsuario], (resultado) => {
21        if (resultado.length > 0) {
22          return res
23            .status(200)
24            .send({ en: 1, m: "Ejercicios obtenidos", ejercicios: resultado });
25        } else {
26          return res.status(200).send({ en: -1, m: "No se encontraron ejercicios" });
27        }
28      });
29    });
30  });

```

Figura 106. Codificación HU023.

```

1 router.post('/listarEjercicios', (req, res) => {
2   let idSubtema = req.body.idSubtema;
3   let idUsuario = req.body.idUsuario;
4
5   let obtenerEjercicios =
6     if (req.body.mensaje === "EjerciciosActivos") {
7       obtenerEjercicios =
8         SELECT subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio
9         FROM EJERCICIO
10        WHERE subtema_idEjercicio = subtema_idEjercicio +
11        WHERE ejercicio_idSubtema = ? AND idUsuario = ? AND estado = 1;
12      } else {
13        obtenerEjercicios =
14          SELECT subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio, subtema_idEjercicio
15          FROM EJERCICIO
16          WHERE subtema_idEjercicio = subtema_idEjercicio +
17          WHERE ejercicio_idSubtema = ? AND idUsuario = ?;
18        }
19      }
20      let ejecutarRequiere = (obtenerEjercicios, [idSubtema, idUsuario], (resultado) => {
21        if (resultado.length > 0) {
22          return res
23            .status(200)
24            .send({ en: 1, m: "Ejercicios obtenidos", ejercicios: resultado });
25        } else {
26          return res.status(200).send({ en: -1, m: "No se encontraron ejercicios" });
27        }
28      });
29    });
30  });

```

Figura 107. Codificación HU024.

3.4 Pruebas

La Tabla 6 muestra las pruebas unitarias a desarrollar en la tercera iteración.

Tabla 6. Pruebas unitarias a ejecutar en la tercera iteración

ID	Descripción	Datos de Entrada	Salida Esperada	Resultado
PU028	Listar ejercicios activos	<ul style="list-style-type: none"> ➤ idSubtema: 37, ➤ idUsuario: 47, ➤ mensaje: "ejerciciosActivos" 	Ejercicios obtenidos	Aprobada
PU029	Listar ejercicios cuando existe ningún subtema	<ul style="list-style-type: none"> ➤ idSubtema: 9999, ➤ idUsuario: 47, ➤ mensaje: "ejerciciosActivos" 	No se encontraron ejercicios	Aprobada
PU030	Listar ejercicios activos e inactivos.	<ul style="list-style-type: none"> ➤ idSubtema: 37, ➤ idUsuario: 47 	Ejercicios obtenidos	Aprobada
PU031	Registrar un nuevo ejercicio	<ul style="list-style-type: none"> ➤ titulo: 'Nuevo Ejercicio de Prueba', ➤ instrucciones: 'Instrucciones del ejercicio de prueba', 	Se registró el subtema con éxito y se enviaron correos de notificación	Aprobada



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

		<ul style="list-style-type: none"> ➤ restricciones: 'Restricciones del ejercicio de prueba', ➤ solucion: 'console.log("Solución de prueba");', ➤ retroalimentacion: 'Retroalimentación del ejercicio de prueba', ➤ idSubtema: 37 		
PU032	Editar un ejercicio existente	<ul style="list-style-type: none"> ➤ id: 27, ➤ titulo: 'Título editado del ejercicio', ➤ instrucciones: 'Instrucciones editadas del ejercicio', ➤ restricciones: 'Restricciones editadas del ejercicio', ➤ solucion: 'console.log("Solución editada");', ➤ retroalimentacion: 'Retroalimentación editada del ejercicio', ➤ estado: 1 	Se editó el ejercicio con éxito	Aprobada
PU033	Editar un ejercicio inexistente.	<ul style="list-style-type: none"> ➤ id: 99999, ➤ titulo: 'Título editado del ejercicio', ➤ instrucciones: 'Instrucciones editadas del ejercicio', ➤ restricciones: 'Restricciones editadas del ejercicio', ➤ solucion: 'console.log("Solución editada");', ➤ retroalimentacion: 'Retroalimentación editada del ejercicio', ➤ estado: 1 	No se pudo editar el ejercicio	Aprobada
PU034	Registrar un nuevo cambio en ejercicio	<ul style="list-style-type: none"> ➤ tipoEntidad: "ejercicio", ➤ idTema: null, ➤ idSubtema: 37, ➤ idEjercicio: 27, ➤ idPregunta: null, ➤ detalles: "Se actualizó el contenido del ejercicio", ➤ idUsuario: 47 	Se registró el cambio con éxito	Aprobada
PU035	Activar/Desactivar ejercicio	<ul style="list-style-type: none"> ➤ id: 27, ➤ estado: 1 	Activar o desactivar ejercicio	Aprobada
PU036	Buscar contenido con término existente	<ul style="list-style-type: none"> ➤ idUsuario: 27, ➤ busqueda: "Título editado del ejercicio". 	Contenido encontrado	Aprobada



PU037	Visualizar ejercicios	➤	N/A	Ejercicios listados	activos	Aprobada
PU038	Mostrar la retroalimentación del ejercicio	➤	N/A	Solución mostrada al usuario		Aprobada

A continuación, se presentan las pruebas unitarias desarrolladas en la tercera iteración.

```

1 describe('Test de listado de ejercicios', () => {
2   it('Listar ejercicios activos del subtema', async () => {
3     const response = await api
4       .post(rutaEjercicios + '/listarEjercicios')
5       .send(bodyListarActivos);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Ejercicios obtenidos");
10    expect(Array.isArray(response.body.ejercicios)).toBe(true);
11  });
12
13  it('Intento de listar ejercicios cuando no hay ninguno', async () => {
14    const bodySinEjercicios = {
15      idSubtema: 999, // ID de subtema que no tiene ejercicios
16      idUsuario: 47,
17      mensaje: "ejerciciosActivos"
18    };
19
20    const response = await api
21      .post(rutaEjercicios + '/listarEjercicios')
22      .send(bodySinEjercicios);
23
24    expect(response.statusCode).toBe(200);
25    expect(response.body.en).toBe(-1);
26    expect(response.body.m).toBe("No se encontraron ejercicios");
27  });
28
29  it('Listar todos los ejercicios del subtema (activos e inactivos)', async () => {
30    const bodyTodosEjercicios = {
31      idSubtema: 37,
32      idUsuario: 47
33    };
34
35    const response = await api
36      .post(rutaEjercicios + '/listarEjercicios')
37      .send(bodyTodosEjercicios);
38
39    expect(response.statusCode).toBe(200);
40    expect(response.body.en).toBe(1);
41    expect(response.body.m).toBe("Ejercicios obtenidos");
42    expect(Array.isArray(response.body.ejercicios)).toBe(true);
43  });
44 });

```

Figura 108. Pruebas unitarias (PU028, PU029, PU030), listar ejercicios.



```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionTres
> camigo@1.0.0 testIteracionTres
> cross-env BD=camigo_prueba jest --verbose --silent tests/ejercicio/ejercicio.test.js

PASS tests/ejercicio/ejercicio.test.js
  Test de listado de ejercicios
    ✓ Listar ejercicios activos del subtema (82 ms)
    ✓ Intento de listar ejercicios cuando no hay ninguno (8 ms)
    ✓ Listar todos los ejercicios del subtema (activos e inactivos) (5 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.094 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 109. Ejecución pruebas unitarias (PU028, PU029, PU030), listar ejercicios.

```
1 describe('Test de registro de ejercicios', () => {
2   it('Registrar un nuevo ejercicio con notificaciones', async () => {
3     const response = await api
4       .post(rutaEjercicios + '/registrarEjercicio')
5       .send(bodyRegistrar);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Se registró el ejercicio con éxito y se enviaron correos de notificación");
10    expect(response.body.idEjercicio).toBeDefined();
11    expect(typeof response.body.idEjercicio).toBe('number');
12  });
13 });
```

Figura 110. Pruebas unitarias (PU031), registrar ejercicios.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionTres
> camigo@1.0.0 testIteracionTres
> cross-env BD=camigo_prueba jest --verbose --silent tests/ejercicio/ejercicio.test.js

PASS tests/ejercicio/ejercicio.test.js
  Test de registro de ejercicios
    ✓ Registrar un nuevo ejercicio con notificaciones (180 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.812 s
Jest did not exit one second after the test run has completed.
```

Figura 111. Ejecución pruebas unitarias (PU031), registrar ejercicios.

```
1 describe('Test de edición de ejercicios', () => {
2   it('Editar un ejercicio existente', async () => {
3     const response = await api
4       .post(rutaEjercicios + '/editarEjercicio')
5       .send(bodyEditar);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Se editó el ejercicio con éxito");
10    expect(response.body.idEjercicio).toBe(bodyEditar.id);
11    expect(response.body.ejercicioEditadoBackend).toEqual(bodyEditar);
12  });
13
14  it('Intentar editar un ejercicio inexistente', async () => {
15    const bodyEjercicioInexistente = {
16      ...bodyEditar,
17      id: 999 // ID que no existe en la base de datos
18    };
19
20    const response = await api
21      .post(rutaEjercicios + '/editarEjercicio')
22      .send(bodyEjercicioInexistente);
23
24    expect(response.statusCode).toBe(200);
25    expect(response.body.en).toBe(-1);
26    expect(response.body.m).toBe("No se pudo editar el ejercicio");
27  });
28 });
```

Figura 112. Pruebas unitarias (PU032, PU033), editar ejercicios.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionTres
> camigo@1.0.0 testIteracionTres
> cross-env BD=camigo_prueba jest --verbose --silent tests/ejercicio/ejercicio.test.js

PASS tests/ejercicio/ejercicio.test.js
  Test de edición de ejercicios
    ✓ Editar un ejercicio existente (89 ms)
    ✓ Intentar editar un ejercicio inexistente (12 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.262 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 113. Ejecución pruebas unitarias (PU032, PU033), editar ejercicios.

```
1 describe('Test de registro de cambios', () => {
2   // Prueba para registrar un cambio exitosamente
3   it('Registrar un nuevo cambio en tema', async () => {
4     const response = await api
5       .post(rutaHistorial + '/registrarCambio')
6       .send(bodyRegistrarCambio);
7
8     expect(response.statusCode).toBe(200);
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Se registró el cambio con éxito");
11    expect(response.body.idCambio).toBeDefined();
12    expect(typeof response.body.idCambio).toBe('number');
13  });
14 });
```

Figura 114. Pruebas unitarias (PU034), historial de cambios en ejercicios.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionTres
> camigo@1.0.0 testIteracionTres
> cross-env BD=camigo_prueba jest --verbose --silent tests/ejercicio/ejercicio.test.js

PASS tests/ejercicio/ejercicio.test.js
  Test de registro de cambios en ejercicio
    ✓ Registrar un nuevo cambio en ejercicio (100 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.511 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 115. Ejecución pruebas unitarias (PU034), historial de cambios en ejercicios.

```
1 describe('Pruebas para activar/desactivar ejercicio', () => {
2   it('deberia activar un ejercicio correctamente', async () => {
3     const response = await api
4       .post(rutaEjercicios + '/activarDesactivarEjercicio')
5       .send(bodyActivar_Desactivar)
6       .expect(200)
7       .expect('Content-Type', /application\/json/);
8
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Cambio el estado del ejercicio");
11    expect(response.body.idEjercicio).toBe(bodyActivar_Desactivar.id);
12  });
13
14  it('deberia desactivar un ejercicio correctamente', async () => {
15    bodyActivar_Desactivar.estado = -1;
16
17    const response = await api
18      .post(rutaEjercicios + '/activarDesactivarEjercicio')
19      .send(bodyActivar_Desactivar)
20      .expect(200)
21      .expect('Content-Type', /application\/json/);
22
23    expect(response.body.en).toBe(1);
24    expect(response.body.m).toBe("Cambio el estado del ejercicio");
25    expect(response.body.idEjercicio).toBe(bodyActivar_Desactivar.id);
26  });
27
28  it('deberia manejar el intento de activar/desactivar un ejercicio inexistente', async () => {
29    const bodyEjercicioInexistente = {
30      ...bodyActivar_Desactivar,
31      id: 999 // ID que no existe
32    };
33
34    const response = await api
35      .post(rutaEjercicios + '/activarDesactivarEjercicio')
36      .send(bodyEjercicioInexistente)
37      .expect(200)
38      .expect('Content-Type', /application\/json/);
39
40    expect(response.body.en).toBe(-1);
41    expect(response.body.m).toBe("No se pudo cambiar el estado del ejercicio");
42  });
43 });
```

Figura 116. Pruebas unitarias (PU035), activar/desactivar ejercicios.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionTres
> camigo@1.0.0 testIteracionTres
> cross-env BD=camigo_prueba jest --verbose --silent tests/ejercicio/ejercicio.test.js

PASS tests/ejercicio/ejercicio.test.js
  Pruebas para activar/desactivar ejercicio
    ✓ deberia activar un ejercicio correctamente (88 ms)
    ✓ deberia desactivar un ejercicio correctamente (18 ms)
    ✓ deberia manejar el intento de activar/desactivar un ejercicio inexistente (5 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.197 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 117. Ejecución pruebas unitarias (PU035), activar/desactivar ejercicios.

```
1 describe('Test de búsqueda de temas', () => {
2   it('Buscar contenido con término existente', async () => {
3     const response = await api
4       .post(rutaTemas + '/buscarTemas')
5       .send(bodyBusquedaValida);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Contenido encontrado");
10    expect(Array.isArray(response.body.temas)).toBe(true);
11    expect(response.body.temas.length).toBeGreaterThan(0);
12
13    // Verificar la estructura de cada tema encontrado
14    const primerTema = response.body.temas[0];
15    expect(primerTema).toHaveProperty('idTema');
16    expect(primerTema).toHaveProperty('progreso');
17    expect(primerTema).toHaveProperty('titulo');
18    expect(primerTema).toHaveProperty('objetivos');
19    expect(primerTema).toHaveProperty('descripcion');
20    expect(primerTema).toHaveProperty('recursos');
21    expect(primerTema).toHaveProperty('estado');
22  });
23 });
24
```

Figura 118. Pruebas unitarias (PU036), buscar ejercicios.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Test de búsqueda de temas
    ✓ Buscar contenido con término existente (86 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 1.351 s, estimated 3 s
Jest did not exit one second after the test run has completed.
```

Figura 119. Ejecución pruebas unitarias (PU036), buscar ejercicios.

```
1 describe('Test de listado de ejercicios', () => {
2   it('listar ejercicios activos del subtema', async () => {
3     const response = await api
4       .post(rutaEjercicios + '/listarEjercicios')
5       .send(bodyListarActivos);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Ejercicios obtenidos");
10    expect(Array.isArray(response.body.ejercicios)).toBe(true);
11  });
12
13  it('Intento de listar ejercicios cuando no hay ninguno', async () => {
14    const bodySinEjercicios = {
15      idSubtema: 999, // ID de subtema que no tiene ejercicios
16      idUsuario: 47,
17      mensaje: "ejerciciosActivos"
18    };
19
20    const response = await api
21      .post(rutaEjercicios + '/listarEjercicios')
22      .send(bodySinEjercicios);
23
24    expect(response.statusCode).toBe(200);
25    expect(response.body.en).toBe(-1);
26    expect(response.body.m).toBe("No se encontraron ejercicios");
27  });
28
29  it('listar todos los ejercicios del subtema (activos e inactivos)', async () => {
30    const bodyTodosEjercicios = {
31      idSubtema: 37,
32      idUsuario: 47
33    };
34
35    const response = await api
36      .post(rutaEjercicios + '/listarEjercicios')
37      .send(bodyTodosEjercicios);
38
39    expect(response.statusCode).toBe(200);
40    expect(response.body.en).toBe(1);
41    expect(response.body.m).toBe("Ejercicios obtenidos");
42    expect(Array.isArray(response.body.ejercicios)).toBe(true);
43  });
44 });
45
```

Figura 120. Pruebas unitarias (PU037), visualizar ejercicios.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Test de búsqueda de temas
    ✓ Buscar contenido con término existente (86 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.351 s, estimated 3 s
Jest did not exit one second after the test run has completed.
```

Figura 121. Ejecución pruebas unitarias (PU037), visualizar ejercicios.

```
1 describe('Test de listado de ejercicios', () => {
2   it('Listar ejercicios activos del subtema', async () => {
3     const response = await api
4       .post(rutaEjercicios + '/listarEjercicios')
5       .send(bodyListarActivos);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Ejercicios obtenidos");
10    expect(Array.isArray(response.body.ejercicios)).toBe(true);
11  });
12
13  it('Intento de listar ejercicios cuando no hay ninguno', async () => {
14    const bodySinEjercicios = {
15      idSubtema: 999, // ID de subtema que no tiene ejercicios
16      idUsuario: 47,
17      mensaje: "ejerciciosActivos"
18    };
19
20    const response = await api
21      .post(rutaEjercicios + '/listarEjercicios')
22      .send(bodySinEjercicios);
23
24    expect(response.statusCode).toBe(200);
25    expect(response.body.en).toBe(-1);
26    expect(response.body.m).toBe("no se encontraron ejercicios");
27  });
28
29  it('Listar todos los ejercicios del subtema (activos e inactivos)', async () => {
30    const bodyTodosEjercicios = {
31      idSubtema: 37,
32      idUsuario: 47
33    };
34
35    const response = await api
36      .post(rutaEjercicios + '/listarEjercicios')
37      .send(bodyTodosEjercicios);
38
39    expect(response.statusCode).toBe(200);
40    expect(response.body.en).toBe(1);
41    expect(response.body.m).toBe("Ejercicios obtenidos");
42    expect(Array.isArray(response.body.ejercicios)).toBe(true);
43  });
44 });
45
```

Figura 122. Pruebas unitarias (PU038), mostrar retroalimentación de ejercicios.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js
PASS tests/tema/tema.test.js
  Test de búsqueda de temas
    ✓ Buscar contenido con término existente (86 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.351 s, estimated 3 s
Jest did not exit one second after the test run has completed.
```

Figura 123. Ejecución pruebas unitarias (PU038), mostrar retroalimentación de ejercicios

4 Cuarta Iteración

4.1 Planificación

La **Tabla 7** muestra las actividades específicas a desarrollar en la cuarta iteración.

Tabla 7. Historias de Usuario a desarrollar en la cuarta iteración.

HU #	Historia de usuario	Fecha de creación	Fecha de entrega
HU025	Listar preguntas de control	19/12/2023	05/01/2024
HU026	Registrar pregunta de control	23/12/2023	05/01/2024
HU027	Editar pregunta de control	26/12/2024	05/01/2024
HU028	Historial de cambios por pregunta de control	28/12/2024	05/01/2024
HU029	Activar/Desactivar pregunta de control	30/12/2024	05/01/2024
HU030	Buscar pregunta de control	01/01/2024	05/01/2024
HU031	Visualizar pregunta de control	03/01/2024	05/01/2024
HU032	Responder pregunta de control	04/01/2024	05/01/2024

4.2 Diseño

La **Figura 47** muestra las funcionalidades de manera general a desarrollar en la cuarta iteración.

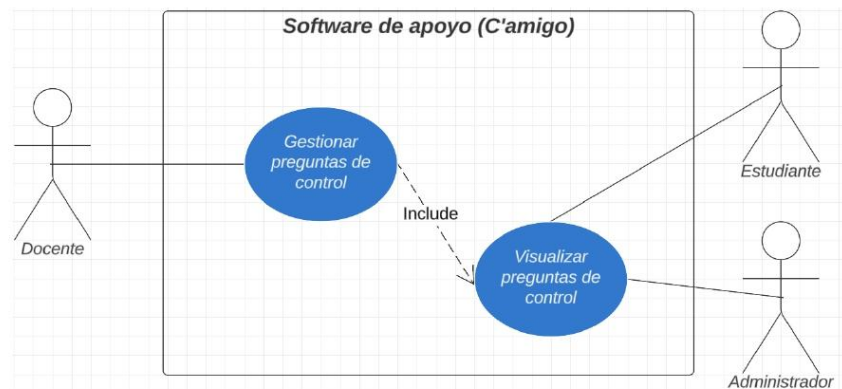


Figura 124. Casos de Uso general para la cuarta iteración.

4.2.1 Diagramas BPMN

A continuación, se presentan los diagramas BPMN pertenecientes a cada Historia de Usuario a desarrollar en la cuarta iteración.

4.2.1.1 HU025: Listar preguntas de control

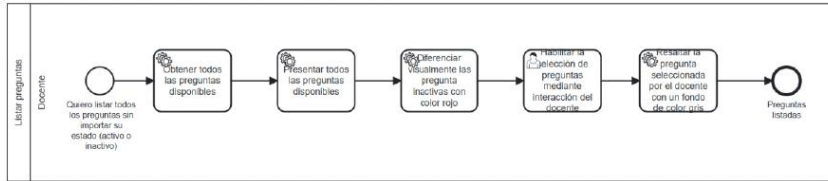


Figura 125. Diagrama BPMN de HU025.

4.2.1.2 HU026: Listar preguntas de control

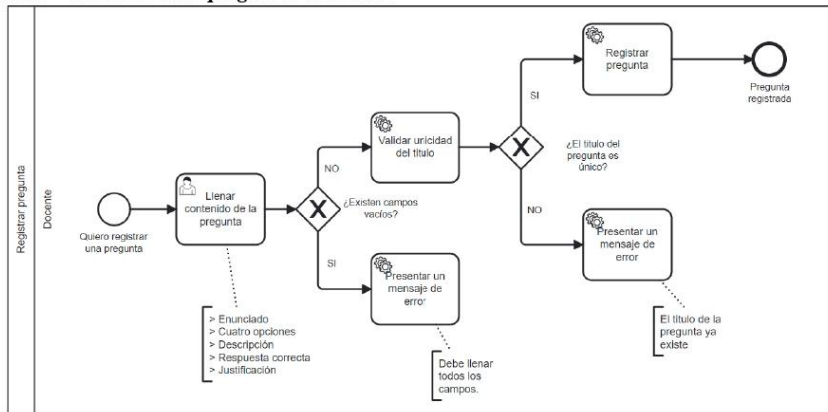


Figura 126. Diagrama BPMN de HU026.

4.2.1.3 HU027: Listar preguntas de control

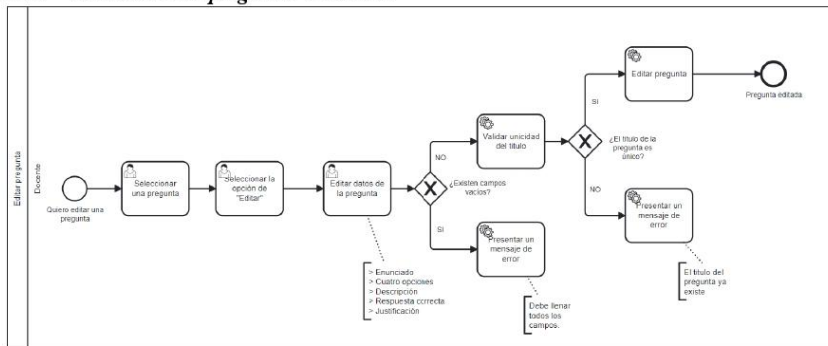


Figura 127. Diagrama BPMN de HU027.

4.2.1.4 HU028: Listar preguntas de control

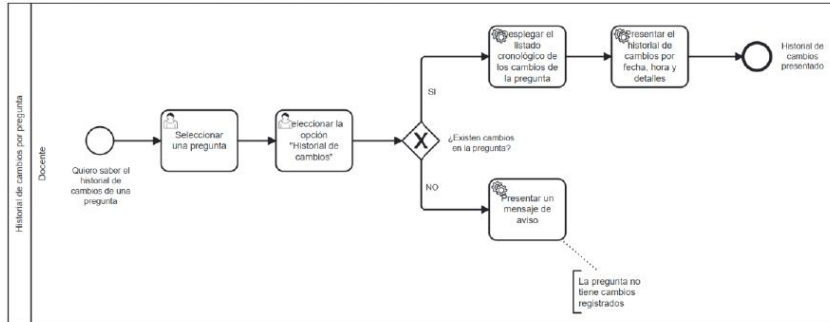


Figura 128. Diagrama BPMN de HU028.

4.2.1.5 HU029: Listar preguntas de control

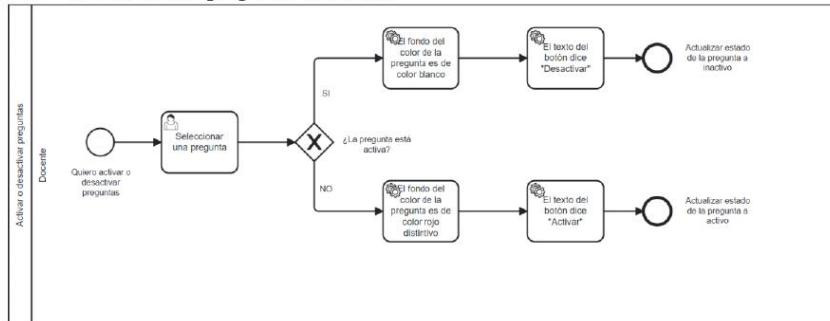


Figura 129. Diagrama BPMN de HU029.

4.2.1.6 HU030: Listar preguntas de control

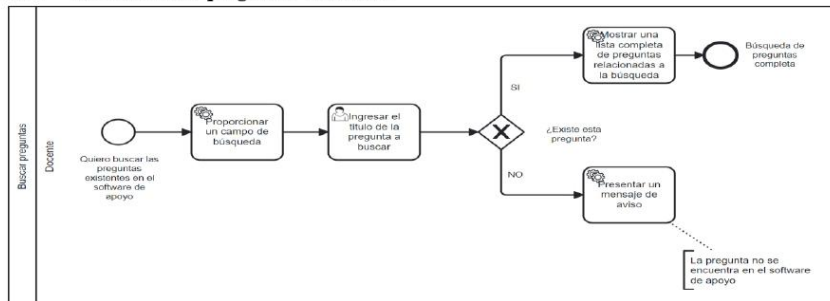


Figura 130. Diagrama BPMN de HU030.

4.2.1.7 HU031: Listar preguntas de control

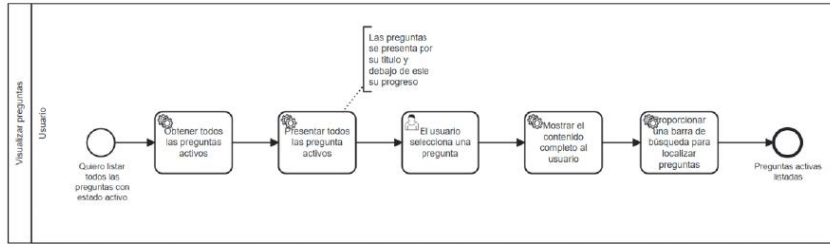


Figura 131. Diagrama BPMN de HU031.

4.2.1.8 HU032: Listar preguntas de control

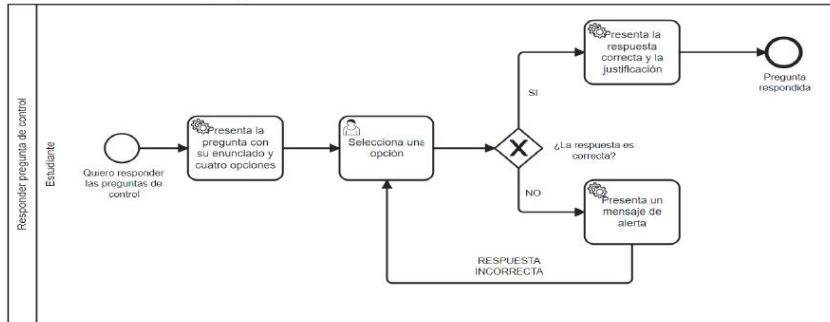


Figura 132. Diagrama BPMN de HU032.

4.2.2 Prototipos

A continuación, se presentan los prototipos pertenecientes a cada Historia de Usuario a desarrollar en la cuarta iteración.

4.2.2.1 HU025: Listar preguntas de control

Preguntas de control



Figura 133. Prototipo de HU025.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

4.2.2.2 HU026: Listar preguntas de control

C-Compañero
DOCENTE

Docente OCT 2023 - FEB 2024

Crear preguntas

Enunciado:

Primera opción:

Segunda opción:

Tercera opción:

Cuarta opción:

Respuesta correcta:
Opción b
Opción c
Opción d

Justificación:

Cancelar Guardar

Figura 134. Prototipo de HU026.

4.2.2.3 HU027: Listar preguntas de control

C-Compañero
DOCENTE

Docente OCT 2023 - FEB 2024

Editar preguntas

Enunciado:

Primera opción:

Segunda opción:

Tercera opción:

Cuarta opción:

Respuesta correcta:
Opción b
Opción c
Opción d

Justificación:

Cancelar Guardar

Figura 135. Prototipo de HU027.

4.2.2.4 HU028: Listar preguntas de control



Figura 136. Prototipo de HU028.

4.2.2.5 HU029: Listar preguntas de control

Preguntas de control

Preguntas inactivas de rojo

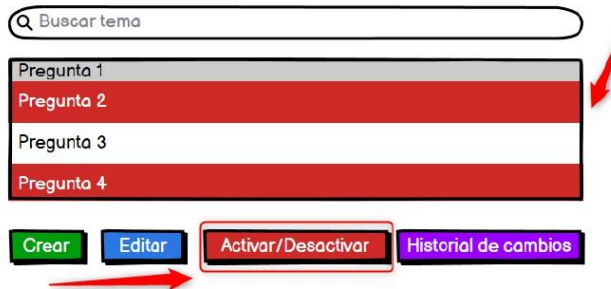


Figura 137. Prototipo de HU029.



4.2.2.6 HU030: Listar preguntas de control

Preguntas de control

Preguntas inactivas de rojo

Q Buscar tema

Pregunta 1
Pregunta 2
Pregunta 3
Pregunta 4

Crear Editar Activar/Desactivar Historial de cambios

Figura 138. Prototipo de HU030.

4.2.2.7 HU031: Listar preguntas de control

ESTUDIANTE, DOCENTE, ADMINISTRADOR

¿Qué sucede si el número ingresado por el usuario es 0?

Selecciona la opción correcta:

- Imprime "El número es positivo".
- No imprime nada.
- Imprime "El número es negativo".
- Imprime un error.

Su progreso en C amigo es de: 50%

Volver a ejercicios

Preguntas de control

Q Buscar pregunta

¿Qué sucede si el número ingresado por el usuario es 0?

¿Qué condición evalúa el programa para decidir si imprime el mensaje?

Figura 139. Prototipo de HU031.



4.2.2.8 HU032: Listar preguntas de control

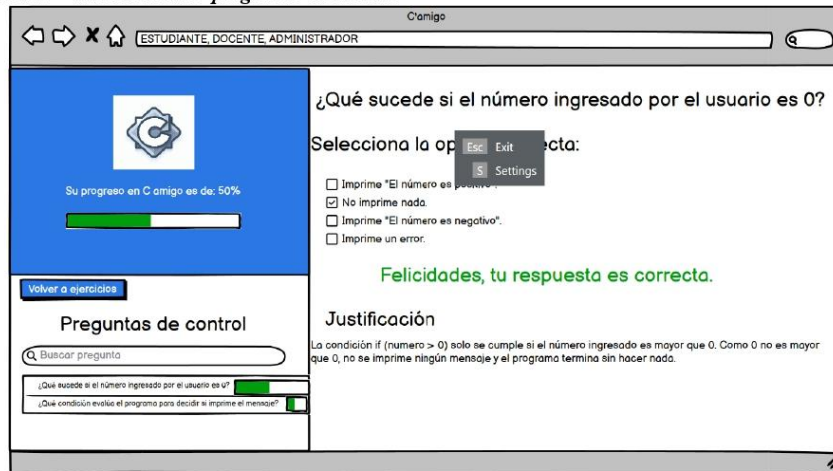


Figura 140. Prototipo de HU032.

4.3 Codificación

A continuación, se presenta los fragmentos principales del código fuente desarrollados en la cuarta iteración.

```
1 function mostrarPreguntas() {
2     const preguntas = [
3         { id: 1, pregunta: "¿Qué sucede si el número ingresado por el usuario es 0?", opciones: ["Imprime 'El número es positivo'", "No imprime nada", "Imprime 'El número es negativo'", "Imprime un error."], respuesta_correcta: "No imprime nada", justificacion: "La condición if (numero > 0) solo se cumple si el número ingresado es mayor que 0. Como 0 no es mayor que 0, no se imprime ningún mensaje y el programa termina sin hacer nada." },
4         { id: 2, pregunta: "¿Qué sucede si el número ingresado por el usuario es 1?", opciones: ["Imprime 'El número es positivo'", "No imprime nada", "Imprime 'El número es negativo'", "Imprime un error."], respuesta_correcta: "Imprime 'El número es positivo'", justificacion: "La condición if (numero > 0) se cumple porque 1 es mayor que 0, por lo tanto se ejecuta el código dentro de la condición." },
5         { id: 3, pregunta: "¿Qué sucede si el número ingresado por el usuario es -1?", opciones: ["Imprime 'El número es positivo'", "No imprime nada", "Imprime 'El número es negativo'", "Imprime un error."], respuesta_correcta: "Imprime 'El número es negativo'", justificacion: "La condición if (numero > 0) no se cumple porque -1 es menor que 0, por lo tanto no se ejecuta el código dentro de la condición." },
6         { id: 4, pregunta: "¿Qué sucede si el número ingresado por el usuario es 0.5?", opciones: ["Imprime 'El número es positivo'", "No imprime nada", "Imprime 'El número es negativo'", "Imprime un error."], respuesta_correcta: "No imprime nada", justificacion: "La condición if (numero > 0) se cumple porque 0.5 es mayor que 0, por lo tanto se ejecuta el código dentro de la condición." }
7     ];
8     const preguntas_html = preguntas.map(pregunta => `
9         <div class="pregunta">
10             <p>¿${pregunta.pregunta}</p>
11             <ul style="list-style-type: none;>
12                 <li><input type="radio"/> ${pregunta.opciones[0]}</li>
13                 <li><input type="radio"/> ${pregunta.opciones[1]}</li>
14                 <li><input type="radio"/> ${pregunta.opciones[2]}</li>
15                 <li><input type="radio"/> ${pregunta.opciones[3]}</li>
16             </ul>
17             <p>Respuesta correcta: ${pregunta.respuesta_correcta}</p>
18             <p>Justificación: ${pregunta.justificacion}</p>
19         </div>
20     `);
21     document.getElementById("preguntas").innerHTML = preguntas_html;
22 }
23
24 // Ejecutar la función al cargar la página
25 window.onload = mostrarPreguntas;
26
27 // Fin del código
```

Figura 141. Codificación HU025.


```

1 router.post("/registrarCambio", (req, res) => {
2   let tipoUnidad = req.body.tipoUnidad;
3   let idtema = req.body.idtema;
4   let idsubtema = req.body.idsubtema;
5   let idejercicio = req.body.idejercicio;
6   let idpregunta = req.body.idpregunta;
7   let detalles = req.body.detalles;
8   let idusuario = req.body.idusuario;
9
10  const registrarCambio =
11    `INSERT INTO historial (tipo_unidad, id_tema, id_subtema, id_ejercicio, id_pregunta, detalles, id_usuario) VALUES (?, ?, ?, ?, ?, ?, ?)`;
12
13  sql.ejecutarResSQL(
14    registrarCambio,
15    [tipoUnidad, idtema, idsubtema, idejercicio, idpregunta, detalles, idusuario],
16    (resultado) => {
17      if (resultado["affectedRows"] > 0) {
18        return res.status(200).send({
19          en: 1,
20          m: "Se registro el cambio con éxito",
21          locambio: resultado["insertId"]});
22      } else {
23        return res.status(200).send({ en: -1, m: "No se pudo registrar el cambio" });
24      }
25    });
26 });

```

Figura 144. Codificación HU028.

```

1 router.post("/activarDesactivarPregunta", [], (req, res) => {
2   let id = req.body.id;
3   let estado = req.body.estado;
4   const actualizarPregunta = `UPDATE pregunta SET estado = ? WHERE id = ?`;
5   sql.ejecutarResSQL(actualizarPregunta, [estado, id], (resultado) => {
6     if (resultado["affectedRows"] > 0)
7       return res
8         .status(200)
9         .send({ en: 1, m: "Cambio el estado de la pregunta", idPregunta: id });
10    return res
11      .status(200)
12      .send({ en: -1, m: "No se pudo cambiar el estado de la pregunta" });
13  });
14 });
15

```

Figura 145. Codificación HU029.

```

1 router.post("/buscarTemas", (req, res) => {
2   const busqueda = req.body.busqueda;
3   const idUsuario = req.body.idUsuario;
4
5   // Query para buscar temas que coincidan con el término de búsqueda
6   const buscarTemas =
7     `SELECT
8     usuario_tema.idtema,
9     usuario_tema.progreso,
10    tema.titulo,
11    tema.objetivos,
12    tema.descripcion,
13    tema.recursos,
14    tema.estado
15  FROM ${DB}.usuario_tema
16  INNER JOIN ${DB}.tema ON usuario_tema.idTema = tema.id
17  WHERE usuario_tema.idusuario = ?
18  AND (
19    tema.titulo LIKE CONCAT('%', ?, '%')
20    OR tema.descripcion LIKE CONCAT('%', ?, '%')
21    OR tema.objetivos LIKE CONCAT('%', ?, '%')
22  )
23  ORDER BY tema.titulo ASC;
24  `;
25
26  sql.ejecutarResSQL(
27    buscarTemas,
28    [idUsuario, busqueda, busqueda, busqueda],
29    (resultado) => {
30      if (resultado.length > 0) {
31        return res.status(200).send({
32          en: 1,
33          m: "Contenido encontrado",
34          temas: resultado
35        });
36      } else {
37        return res.status(200).send({
38          en: -1,
39          m: "No se encontro contenido que coincida con la búsqueda"
40        });
41      }
42    });
43  });
44 });

```

Figura 146. Codificación HU030.

```
1 router.post("/actualizarPregunta", (req, res) => {
2   idPregunta = req.body.idPregunta;
3   idEjercicio = req.body.idEjercicio;
4   idSubtema = req.body.idSubtema;
5   idTema = req.body.idTema;
6   idUsuario = req.body.idUsuario;
7   // Actualizar estado de la pregunta
8   const actualizarPregunta = "UPDATE pregunta SET estado_completado = 1 WHERE (idUsuario = ? and idPregunta = ?)";
9   const resultado = db.query(actualizarPregunta, [idUsuario, idPregunta]);
10  // Verificar si se actualizó correctamente
11  if (resultado.affectedRows > 0) {
12    // Actualizar progreso de la pregunta
13    const actualizarProgresoPregunta = "UPDATE progreso SET progreso = ? WHERE (idUsuario = ? and idPregunta = ?)";
14    const resultado2 = db.query(actualizarProgresoPregunta, [100, idUsuario, idPregunta]);
15    // Actualizar progreso de los ejercicios
16    const actualizarProgresoEjercicio = "UPDATE progreso SET progreso = ? WHERE (idUsuario = ? and idEjercicio = ?)";
17    const resultado3 = db.query(actualizarProgresoEjercicio, [100, idUsuario, idEjercicio]);
18    // Actualizar progreso de los subtemas
19    const actualizarProgresoSubtema = "UPDATE progreso SET progreso = ? WHERE (idUsuario = ? and idSubtema = ?)";
20    const resultado4 = db.query(actualizarProgresoSubtema, [100, idUsuario, idSubtema]);
21    // Actualizar progreso de los temas
22    const actualizarProgresoTema = "UPDATE progreso SET progreso = ? WHERE (idUsuario = ? and idTema = ?)";
23    const resultado5 = db.query(actualizarProgresoTema, [100, idUsuario, idTema]);
24    // Enviar respuesta al cliente
25    res.status(200).send({ en: 1, m: "Pregunta actualizada correctamente" });
26  } else {
27    res.status(200).send({ en: -1, m: "No se pudo actualizar la pregunta" });
28  }
29 });
```

Figura 147. Codificación HU031.

```
1 router.post("/completarPregunta", (req, res) => {
2   let idPregunta = req.body.idPregunta;
3   let idEjercicio = req.body.idEjercicio;
4   let idSubtema = req.body.idSubtema;
5   let idTema = req.body.idTema;
6   let idUsuario = req.body.idUsuario;
7   console.log("datos del frontend:", "pregunta", idPregunta, "ejercicio", idEjercicio, "subtema", idSubtema, "tema", idTema, "usuario", idUsuario);
8   // Cambiar estado_completado de la pregunta a 1
9   const completarPregunta = "UPDATE ejercicio.pregunta SET estado_completado = 1 WHERE (idUsuario = ? and idPregunta = ?)";
10  const resultado = db.query(completarPregunta, [idUsuario, idPregunta]);
11  console.log("resultado de completarPregunta:", resultado);
12  if (resultado.affectedRows > 0) {
13    // Actualizar progreso de la pregunta
14    const actualizarProgresoPregunta = "UPDATE progreso SET progreso = ? WHERE (idUsuario = ? and idPregunta = ?)";
15    const resultado2 = db.query(actualizarProgresoPregunta, [100, idUsuario, idPregunta]);
16    console.log("resultado de actualizarProgresoPregunta:", resultado2);
17    // Actualizar progreso de los ejercicios
18    const actualizarProgresoEjercicio = "UPDATE progreso SET progreso = ? WHERE (idUsuario = ? and idEjercicio = ?)";
19    const resultado3 = db.query(actualizarProgresoEjercicio, [100, idUsuario, idEjercicio]);
20    console.log("resultado de actualizarProgresoEjercicio:", resultado3);
21    // Actualizar progreso de los subtemas
22    const actualizarProgresoSubtema = "UPDATE progreso SET progreso = ? WHERE (idUsuario = ? and idSubtema = ?)";
23    const resultado4 = db.query(actualizarProgresoSubtema, [100, idUsuario, idSubtema]);
24    console.log("resultado de actualizarProgresoSubtema:", resultado4);
25    // Actualizar progreso de los temas
26    const actualizarProgresoTema = "UPDATE progreso SET progreso = ? WHERE (idUsuario = ? and idTema = ?)";
27    const resultado5 = db.query(actualizarProgresoTema, [100, idUsuario, idTema]);
28    console.log("resultado de actualizarProgresoTema:", resultado5);
29    // Enviar respuesta al cliente
30    res.status(200).send({ en: 1, m: "Mensaje, progresoEjercicio, progresoSubtema, progresoTema, progresoUsuario" });
31  } else {
32    res.status(200).send({ en: -1, m: "error" });
33  }
34 }
35 // Actualizar progreso de los ejercicios
36 const actualizarProgresoEjercicio = "UPDATE progreso SET progreso = ? WHERE (idUsuario = ? and idEjercicio = ?)";
37 const resultado6 = db.query(actualizarProgresoEjercicio, [100, idUsuario, idEjercicio]);
38 console.log("resultado de actualizarProgresoEjercicio:", resultado6);
39 // Actualizar progreso de los subtemas
40 const actualizarProgresoSubtema = "UPDATE progreso SET progreso = ? WHERE (idUsuario = ? and idSubtema = ?)";
41 const resultado7 = db.query(actualizarProgresoSubtema, [100, idUsuario, idSubtema]);
42 console.log("resultado de actualizarProgresoSubtema:", resultado7);
43 // Actualizar progreso de los temas
44 const actualizarProgresoTema = "UPDATE progreso SET progreso = ? WHERE (idUsuario = ? and idTema = ?)";
45 const resultado8 = db.query(actualizarProgresoTema, [100, idUsuario, idTema]);
46 console.log("resultado de actualizarProgresoTema:", resultado8);
47 // Enviar respuesta al cliente
48 res.status(200).send({ en: -1, m: "error" });
49 } else {
50   res.status(200).send({ en: -1, m: "No se pudo completar la pregunta" });
51 }
52 });
53 });
```

Figura 148. Codificación HU032.



4.4 Pruebas

La **Tabla 8** muestra las pruebas unitarias a desarrollar en la cuarta iteración.

Tabla 8. Pruebas unitarias a ejecutar en la cuarta iteración

ID	Descripción	Datos de Entrada	Salida Esperada	Resultado
PU039	Listar preguntas activas	<ul style="list-style-type: none"> ➤ idEjercicio: 27, ➤ idUsuario: 47, ➤ mensaje: "preguntasActivas" 	Preguntas obtenidas	Aprobada
PU040	Listar ejercicios cuando existe ningún ejercicio.	<ul style="list-style-type: none"> ➤ idEjercicio: 999, ➤ idUsuario: 47, ➤ mensaje: "preguntasActivas" 	No se encontraron preguntas	Aprobada
PU041	Listar preguntas activas e inactivas.	<ul style="list-style-type: none"> ➤ idEjercicio: 27, ➤ idUsuario: 47 	Preguntas obtenidas	Aprobada
PU042	Registrar una nueva pregunta	<ul style="list-style-type: none"> ➤ enunciado: 'Nuevo enunciado de prueba', ➤ opcion_a: 'Opción A de prueba', ➤ opcion_b: 'Opción B de prueba', ➤ opcion_c: 'Opción C de prueba', ➤ opcion_d: 'Opción D de prueba', ➤ respuesta_correcta: 'a', ➤ justificación: 'Justificación de la respuesta de prueba', ➤ idEjercicio: 27 	Se registró la pregunta con éxito y se enviaron correos de notificación	Aprobada
PU043	Editar una pregunta existente	<ul style="list-style-type: none"> ➤ id: 32, ➤ enunciado: 'Enunciado editado', ➤ opcion_a: 'Opción A editada', ➤ opcion_b: 'Opción B editada', ➤ opcion_c: 'Opción C editada', ➤ opcion_d: 'Opción D editada', ➤ respuesta_correcta: 'b', ➤ justificación: 'Justificación editada', ➤ estado: 1 	Se editó la pregunta con éxito	Aprobada
PU044	Editar una pregunta inexistente.	<ul style="list-style-type: none"> ➤ id: 999999, ➤ enunciado: 'Enunciado editado', ➤ opcion_a: 'Opción A editada', ➤ opcion_b: 'Opción B editada', ➤ opcion_c: 'Opción C editada', ➤ opcion_d: 'Opción D editada', ➤ respuesta_correcta: 'b', ➤ justificación: 'Justificación editada', ➤ estado: 1 	No se pudo editar la pregunta	Aprobada
PU045	Registrar un nuevo cambio en pregunta	<ul style="list-style-type: none"> ➤ tipoEntidad: "pregunta", ➤ idTema: 32, ➤ idSubtema: 37, 	Se registró el cambio con éxito	Aprobada

		<ul style="list-style-type: none"> ➤ idEjercicio: 27, ➤ idPregunta: 32, ➤ detalles: "Se actualizó el contenido de la pregunta", ➤ idUsuario: 47 		
PU046	Activar/Desactivar pregunta	<ul style="list-style-type: none"> ➤ id: 32, ➤ estado: 1 	Activar o desactivar pregunta	Aprobada
PU047	Buscar contenido con término existente	<ul style="list-style-type: none"> ➤ idUsuario: 27, ➤ busqueda: "Titulo editado del ejercicio". 	Contenido encontrado	Aprobada
PU048	Visualizar preguntas	➤ N/A	Preguntas activas listadas	Aprobada
PU049	Responder pregunta de control	<ul style="list-style-type: none"> ➤ idPregunta: 32, ➤ idEjercicio: 27, ➤ idSubtema: 37, ➤ idTema: 32, ➤ idUsuario: 47 	Pregunta contestada con éxito.	Aprobada

A continuación, se presentan las seis pruebas unitarias desarrolladas en la cuarta iteración.

```

1 describe('Test de listado de preguntas', () => {
2   it('listar preguntas activas del ejercicio', async () => {
3     const response = await api
4       .post(`${rutaPreguntas} + '/listarPreguntas'`)
5       .send({body: listaActivas});
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.cn).toBe(1);
9     expect(response.body.m).toBe("Preguntas obtenidas");
10    expect(Array.isArray(response.body.preguntas)).toBe(true);
11  });
12
13  it('Intento de listar preguntas cuando no hay ninguna', async () => {
14    const bodySinPreguntas = {
15      idEjercicio: 999, // ID de ejercicio que no tiene preguntas
16      idUsuario: 47,
17      mensaje: "preguntasActivas"
18    };
19
20    const response = await api
21      .post(`${rutaPreguntas} + '/listarPreguntas'`)
22      .send({body: listaSinPreguntas});
23
24    expect(response.statusCode).toBe(200);
25    expect(response.body.cn).toBe(1);
26    expect(response.body.m).toBe("No se encontraron preguntas");
27  });
28
29  it('listar todas las preguntas del ejercicio (activas e inactivas)', async () => {
30    const bodyTodasPreguntas = {
31      idEjercicio: 27,
32      idUsuario: 47
33    };
34
35    const response = await api
36      .post(`${rutaPreguntas} + '/listarPreguntas'`)
37      .send({body: todasPreguntas});
38
39    expect(response.statusCode).toBe(200);
40    expect(response.body.cn).toBe(1);
41    expect(response.body.m).toBe("Preguntas obtenidas");
42    expect(Array.isArray(response.body.preguntas)).toBe(true);
43  });
44 });

```

Figura 149. Pruebas unitarias (PU039, PU040, PU041), listar preguntas.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCuatro
> camigo@1.0.0 testIteracionCuatro
> cross-env BD=camigo_prueba jest --verbose --silent tests/pregunta/pregunta.test.js

PASS tests/pregunta/pregunta.test.js
  Test de listado de preguntas
    ✓ Listar preguntas activas del ejercicio (80 ms)
    ✓ Intento de listar preguntas cuando no hay ninguna (8 ms)
    ✓ Listar todas las preguntas del ejercicio (activas e inactivas) (7 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.206 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 150. Ejecución pruebas unitarias (PU039, PU040, PU041), listar preguntas.

```
1 describe('Test de registro de preguntas', () => {
2   it('Registrar una nueva pregunta con notificaciones', async () => {
3     const response = await api
4       .post(rutaPreguntas + '/registrarPregunta')
5       .send(bodyRegistrar);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Se registró la pregunta con éxito y se enviaron correos de notificación");
10    expect(response.body.idPregunta).toBeDefined();
11    expect(typeof response.body.idPregunta).toBe('number');
12  });
13 });
```

Figura 151. Pruebas unitarias (PU042), registrar preguntas.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCuatro
> camigo@1.0.0 testIteracionCuatro
> cross-env BD=camigo_prueba jest --verbose --silent tests/pregunta/pregunta.test.js

PASS tests/pregunta/pregunta.test.js
  Test de registro de preguntas
    ✓ Registrar una nueva pregunta con notificaciones (162 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.207 s, estimated 7 s
Jest did not exit one second after the test run has completed.
```

Figura 152. Ejecución pruebas unitarias (PU042), registrar preguntas.

```
1 describe('Test de edición de preguntas', () => {
2   it('Editar una pregunta existente', async () => {
3     const response = await api
4       .post(rutaPreguntas + '/editarPregunta')
5       .send(bodyEditar);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Se editó la pregunta con éxito");
10    expect(response.body.idPregunta).toBe(bodyEditar.id);
11    expect(response.body.preguntaEditadaBackend).toEqual(bodyEditar);
12  });
13
14  it('Intentar editar una pregunta inexistente', async () => {
15    const bodyPreguntaInexistente = {
16      ...bodyEditar,
17      id: 999 // ID que no existe en la base de datos
18    };
19
20    const response = await api
21      .post(rutaPreguntas + '/editarPregunta')
22      .send(bodyPreguntaInexistente);
23
24    expect(response.statusCode).toBe(200);
25    expect(response.body.en).toBe(-1);
26    expect(response.body.m).toBe("No se pudo editar la pregunta");
27  });
28 });
29
```

Figura 153. Pruebas unitarias (PU043, PU044), editar preguntas.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCuatro
> camigo@1.0.0 testIteracionCuatro
> cross-env BD=camigo_prueba jest --verbose --silent tests/pregunta/pregunta.test.js

PASS tests/pregunta/pregunta.test.js
  Test de edición de preguntas
    ✓ Editar una pregunta existente (106 ms)
    ✓ Intentar editar una pregunta inexistente (9 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.23 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 154. Ejecución pruebas unitarias (PU043, PU044), editar preguntas.

```
1 describe('Test de registro de cambios', () => {
2   // Prueba para registrar un cambio exitosamente
3   it('Registrar un nuevo cambio en tema', async () => {
4     const response = await api
5       .post(rutaHistorial + '/registrarCambio')
6       .send(bodyRegistrarCambio);
7
8     expect(response.statusCode).toBe(200);
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Se registró el cambio con éxito");
11    expect(response.body.idCambio).toBeDefined();
12    expect(typeof response.body.idCambio).toBe('number');
13  });
14 });
```

Figura 155. Pruebas unitarias (PU045), historial de cambios en preguntas.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCuatro
> camigo@1.0.0 testIteracionCuatro
> cross-env BD=camigo_prueba jest --verbose --silent tests/pregunta/pregunta.test.js

PASS tests/pregunta/pregunta.test.js
  Test de registro de cambios en pregunta
    ✓ Registrar un nuevo cambio en pregunta (106 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.454 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 156. Ejecución pruebas unitarias (PU045), historial de cambios en preguntas.

```
1 describe('Pruebas para activar/desactivar pregunta', () => {
2   it('debería activar una pregunta correctamente', async () => {
3     const response = await api
4       .post(rutaPreguntas + '/activarDesactivarPregunta')
5       .send(bodyActivar_Desactivar)
6       .expect(200)
7       .expect('Content-Type', /application\/json/);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Cambio el estado de la pregunta");
10    expect(response.body.idPregunta).toBe(bodyActivar_Desactivar.id);
11  });
12
13   it('debería desactivar una pregunta correctamente', async () => {
14     bodyActivar_Desactivar.estado = 1;
15
16     const response = await api
17       .post(rutaPreguntas + '/activarDesactivarPregunta')
18       .send(bodyActivar_Desactivar)
19       .expect(200)
20       .expect('Content-Type', /application\/json/);
21     expect(response.body.en).toBe(1);
22     expect(response.body.m).toBe("Cambio el estado de la pregunta");
23     expect(response.body.idPregunta).toBe(bodyActivar_Desactivar.id);
24  });
25
26   it('debería manejar el intento de activar/desactivar una pregunta inexistente', async () => {
27     const bodyPreguntaInexistente = {
28       ..bodyActivar_Desactivar,
29       id: 999 // ID que no existe
30     };
31
32     const response = await api
33       .post(rutaPreguntas + '/activarDesactivarPregunta')
34       .send(bodyPreguntaInexistente)
35       .expect(200)
36       .expect('Content-Type', /application\/json/);
37     expect(response.body.en).toBe(-1);
38     expect(response.body.m).toBe("No se pudo cambiar el estado de la pregunta");
39  });
40
41  });
42
43 });
```

Figura 157. Pruebas unitarias (PU046), activar/desactivar preguntas.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCuatro
> camigo@1.0.0 testIteracionCuatro
> cross-env BD=camigo_prueba jest --verbose --silent tests/pregunta/pregunta.test.js

PASS tests/pregunta/pregunta.test.js
  Pruebas para activar/desactivar pregunta
    ✓ debería activar una pregunta correctamente (91 ms)
    ✓ debería desactivar una pregunta correctamente (16 ms)
    ✓ debería manejar el intento de activar/desactivar una pregunta inexistente (4 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.296 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 158. Ejecución pruebas unitarias (PU046), activar/desactivar preguntas.

```
1 router.post("/buscarTemas", (req, res) => {
2   const busqueda = req.body.busqueda;
3   const idUsuario = req.body.idUsuario;
4
5   // Query para buscar temas que coincidan con el término de búsqueda
6   const buscarTemas = `
7   SELECT
8     usuario_tema.idTema,
9     usuario_tema.progreso,
10    tema.titulo,
11    tema.objetivos,
12    tema.descripcion,
13    tema.recursos,
14    tema.estado
15  FROM ${BD}.usuario_tema
16  INNER JOIN ${BD}.tema ON usuario_tema.idTema = tema.id
17  WHERE usuario_tema.idUsuario = ?
18  AND (
19    tema.titulo LIKE CONCAT('%', ?, '%')
20    OR tema.descripcion LIKE CONCAT('%', ?, '%')
21    OR tema.objetivos LIKE CONCAT('%', ?, '%')
22  )
23  ORDER BY tema.titulo ASC;
24  `;
25
26  sql.ejecutarResSQL(
27    buscarTemas,
28    [idUsuario, busqueda, busqueda],
29    (resultado) => {
30      if (resultado.length > 0) {
31        return res.status(200).send({
32          en: 1,
33          m: "Contenido encontrado",
34          temas: resultado
35        });
36      } else {
37        return res.status(200).send({
38          en: -1,
39          m: "No se encontro contenido que coincida con la busqueda"
40        });
41      }
42    }
43  );
44 });
```

Figura 159. Pruebas unitarias (PU047), buscar preguntas.



```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Test de búsqueda de temas
    / Buscar contenido con término existente (86 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 1.351 s, estimated 3 s
Jest did not exit one second after the test run has completed.
```

Figura 160. Ejecución pruebas unitarias (PU047), buscar preguntas.

```
1 describe('Test de listado de preguntas', () => {
2   it('listar preguntas activas del ejercicio', async () => {
3     const response = await api
4       .post(rutaPreguntas + '/listarPreguntas')
5       .send(bodyListarActivas);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Preguntas obtenidas");
10    expect(Array.isArray(response.body.preguntas)).toBe(true);
11  });
12
13  it('Intento de listar preguntas cuando no hay ninguna', async () => {
14    const bodySinPreguntas = {
15      idEjercicio: 999, // ID de ejercicio que no tiene preguntas
16      idUsuario: 47,
17      mensaje: "preguntasActivas"
18    };
19
20    const response = await api
21      .post(rutaPreguntas + '/listarPreguntas')
22      .send(bodySinPreguntas);
23
24    expect(response.statusCode).toBe(200);
25    expect(response.body.en).toBe(-1);
26    expect(response.body.m).toBe("No se encontraron preguntas");
27  });
28
29  it('listar todas las preguntas del ejercicio (activas e inactivas)', async () => {
30    const bodyTodasPreguntas = {
31      idEjercicio: 27,
32      idUsuario: 47
33    };
34
35    const response = await api
36      .post(rutaPreguntas + '/listarPreguntas')
37      .send(bodyTodasPreguntas);
38
39    expect(response.statusCode).toBe(200);
40    expect(response.body.en).toBe(1);
41    expect(response.body.m).toBe("Preguntas obtenidas");
42    expect(Array.isArray(response.body.preguntas)).toBe(true);
43  });
44 });
```

Figura 161. Pruebas unitarias (PU048), visualizar subtemas.



```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCuatro
> camigo@1.0.0 testIteracionCuatro
> cross-env BD=camigo_prueba jest --verbose --silent tests/pregunta/pregunta.test.js

PASS tests/pregunta/pregunta.test.js
  Test de listado de preguntas
    ✓ Listar preguntas activas del ejercicio (80 ms)
    ✓ Intento de listar preguntas cuando no hay ninguna (8 ms)
    ✓ Listar todas las preguntas del ejercicio (activas e inactivas) (7 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.206 s, estimated 2 s
Test did not exit one second after the test run has completed.
```

Figura 162. Ejecución pruebas unitarias (PU048), visualizar subtemas.

```
1 describe('Test de completar pregunta', () => {
2   it('Completar una pregunta existente', async () => {
3     const response = await api
4       .post(rutaPreguntas + '/completarPregunta')
5       .send(bodyCompletarPregunta);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.progresoEjercicio).toBeDefined();
10    expect(response.body.progresoSubtema).toBeDefined();
11    expect(response.body.progresoTema).toBeDefined();
12    expect(response.body.progresoUsuario).toBeDefined();
13  });
14
15  it('Intentar completar una pregunta inexistente', async () => {
16    const bodyPreguntaInexistente = {
17      ...bodyCompletarPregunta,
18      idPregunta: 999 // ID que no existe
19    };
20
21    const response = await api
22      .post(rutaPreguntas + '/completarPregunta')
23      .send(bodyPreguntaInexistente);
24
25    expect(response.statusCode).toBe(200);
26    expect(response.body.en).toBe(-1);
27    expect(response.body.m).toBe("No se pudo completar la pregunta");
28  });
29 });
```



Figura 163. Pruebas unitarias (PU049), responder pregunta control.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCuatro
> camigo@1.0.0 testIteracionCuatro
> cross-env BD=camigo_prueba jest --verbose --silent tests/pregunta/pregunta.test.js

PASS tests/pregunta/pregunta.test.js
  Test de completar pregunta
    ✓ Completar una pregunta existente (209 ms)
    ✓ Intentar completar una pregunta inexistente (8 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.693 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 164. Ejecución pruebas unitarias (PU049), responder pregunta de control.

5 Quinta Iteración

5.1 Planificación

La **Tabla 7** muestra las actividades específicas a desarrollar en la quinta iteración.

Tabla 9. Historias de Usuario a desarrollar en la quinta iteración.

HU #	Historia de usuario	Fecha de creación	Fecha de entrega
HU033	Listar comentarios	15/12/2023	06/02/2024
HU034	Registrar pregunta de control	20/12/2023	06/02/2024
HU035	Editar pregunta de control	23/12/2024	06/02/2024
HU036	Historial de cambios por pregunta de control	26/12/2024	06/02/2024
HU037	Activar/Desactivar pregunta de control	27/12/2024	06/02/2024
HU038	Buscar pregunta de control	28/01/2024	06/02/2024
HU039	Visualizar pregunta de control	30/01/2024	06/02/2024
HU040	Responder pregunta de control	01/02/2024	06/02/2024

5.2 Diseño

La **Figura 47** muestra las funcionalidades de manera general a desarrollar en la cuarta iteración.

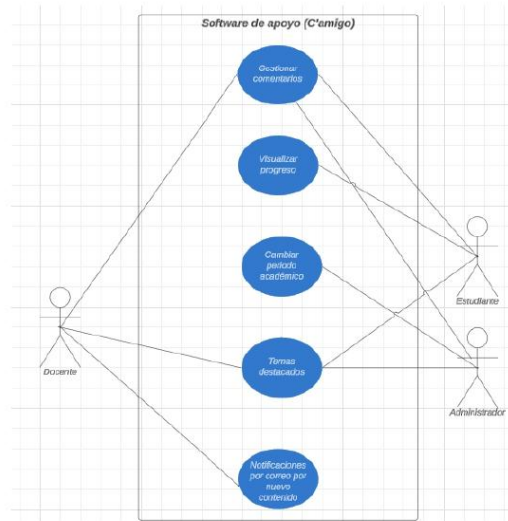


Figura 165. Casos de Uso general para la cuarta iteración.

5.2.1 Diagramas BPMN

A continuación, se presentan los diagramas BPMN pertenecientes a cada Historia de Usuario a desarrollar en la cuarta iteración.

5.2.1.1 HU033: Listar comentarios

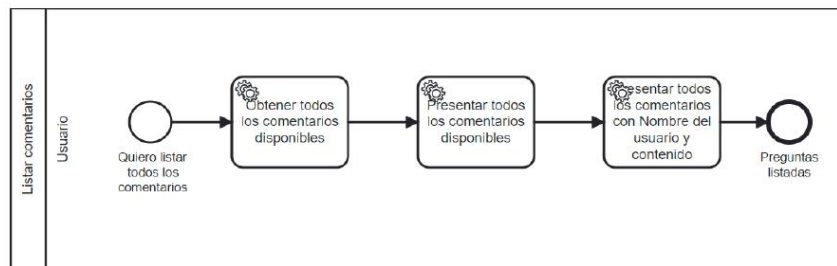


Figura 166. Diagrama BPMN de HU033.

5.2.1.2 HU034: Registrar comentarios

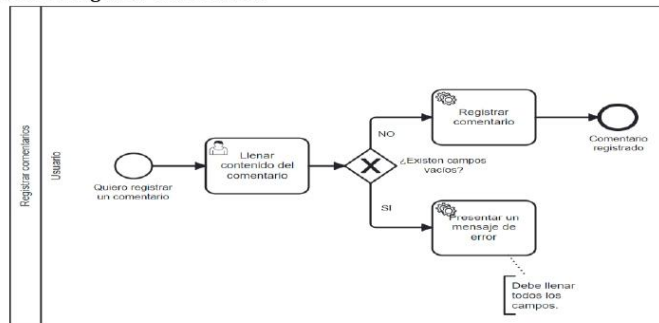


Figura 167. Diagrama BPMN de HU034.

5.2.1.3 HU035: Editar comentarios

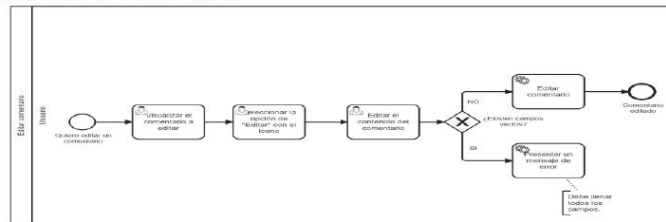


Figura 168. Diagrama BPMN de HU035.

5.2.1.4 HU036: Visualizar comentarios

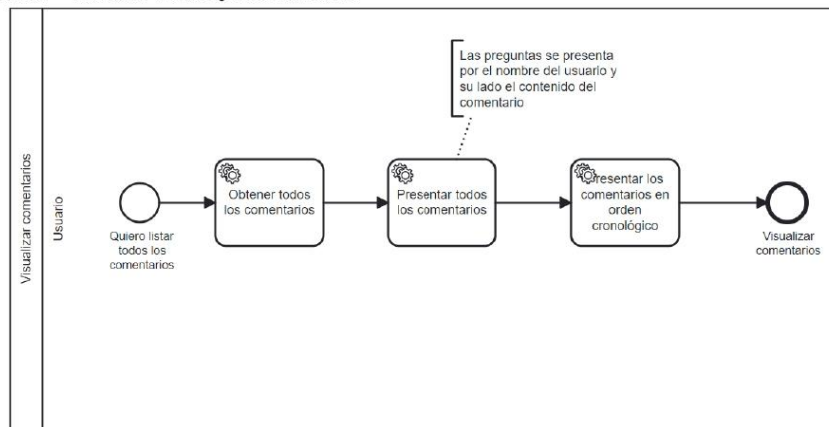


Figura 169. Diagrama BPMN de HU036.

5.2.1.5 HU037: Visualizar progreso

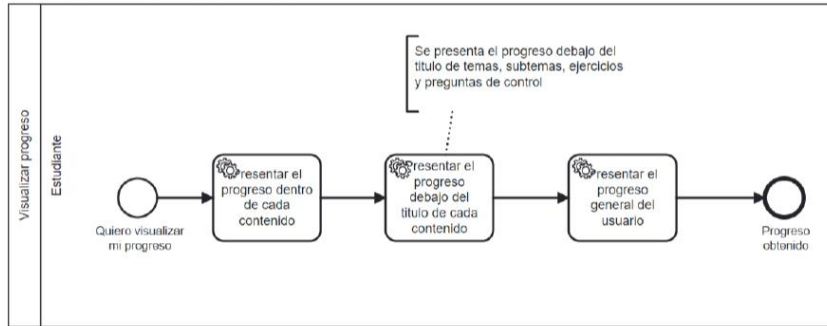


Figura 170. Diagrama BPMN de HU037.

5.2.1.6 HU038: Cambiar periodo académico.

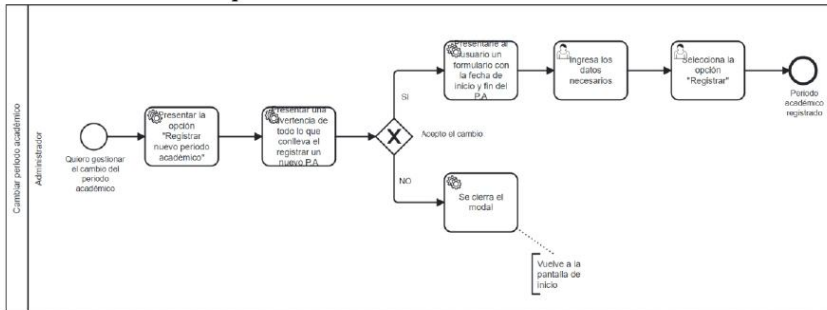


Figura 171. Diagrama BPMN de HU038.

5.2.1.7 HU039: Temas destacados

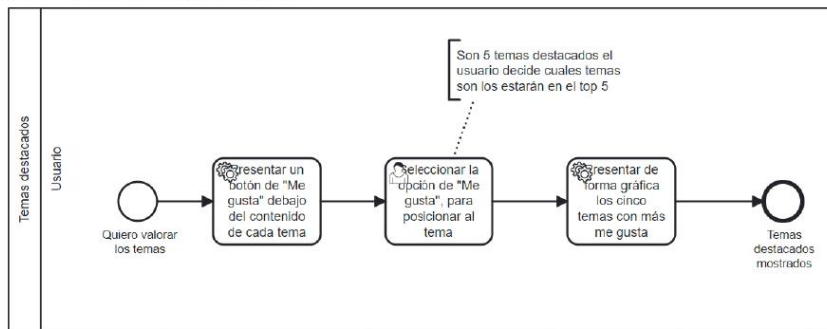


Figura 172. Diagrama BPMN de HU039.

5.2.1.8 HU040: Notificaciones por email de nuevo contenido

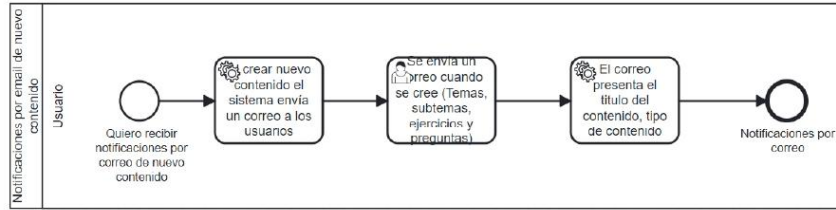


Figura 173. Diagrama BPMN de HU040.

5.2.2 Prototipos

A continuación, se presentan los prototipos pertenecientes a cada Historia de Usuario a desarrollar en la cuarta iteración.


5.2.2.1 HU033: Listar comentarios

¿Tienes dudas ?, coméntalas:

Esto es un nuevo comentario

Realizar comentario

Comentarios

Usuario en sesión: Tengo una duda, puedo ingresar otro tipo de dato y el funcionamiento del código seguirá igual 

Otro usuario: Excelente ejercicio

Figura 174. Prototipo de HU033.

5.2.2.2 HU034: Registrar comentarios

¿Tienes dudas ?, coméntalas:

Esto es un nuevo comentario

Realizar comentario

Figura 175. Prototipo de HU034.




5.2.2.3 HU035: Editar comentarios

¿Tienes dudas ?, coméntalas:

Tengo una duda, puedo ingresar otro tipo de dato y el funcionamiento del código seguirá igual

Comentarios

Usuario en sesión: Tengo una duda, puedo ingresar otro tipo de dato y el funcionamiento del código seguirá igual 

Otro usuario: Excelente ejercicio





Figura 176. Prototipo de HU035.

5.2.2.4 HU036: Visualizar comentarios

Comentarios

Usuario en sesión: Tengo una duda, puedo ingresar otro tipo de dato y el funcionamiento del código seguirá igual 

Otro usuario: Excelente ejercicio

Figura 177. Prototipo de HU036.

5.2.2.5 HU037: Visualizar progreso

ESTUDIANTE, DOCENTE, ADMINISTRADOR

Código

Estructura Algorítmicas Condicionales

Objetivos de aprendizaje:

- Aprender estructuras algorítmicas condicionales
- Entender el correcto funcionamiento de esta estructura de control

Descripción:

Las estructuras algorítmicas condicionales son construcciones fundamentales en la programación y la resolución de problemas algorítmicos. Estas estructuras permiten que un programa tome decisiones basadas en ciertas condiciones y ejecute diferentes opciones en función de si esas condiciones se cumplen o no.

La estructura algorítmica condicional simple es una construcción básica en programación que permite ejecutar un bloque de código si una condición dada es verdadera (True).

Recursos adicionales:

- Video tutorial
- Libro relacionado al tema

Su progreso en Código es de: 50%

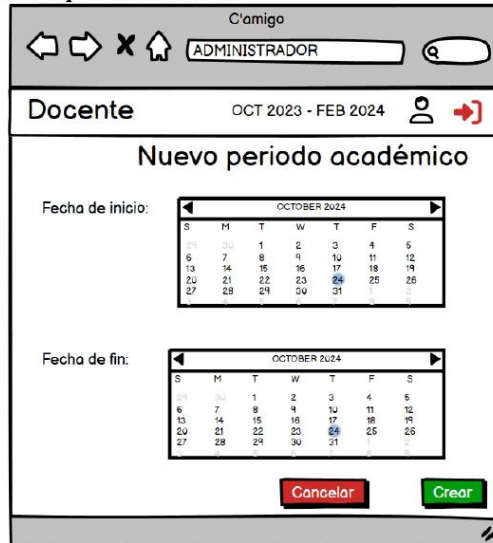
Temas

Buscar tema

Lógica de programación	<div style="width: 100%;"></div>
Conceptos fundamentales de algoritmos	<div style="width: 100%;"></div>
Estructuras algorítmicas de control	<div style="width: 100%;"></div>
Programación modular y Estructuras de datos sencillas	<div style="width: 100%;"></div>
Conceptos de programación tiempo	<div style="width: 100%;"></div>

Figura 178. Prototipo de HU037.

5.2.2.6 HU038: Cambiar periodo académico



The interface shows a user role of 'ADMINISTRADOR' and a selected period of 'OCT 2023 - FEB 2024'. The main section is titled 'Nuevo periodo académico' and contains two calendar pickers for 'OCTOBER 2024'. The 'Fecha de inicio' calendar has the 23rd selected, and the 'Fecha de fin' calendar has the 23rd selected. At the bottom, there are 'Cancelar' and 'Crear' buttons.

Figura 179. Prototipo de HU038.

5.2.2.7 HU039: Temas destacados



The interface shows a user role of 'ESTUDIANTE, DOCENTE, ADMINISTRADO' and a selected period of 'OCT 2023 - FEB 2024'. The main section is titled 'Temas destacados' and contains a text description: 'Aquí se presentan los cinco temas mejor recibidos por los usuarios del software de apoyo'. To the right of the text is a bar chart with five bars of varying heights, representing the top five topics.

Figura 180. Prototipo de HU039.

5.2.2.8 HU040: Notificaciones de email de nuevo contenido



Figura 181. Prototipo de HU040.

5.3 Codificación

A continuación, se presenta los fragmentos principales del código fuente desarrollados en la quinta iteración.



Figura 182. Codificación HU033.



Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

```

1  router.post('/registroPregunta', [], (req, res) => {
2      let enunciado = req.body.enunciado;
3      let opcion_a = req.body.opcion_a;
4      let opcion_b = req.body.opcion_b;
5      let opcion_c = req.body.opcion_c;
6      let opcion_d = req.body.opcion_d;
7      let respuesta_correcta = req.body.respuesta_correcta;
8      let justificacion = req.body.justificacion;
9      let estado = 1;
10     let enunciado = req.body.enunciado;
11
12     const registrarPregunta =
13     `INSERT INTO pregunta (enunciado, opcion_a, opcion_b, opcion_c, opcion_d, respuesta_correcta, justificacion, estado, idjercicio) VALUES (?, ?, ?, ?, ?, ?, ?, ?)`;
14
15     sql.executeAsync(
16         registrarPregunta,
17         [enunciado, opcion_a, opcion_b, opcion_c, opcion_d, respuesta_correcta, justificacion, estado, idjercicio],
18         (resultado) => {
19             if (resultado['affectedRows'] > 0) {
20                 console.log('Resultado = ' + resultado['insertId']);
21                 // Se puede consultar para verificar en el ejercicio pregunta por cada usuario
22                 const buscarPorEjercicioIDPregunta = `SELECT id_usuario, idPregunta FROM pregunta WHERE id_usuario = ?`;
23                 sql.executeAsync(
24                     buscarPorEjercicioIDPregunta,
25                     [idPregunta],
26                     (resultado) => {
27                         if (resultado.length > 0) {
28                             console.log('Resultado = ' + resultado);
29                             const actualizarCuentas = `UPDATE cuentas SET correo = ?`;
30                             sql.executeAsync(
31                                 actualizarCuentas,
32                                 [resultadoCorreos],
33                                 () => {
34                                     if (resultadoCorreos.length > 0) {
35                                         const mensaje = "Debe crear una nueva cuenta en Cuentas";
36                                         console.log(mensaje);
37                                         console.log('mensaje: ' + mensaje);
38                                         // Acceso a la plataforma para poner a prueba sus conocimientos
39                                         //<!--</>
40                                         //<!--</>
41                                         const COMENTAR ESTE CODIGO </>
42                                         const http = require('https');
43                                         http.get('https://forocitas.unl.edu.ec/index.php.unl.php?tit=Logo%20Universidad%20Nacional%20de%20Loja&style=distribuido; height:auto;',
44                                             () => {});
45                                     }
46                                     resultadoCorreos.forEach(cuenta => {
47                                         sql.executeAsync(
48                                             actualizarCuentas,
49                                             [cuenta.correo],
50                                             () => {});
51                                     });
52                                     return res.status(200).send({
53                                         msg: 'Se registro la pregunta con éxito y se actualizaron correos de notificación',
54                                         idPregunta: idPreguntaInsertada,
55                                     });
56                                 } else {
57                                     return res.status(200).send({
58                                         msg: 'Se registro la pregunta con éxito, pero no hay cuentas para notificar',
59                                         idPregunta: idPreguntaInsertada,
60                                     });
61                                 }
62                             } else {
63                                 return res.status(200).send({ msg: '-1', msg: 'No se pudo registrar en el ejercicio pregunta porque faltan preguntas a ejecutar' });
64                             }
65                         }
66                     }
67                 } else {
68                     return res.status(200).send({ msg: '-1', msg: 'No se pudo registrar la pregunta' });
69                 }
70             }
71         });
72     }
73 });

```

Figura 183. Codificación HU034.

```

1  router.post('/editarPregunta', [], (req, res) => {
2      let id = req.body.id;
3      let enunciado = req.body.enunciado;
4      let opcion_a = req.body.opcion_a;
5      let opcion_b = req.body.opcion_b;
6      let opcion_c = req.body.opcion_c;
7      let opcion_d = req.body.opcion_d;
8      let respuesta_correcta = req.body.respuesta_correcta;
9      let justificacion = req.body.justificacion;
10     let estado = req.body.estado;
11     let preguntaEditarBackend = { id, enunciado, opcion_a, opcion_b, opcion_c, opcion_d, respuesta_correcta, justificacion, estado };
12     const editarPregunta =
13     `UPDATE pregunta SET enunciado = ?, opcion_a = ?, opcion_b = ?, opcion_c = ?, opcion_d = ?, respuesta_correcta = ?, justificacion = ?, estado = ? WHERE id = ?`;
14
15     sql.executeAsync(
16         editarPregunta,
17         [enunciado, opcion_a, opcion_b, opcion_c, opcion_d, respuesta_correcta, justificacion, estado, id],
18         (resultado) => {
19             if (resultado['affectedRows'] > 0)
20                 return res
21                 .status(200)
22                 .send({ msg: 'Se editó la pregunta con éxito', idPregunta: id, preguntaEditarBackend });
23             return res.status(200).send({ msg: '-1', msg: 'No se pudo editar la pregunta' });
24         });
25     }
26 });

```

Figura 184. Codificación HU035.

```

1 router.post('/registrarCambio', (req, res) => {
2   let tipoCambio = req.body.tipoCambio;
3   let tipoCambio = req.body.tipoCambio;
4   let idSubtema = req.body.idSubtema;
5   let idPregunta = req.body.idPregunta;
6   let idUsuario = req.body.idUsuario;
7   let detalles = req.body.detalles;
8
9   const registrarCambio =
10    `INSERT INTO cambios (tipo_cambio, id_tema, id_subtema, id_pregunta, id_usuario) VALUES (?, ?, ?, ?, ?)`;
11
12   sql.ejecutarResSQL(
13     registrarCambio,
14     [tipoCambio, idSubtema, idPregunta, idUsuario],
15     (resultado) => {
16       if (resultado["affectedRows"] > 0) {
17         return res.status(200).send({
18           en: 1,
19           m: "Se registro el cambio con éxito",
20           locacion: resultado["insertId"],
21         });
22       } else {
23         return res.status(200).send({ en: -1, m: "No se pudo registrar el cambio" });
24       }
25     });
26 }
27 });

```

Figura 185. Codificación HU036.

```

1 router.post("/activarDesactivarPregunta", [], (req, res) => {
2   let id = req.body.id;
3   let estado = req.body.estado;
4   const actualizarPregunta = "UPDATE pregunta SET estado = ? WHERE id = ?";
5   sql.ejecutarResSQL(actualizarPregunta, [estado, id], (resultado) => {
6     if (resultado["affectedRows"] > 0)
7       return res
8         .status(200)
9         .send({ en: 1, m: "Cambio el estado de la pregunta", idPregunta: id });
10    return res
11      .status(200)
12      .send({ en: -1, m: "No se pudo cambiar el estado de la pregunta" });
13  });
14 });
15

```

Figura 186. Codificación HU037.

```

1 router.post("/buscartemas", (req, res) => {
2   const busqueda = req.body.busqueda;
3   const idUsuario = req.body.idUsuario;
4
5   // Query para buscar temas que coincidan con el término de búsqueda
6   const buscarTemas = `
7     SELECT
8       usuario_tema.idTema,
9       usuario_tema.progreso,
10      tema.titulo,
11      tema.objetivos,
12      tema.descripcion,
13      tema.recursos,
14      tema.estado
15    FROM ${DB} usuario_tema
16    INNER JOIN ${DB} tema ON usuario_tema.idTema = tema.id
17    WHERE usuario_tema.idUsuario = ?
18    AND (
19      tema.titulo LIKE CONCAT('%', ?, '%')
20      OR tema.descripcion LIKE CONCAT('%', ?, '%')
21      OR tema.objetivos LIKE CONCAT('%', ?, '%')
22    )
23    ORDER BY tema.titulo ASC;
24  `;
25
26   sql.ejecutarResSQL(
27     buscarTemas,
28     [idUsuario, busqueda, busqueda, busqueda],
29     (resultado) => {
30       if (resultado.length > 0) {
31         return res.status(200).send({
32           en: 1,
33           m: "Contenido encontrado",
34           temas: resultado
35         });
36       } else {
37         return res.status(200).send({
38           en: -1,
39           m: "No se encontro contenido que coincida con la búsqueda"
40         });
41       }
42     });
43 }
44 });

```

Figura 187. Codificación HU038.

```

1 router.post("/actualizarPregunta", (req, res) => {
2   let idPregunta = req.body.idPregunta;
3   let idUsuario = req.body.idUsuario;
4   let idEjercicio = req.body.idEjercicio;
5   let idSubtema = req.body.idSubtema;
6   let idTema = req.body.idTema;
7   let idSubtema = req.body.idSubtema;
8   let idUsuario = req.body.idUsuario;
9
10  // Consultar pregunta
11  const query = `SELECT * FROM preguntas WHERE idPregunta = ?`;
12  const sql = `UPDATE preguntas SET estado = ? WHERE idPregunta = ?`;
13  const sql2 = `UPDATE progreso SET progreso = ? WHERE idEjercicio = ? AND idUsuario = ?`;
14  const sql3 = `UPDATE progreso SET progreso = ? WHERE idSubtema = ? AND idUsuario = ?`;
15  const sql4 = `UPDATE progreso SET progreso = ? WHERE idTema = ? AND idUsuario = ?`;
16  const sql5 = `UPDATE progreso SET progreso = ? WHERE idSubtema = ? AND idUsuario = ?`;
17
18  const db = require('mysql');
19  const connection = db.connect({
20    host: 'localhost',
21    user: 'root',
22    password: 'root',
23    database: 'examen'
24  });
25
26  connection.query(sql, [idPregunta], (err, results) => {
27    if (err) {
28      console.log('Error al consultar la pregunta');
29      return res.status(500).send({ m: 'Error al consultar la pregunta' });
30    }
31
32    if (results.length > 0) {
33      const pregunta = results[0];
34      const estado = req.body.estado;
35      connection.query(sql, [idPregunta, estado], (err, results) => {
36        if (err) {
37          console.log('Error al actualizar la pregunta');
38          return res.status(500).send({ m: 'Error al actualizar la pregunta' });
39        }
40
41        // Actualizar progreso
42        connection.query(sql2, [idEjercicio, idUsuario, pregunta.progreso], (err, results) => {
43          if (err) {
44            console.log('Error al actualizar el progreso del ejercicio');
45            return res.status(500).send({ m: 'Error al actualizar el progreso del ejercicio' });
46          }
47
48          connection.query(sql3, [idSubtema, idUsuario, pregunta.progreso], (err, results) => {
49            if (err) {
50              console.log('Error al actualizar el progreso del subtema');
51              return res.status(500).send({ m: 'Error al actualizar el progreso del subtema' });
52            }
53
54            connection.query(sql4, [idTema, idUsuario, pregunta.progreso], (err, results) => {
55              if (err) {
56                console.log('Error al actualizar el progreso del tema');
57                return res.status(500).send({ m: 'Error al actualizar el progreso del tema' });
58              }
59
60              connection.query(sql5, [idSubtema, idUsuario, pregunta.progreso], (err, results) => {
61                if (err) {
62                  console.log('Error al actualizar el progreso del subtema');
63                  return res.status(500).send({ m: 'Error al actualizar el progreso del subtema' });
64                }
65              });
66            });
67          });
68        });
69      });
70    } else {
71      console.log('No se encontró la pregunta');
72      return res.status(404).send({ m: 'No se encontró la pregunta' });
73    }
74  });
75
76  connection.end();
77
78  });
79
80  });

```

Figura 188. Codificación HU039.

```

1 router.post("/completarPregunta", (req, res) => {
2   let idPregunta = req.body.idPregunta;
3   let idEjercicio = req.body.idEjercicio;
4   let idSubtema = req.body.idSubtema;
5   let idTema = req.body.idTema;
6   let idSubtema = req.body.idSubtema;
7   let idUsuario = req.body.idUsuario;
8
9   console.log("datos del frontend:", "pregunta", idPregunta, "ejercicio", idEjercicio, "subtema", idSubtema, "tema", idTema, "usuario", idUsuario);
10
11   // Cambiar estado completado de la pregunta a 1
12   const completarPregunta = `UPDATE ejercicio_pregunta SET estado_completado = 1 WHERE (idUsuario = ? and idPregunta = ?)`;
13   sql.execute(completarPregunta, [idUsuario, idPregunta], (resultado) => {
14     console.log("completarPregunta", resultado["affectedRows"]);
15     if (resultado["affectedRows"] > 0) {
16       calcularActualizarProgresoEjercicio(idEjercicio, idUsuario)
17       .then((mensaje, progresoEjercicio) => {
18         console.log(mensaje);
19         console.log(progresoEjercicio);
20         calcularActualizarProgresoSubtema(idSubtema, idUsuario)
21         .then((mensaje, progresoSubtema) => {
22           console.log(mensaje);
23           console.log(progresoSubtema);
24           calcularActualizarProgresoTema(idTema, idUsuario)
25           .then((mensaje, progresoTema) => {
26             console.log(mensaje);
27             console.log(progresoTema);
28             calcularActualizarProgresoGeneral(idUsuario)
29             .then((mensaje, progresoUsuario) => {
30               console.log(mensaje);
31               console.log(progresoUsuario);
32               return res.status(200).send({ m: mensaje, progresoEjercicio, progresoSubtema, progresoTema, progresoUsuario });
33             });
34           });
35         });
36       });
37     } else {
38       return res.status(200).send({ m: error });
39     }
40   });
41
42   .catch((error) => {
43     return res.status(200).send({ m: error });
44   });
45
46   .catch((error) => {
47     return res.status(200).send({ m: error });
48   });
49
50   .catch((error) => {
51     return res.status(200).send({ m: error });
52   });
53
54   .catch((error) => {
55     return res.status(200).send({ m: error });
56   });
57
58   .catch((error) => {
59     return res.status(200).send({ m: error });
60   });
61
62   .catch((error) => {
63     return res.status(200).send({ m: error });
64   });
65
66   .catch((error) => {
67     return res.status(200).send({ m: error });
68   });
69
70   .catch((error) => {
71     return res.status(200).send({ m: error });
72   });
73
74   .catch((error) => {
75     return res.status(200).send({ m: error });
76   });
77
78   .catch((error) => {
79     return res.status(200).send({ m: error });
80   });
81
82   .catch((error) => {
83     return res.status(200).send({ m: error });
84   });
85
86   .catch((error) => {
87     return res.status(200).send({ m: error });
88   });
89
90   .catch((error) => {
91     return res.status(200).send({ m: error });
92   });
93
94   .catch((error) => {
95     return res.status(200).send({ m: error });
96   });
97
98   .catch((error) => {
99     return res.status(200).send({ m: error });
100  });
101
102  });
103
104  });

```

Figura 189. Codificación HU040.

5.4 Pruebas

La Tabla 8 muestra las pruebas unitarias a desarrollar en la cuarta iteración.



Tabla 10. Pruebas unitarias a ejecutar en la cuarta iteración

ID	Descripción	Datos de Entrada	Salida Esperada	Resultado
PU050	Listar comentarios existentes	➤ idEjercicio: 27	Comentarios obtenidos.	Aprobada
PU051	Listar comentarios cuando no existe ningún ejercicio.	➤ idEjercicio: 999	No se encontraron comentarios	Aprobada
PU052	Registrar un nuevo comentario	➤ nombreUsuario: 'Usuario Test', ➤ contenido: 'Este es un comentario de prueba', ➤ idEjercicio: 27	Se registró el comentario con éxito.	Aprobada
PU053	Editar un comentario existente	➤ id: 13, ➤ nombreUsuario: 'Usuario Editado', ➤ contenido: 'Contenido del comentario editado'	Se editó el comentario con éxito	Aprobada
PU054	Editar un comentario inexistente.	➤ id: 999999, ➤ nombreUsuario: 'Usuario Editado', ➤ contenido: 'Contenido del comentario editado'	No se pudo editar el comentario	Aprobada
PU055	Editar un comentario con datos incompletos	➤ id: 13, ➤ nombreUsuario: 'Usuario Editado', ➤ contenido: null	No se pudo editar el comentario	Aprobada
PU056	Visualizar comentarios	➤ N/A	Comentarios listados	Aprobada
PU057	Visualizar progreso de un usuario existente	➤ idUsuario: 47	Progreso de los usuarios disponible para su visualización	Aprobada
PU058	Visualizar progreso de un usuario inexistente	➤ idUsuario: 9999999	No se encontró el progreso del usuario	Aprobada
PU059	Obtener progreso sin proporcionar información del usuario	➤ idUsuario: null	No se encontró el progreso del usuario	Aprobada
PU060	Registrar un nuevo periodo académico exitosamente	➤ mesInicio: '2025-03-01', ➤ mesFin: '2025-07-01'	Se registró el nuevo periodo académico y se realizó la copia de seguridad del periodo anterior con éxito	Aprobada
PU061	Obtener valoración existente de los temas	➤ idUsuario: 47, ➤ idTema: 32	Valoración existente obtenida con éxito.	Aprobada



PU062	Registrar nuevo contenido y enviar notificaciones por email.	➤ Datos para crear contenido (Temas, Subtemas, Ejercicios y Preguntas de Control)	Contenido creado con éxito y notificado a los usuarios.	Aprobada
-------	--	---	---	----------

A continuación, se presentan las pruebas unitarias desarrolladas en la quinta iteración.

```

1 describe('Test de listado de comentarios', () => {
2   it('Listar comentarios de un ejercicio con comentarios existentes', async () => {
3     const response = await api
4       .post(rutaComentarios + '/listarComentarios')
5       .send(bodyListar);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Comentarios obtenidos");
10    expect(Array.isArray(response.body.comentarios)).toBe(true);
11  });
12
13  it('Intento de listar comentarios cuando no hay ninguno', async () => {
14    const bodySinComentarios = {
15      idEjercicio: 999 // ID de ejercicio que no tiene comentarios
16    };
17
18    const response = await api
19      .post(rutaComentarios + '/listarComentarios')
20      .send(bodySinComentarios);
21
22    expect(response.statusCode).toBe(200);
23    expect(response.body.en).toBe(-1);
24    expect(response.body.m).toBe("No se encontraron comentarios");
25    expect(Array.isArray(response.body.comentarios)).toBe(true);
26    expect(response.body.comentarios).toHaveLength(0);
27  });
28 });

```

Figura 190. Pruebas unitarias (PU050, PU051), listar comentarios.

```

PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCinco
> camigo@1.0.0 testIteracionCinco
> cross-env BD=camigo_prueba jest --verbose --silent tests/comentario/comentario.test.js

PASS tests/comentario/comentario.test.js
  Test de listado de comentarios
    ✓ Listar comentarios de un ejercicio con comentarios existentes (77 ms)
    ✓ Intento de listar comentarios cuando no hay ninguno (10 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.128 s, estimated 2 s
Jest did not exit one second after the test run has completed.

```

Figura 191. Ejecución pruebas unitarias (PU050, PU051), listar comentarios.

```
1 describe('Test de registro de comentarios', () => {
2   it('Registrar un nuevo comentario exitosamente', async () => {
3     const response = await api
4       .post(rutaComentarios + '/registrarComentario')
5       .send(bodyRegistrar);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Se registró el comentario con éxito");
10    expect(response.body.idComentario).toBeDefined();
11    expect(typeof response.body.idComentario).toBe('number');
12  });
13 });
```

Figura 192. Pruebas unitarias (PU052), registrar comentarios.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCinco
> camigo@1.0.0 testIteracionCinco
> cross-env BD=camigo_prueba jest --verbose --silent tests/comentario/comentario.test.js

PASS tests/comentario/comentario.test.js
  Test de registro de comentarios
    ✓ Registrar un nuevo comentario exitosamente (126 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 1.531 s, estimated 7 s
Jest did not exit one second after the test run has completed.
```

Figura 193. Ejecución pruebas unitarias (PU052), registrar comentarios.

```
1 describe('Test de edicion de comentarios', () => {
2   it('Editar un comentario existente', async () => {
3     const response = await api
4       .post(rutaComentarios + '/editarComentario')
5       .send(bodyEditar);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Se editó el comentario con éxito");
10    expect(response.body.idComentario).toBe(bodyEditar.id);
11  });
12
13   it('Intentar editar un comentario inexistente', async () => {
14     const bodyComentarioInexistente = {
15       ..bodyEditar,
16       id: 999 // ID que no existe en la base de datos
17     };
18
19     const response = await api
20       .post(rutaComentarios + '/editarComentario')
21       .send(bodyComentarioInexistente);
22
23     expect(response.statusCode).toBe(200);
24     expect(response.body.en).toBe(-1);
25     expect(response.body.m).toBe("No se pudo editar el comentario");
26   });
27
28   it('Intentar editar un comentario con datos incompletos', async () => {
29     const bodyIncompleto = {
30       id: 1,
31       // Falta nombreUsuario y contenido
32     };
33
34     const response = await api
35       .post(rutaComentarios + '/editarComentario')
36       .send(bodyIncompleto);
37
38     expect(response.statusCode).toBe(200);
39     expect(response.body.en).toBe(-1);
40     expect(response.body.m).toBe("No se pudo editar el comentario");
41   });
42 });
```

Figura 194. Pruebas unitarias (PU053, PU054, PU055), editar comentarios

Figura 195. Pruebas unitarias (PU053, PU054, PU055), editar comentarios.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCinco
> camigo@1.0.0 testIteracionCinco
> cross-env BD=camigo_prueba jest --verbose --silent tests/comentario/comentario.test.js

PASS tests/comentario/comentario.test.js
  Test de edición de comentarios
    ✓ Editar un comentario existente (83 ms)
    ✓ Intentar editar un comentario inexistente (9 ms)
    ✓ Intentar editar un comentario con datos incompletos (5 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.087 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 196. Ejecución pruebas unitarias (PU053, PU054, PU055), editar comentarios.

```
1 describe('Test de listado de comentarios', () => {
2   it('Listar comentarios de un ejercicio con comentarios existentes', async () => {
3     const response = await api
4       .post(rutaComentarios + '/listarComentarios')
5       .send(bodyListar);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Comentarios obtenidos");
10    expect(Array.isArray(response.body.comentarios)).toBe(true);
11  });
12
13  it('Intento de listar comentarios cuando no hay ninguno', async () => {
14    const bodySinComentarios = {
15      idEjercicio: 999 // ID de ejercicio que no tiene comentarios
16    };
17
18    const response = await api
19      .post(rutaComentarios + '/listarComentarios')
20      .send(bodySinComentarios);
21
22    expect(response.statusCode).toBe(200);
23    expect(response.body.en).toBe(-1);
24    expect(response.body.m).toBe("No se encontraron comentarios");
25    expect(Array.isArray(response.body.comentarios)).toBe(true);
26    expect(response.body.comentarios).toHaveLength(0);
27  });
28 });
```

Figura 197. Pruebas unitarias (PU056), visualizar comentarios.



```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCinco
> camigo@1.0.0 testIteracionCinco
> cross-env BD=camigo_prueba jest --verbose --silent tests/comentario/comentario.test.js

PASS tests/comentario/comentario.test.js
  Test de listado de comentarios
    ✓ Listar comentarios de un ejercicio con comentarios existentes (77 ms)
    ✓ Intento de listar comentarios cuando no hay ninguno (10 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.128 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 198. Ejecución pruebas unitarias (PU056), visualizar comentarios.

```
1 describe('Test de progreso de usuario', () => {
2   it('Obtener progreso de un usuario existente', async () => {
3     const response = await api
4       .post(rutaUsuario + '/progresoUsuario')
5       .send(bodyProgresoExistente);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     expect(response.body.m).toBe("Progreso del usuario");
10    expect(response.body.progreso).toBeDefined();
11    expect(typeof response.body.progreso).toBe('number');
12  });
13
14  it('Intentar obtener progreso de un usuario inexistente', async () => {
15    const response = await api
16      .post(rutaUsuario + '/progresoUsuario')
17      .send(bodyProgresoInexistente);
18
19    expect(response.statusCode).toBe(200);
20    expect(response.body.en).toBe(-1);
21    expect(response.body.m).toBe("No se encontro el progreso del usuario");
22  });
23
24  it('Intentar obtener progreso sin proporcionar idUsuario', async () => {
25    const response = await api
26      .post(rutaUsuario + '/progresoUsuario')
27      .send({});
28
29    expect(response.statusCode).toBe(200);
30    expect(response.body.en).toBe(-1);
31    expect(response.body.m).toBe("No se encontro el progreso del usuario");
32  });
33 });
```

Figura 199. Pruebas unitarias (PU057, PU058, PU059), visualizar progreso.



```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCinco
> camigo@1.0.0 testIteracionCinco
> cross-env BD=camigo_prueba jest --verbose --silent tests/comentario/comentario.test.js

PASS tests/comentario/comentario.test.js
  Test de progreso de usuario
    ✓ Obtener progreso de un usuario existente (107 ms)
    ✓ Intentar obtener progreso de un usuario inexistente (8 ms)
    ✓ Intentar obtener progreso sin proporcionar idUsuario (4 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:       1.762 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 200. Ejecución pruebas unitarias (PU057, PU058, PU059), visualizar progreso.

```
1 describe('Test de registro de periodo académico', () => {
2   it('Registrar un nuevo periodo académico exitosamente', async () => {
3     const response = await api
4       .post(rutaPeriodo + '/registrarPeriodoAcademico')
5       .send(bodyPeriodoValido);
6
7     expect(response.statusCode).toBe(200);
8     // Asumiendo que copiaSeguridadPeriodoAnterior devuelve un mensaje exitoso
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Se registró el periodo académico con éxito");
11    expect(response.body.idPeriodo).toBeDefined();
12    expect(typeof response.body.idPeriodo).toBe('number');
13  });
14 });
```

Figura 201. Pruebas unitarias (PU060), registrar periodo académico.

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCinco
> camigo@1.0.0 testIteracionCinco
> cross-env BD=camigo_prueba jest --verbose --silent tests/comentario/comentario.test.js

PASS tests/comentario/comentario.test.js
  Test de registro de periodo académico
    ✓ Registrar un nuevo periodo académico exitosamente (159 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:       1.205 s, estimated 17 s
Jest did not exit one second after the test run has completed.
```

Figura 202. Ejecución pruebas unitarias (PU060), registrar periodo académico.

```
1 describe('Test de registro de periodo academico', () => {
2   it('Registrar un nuevo periodo academico exitosamente', async () => {
3     const response = await api
4       .post(rutaPeriodo + '/registrarPeriodoAcademico')
5       .send(bodyPeriodoValido);
6
7     expect(response.statusCode).toBe(200);
8     expect(response.body.en).toBe(1);
9     // Verificamos que el mensaje sea uno de los dos posibles mensajes de exito
10    expect([
11      "Se registró el nuevo periodo academico y se realizo la copia de seguridad del periodo anterior con exito",
12      "Se registro el periodo academico. No se realizo copia de seguridad por ser el primer periodo."
13    ]).toContain(response.body.m);
14
15    // Verificamos que exista el ID del periodo academico
16    expect(response.body.idPeriodoAcademico).toBeDefined();
17    expect(typeof response.body.idPeriodoAcademico).toBe('number');
18
19    // Si hay un periodo anterior, verificamos su ID
20    if (response.body.m.includes("copia de seguridad del periodo anterior")) {
21      expect(response.body.idPeriodoAnterior).toBeDefined();
22      expect(typeof response.body.idPeriodoAnterior).toBe('number');
23    }
24  });
25
26 });
```

Figura 203. Pruebas unitarias (PU061), obtener valoración existente de los temas (Temas destacados).

```
PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionCinco
> camigo@1.0.0 testIteracionCinco
> cross-env BD=camigo_prueba jest --verbose --silent tests/comentario/comentario.test.js
PASS tests/comentario/comentario.test.js
  Test de consulta de valoración
    ✓ Obtener valoración existente (104 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 1.781 s, estimated 2 s
Jest did not exit one second after the test run has completed.
```

Figura 204. Ejecución pruebas unitarias (PU061), obtener valoración existente de los temas (Temas destacados).

```
1 describe('Test de registro de temas', () => {
2   // Prueba para registrar un tema exitosamente
3   it('Registrar un nuevo tema con notificaciones', async () => {
4     const response = await api
5       .post(rutaTemas + '/registrarTema')
6       .send(bodyRegistrar);
7
8     expect(response.statusCode).toBe(200);
9     expect(response.body.en).toBe(1);
10    expect(response.body.m).toBe("Se registró el tema con exito y se enviaron correos de notificación");
11    expect(response.body.idTema).toBeDefined();
12    expect(typeof response.body.idTema).toBe('number');
13  });
14 });
```

Figura 205. Pruebas unitarias (PU062), registrar nuevo contenido y enviar notificaciones por email



```

PS C:\Users\juanc\Documents\repositorio_GitHub\server_camigo> npm run testIteracionUnoParteDos
> camigo@1.0.0 testIteracionUnoParteDos
> cross-env BD=camigo_prueba jest --verbose --silent tests/tema/tema.test.js

PASS tests/tema/tema.test.js
  Test de registro de temas
    ✓ Registrar un nuevo tema con notificaciones (117 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.129 s, estimated 7 s
Jest did not exit one second after the test run has completed.

```

Figura 206. Ejecución pruebas unitarias (PU062), registrar nuevo contenido y enviar notificaciones por email

La prueba unitaria se enfoca en el registro de contenido del software de apoyo, abarcando temas, subtemas, ejercicios y preguntas de control. Las pruebas se realizan en cuatro iteraciones secuenciales: **Primera Iteración** para registro de temas, **Segunda Iteración** para subtemas, **Tercera Iteración** para ejercicios y **Cuarta Iteración** para preguntas de control. Cada prueba unitaria verifica el envío de notificaciones, ejemplificado con el registro de temas, confirmando la efectividad de las notificaciones por correo electrónico véase figura 206.

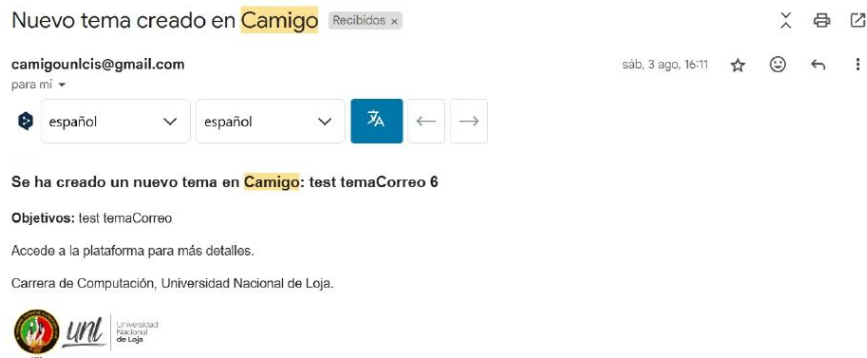


Figura 207. Notificación de nuevo contenido mediante correo electrónico.




unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Para constancia y validación de la información expuesta, firma a continuación el tutor del Trabajo de Integración Curricular, en su calidad de docente guía en el desarrollo del proyecto:

Ing. José Oswaldo Guamán Quinche, Mg. Sc.	 FIRMA ELECTRÓNICA DEL: JOSE OSWALDO GUAMAN QUINCHE
---	--

Anexo 6: Arquitectura 4+1

Arquitectura 4+1

Proyecto: Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja

Ficha del documento

Ficha del documento

Fecha	Revisión	Autor	Verificado
26/01/2024	26/01/2024	Juan Carlos Armijos	Ing. José Oswaldo Guamán.

Documento validado por las partes en fecha: 26/01/2024

Por la universidad	Firma
Ing. José Oswaldo Guamán.	



1 Introducción

El software de apoyo para la enseñanza de programación en C, alineado al plan de estudios del primer ciclo de la carrera de Computación en la Universidad Nacional de Loja. Evalúa su aceptación mediante el Modelo de Aceptación Tecnológica (TAM), promoviendo un aprendizaje adaptado y efectivo para los estudiantes.

Este documento elabora el informe de la arquitectura de software para la "Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja". La arquitectura del sistema se resume en 5 vistas o concurrencias. El documento sigue el modelo arquitectónico 4+1.

1.1 Propósito

Este documento presenta una descripción integral de la arquitectura del sistema, empleando diversas vistas arquitectónicas para abordar distintos aspectos del mismo. Su propósito es capturar y comunicar las decisiones arquitectónicas fundamentales adoptadas en el desarrollo del sistema.

Este documento detalla la arquitectura del software a través de cinco vistas distintas, abordando tanto el comportamiento estático como dinámico del sistema. Incluye todos los diagramas necesarios junto con sus respectivas descripciones. La utilización del modelo de vista 4 + 1 garantiza una representación más global del software, lo que facilita a diversas partes interesadas encontrar la información que requieren en el documento de arquitectura.

1.2 Alcance

El documento de arquitectura de software se aplica a cada aspecto estático y dinámico del sistema. Dado que el modelo de vista 4 + 1 se utiliza como modelo de referencia, incorpora muchas vistas del sistema, lo que hace que el documento sea completo y coherente.

Bajo el comportamiento estático del sistema, el documento analiza los diagramas de clase, los diagramas de paquetes y otros diseños de arquitectura estática. Los aspectos dinámicos del sistema se elaboran utilizando realizaciones de casos de uso y diagramas de secuencia del sistema.

1.3 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
SA	Software de apoyo
UNL	Universidad Nacional de Loja

1.4 Resumen

El informe presentará un análisis detallado de la arquitectura del software de apoyo especializado en el aprendizaje de programación en C. Otras secciones cubren la representación arquitectónica del proyecto, incluida la representación arquitectónica, los objetivos y limitaciones arquitectónicas y las realizaciones de casos de uso. Las secciones posteriores cubren los detalles específicos detallados de las 4 vistas principales (vista lógica, vista de proceso, vista de desarrollo y vista de implementación) del sistema.



2 Representación Arquitectónica

Esta sección detalla la arquitectura usando las vistas definidas en el modelo "4 + 1". Las vistas utilizadas para documentar el Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja

2.1 Escenarios

Audiencia: Todas las partes interesadas del sistema, incluidos los usuarios finales.

Área: Describe el conjunto de escenarios y / o casos de uso que representan alguna funcionalidad importante y central del sistema. Describe los actores y casos de uso del sistema.

2.2 Vista Lógica

Audiencia: Diseñadores, Programadores.

Área: Requisitos funcionales, jerarquía de objetos, capas de sistema. Describe el diseño del modelo de objetos.

2.3 Vista de Procesos

Audiencia: Programadores.

Área: Aborda los aspectos de dinamismo del software. Expone los procesos del sistema y como se comunican.

2.4 Vista de desarrollo

Audiencia: Programadores, Probadores.

Área: Se detallan los módulos del sistema

2.5 Vista Física

Audiencia: Administradores de bases de datos, Ingenieros de sistemas, Gestores de despliegue

Área: Describe los elementos persistentes de importancia arquitectónica en el modelo de datos. Describe la asignación del software al hardware.

3 Metas y limitaciones arquitectónicas

3.1 Del lado del servidor

El software de apoyo se alojará en producción en los servidores de la carrera de computación de la Universidad Nacional de Loja. Y las limitaciones vendrán condicionadas por la infraestructura de servidores de la carrera de computación.

3.2 Del lado del cliente

El software de apoyo será consumido por cualquier navegador dentro del laboratorio de software de la carrera. Se necesitará tener de conexión a internet



para utilizar la aplicación y sus funcionalidades.

3.3 Seguridad

Las características de seguridad son manejadas por los encargados del proyecto. Se les dará funciones de acceso completo tanto en la aplicación como en los niveles de la base de datos. Las cuentas de usuario serán creadas por los mismos usuarios. Los usuarios finales necesitarán tener acceso mediante sus credenciales. Todas las contraseñas están encriptadas para garantizar una mayor seguridad. La página de administración de la página será únicamente accedida por aquellos encargados del proyecto.

3.4 Persistencia

Todos los datos serán guardados en el servidor de la carrera de computación, específicamente en una base de datos relacional, en este caso MySQL en su versión comunitaria.

3.5 Confiabilidad/Disponibilidad

El software de apoyo se someterá a pruebas unitarias y de aceptación antes de ser implementado a los servidores del laboratorio de software en la carrera de computación, para asegurarse que la aplicación es confiable.

3.6 Rendimiento

El rendimiento del software de apoyo puede depender del hardware disponible en los servidores de la carrera, la red y las capacidades de conexión a Internet.

3.7 Portabilidad

El software de apoyo será diseñado para ser un producto independiente y completo que pueda ser usado en cualquier dispositivo con acceso a internet.

3.8 Herramientas de desarrollo

IDE de programación: Visual Studio Code.

Backend: Node js + Express.

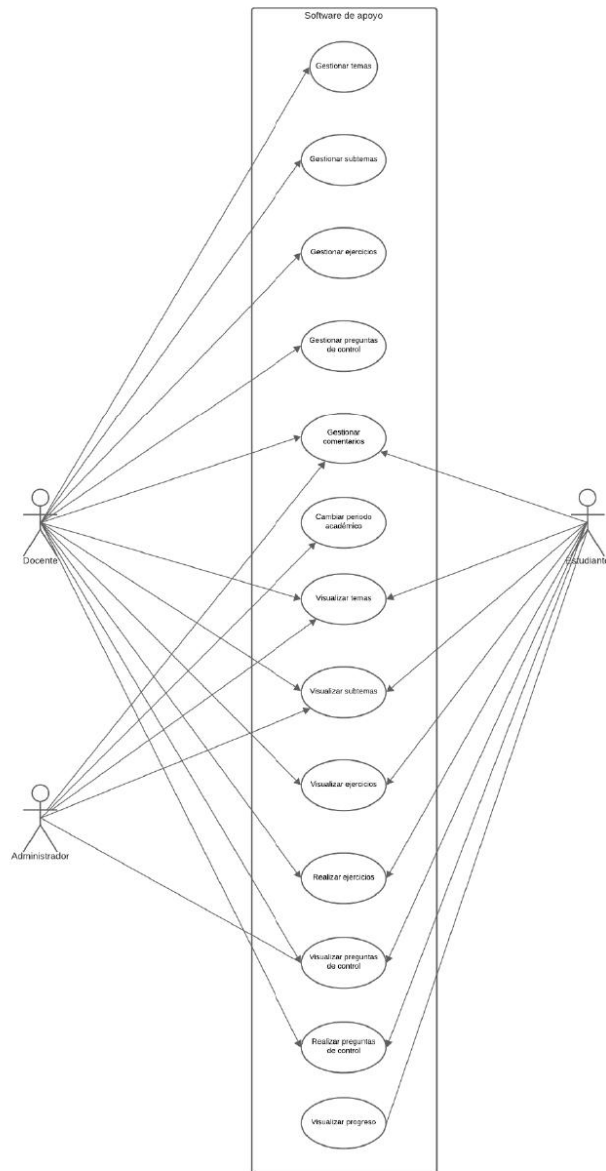
Frontend: React js, Material UI.

Base de datos: MySQL.



4 Vista de escenarios

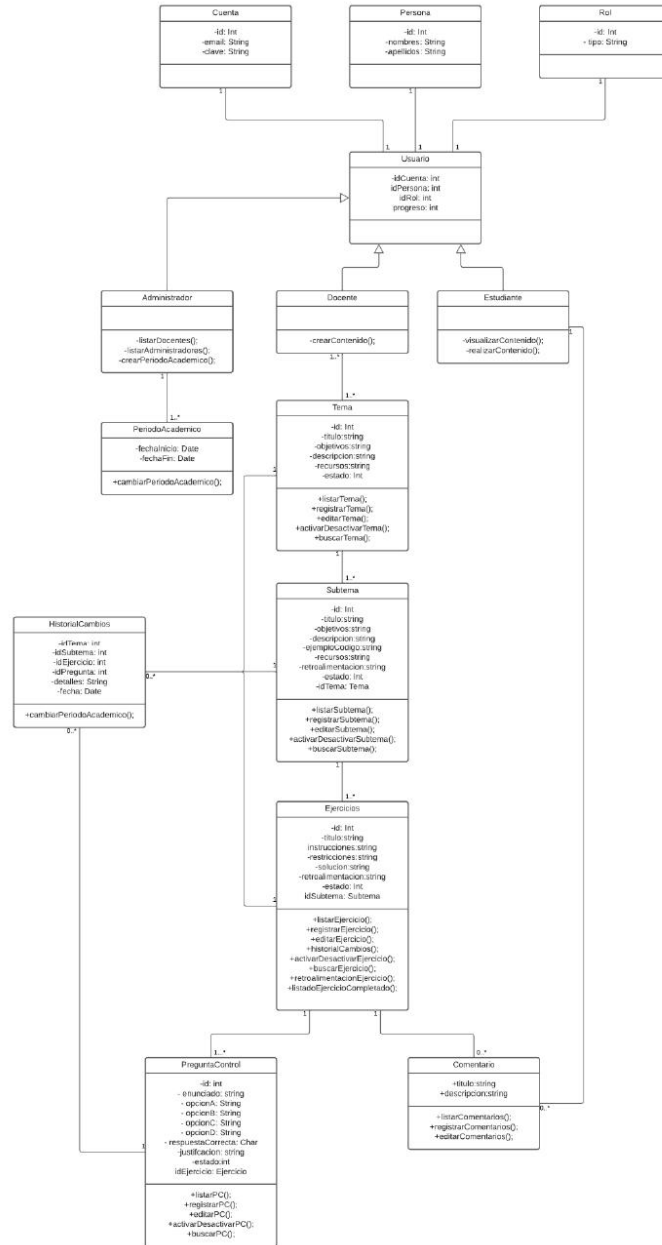
4.1 Diagrama de casos de uso general.





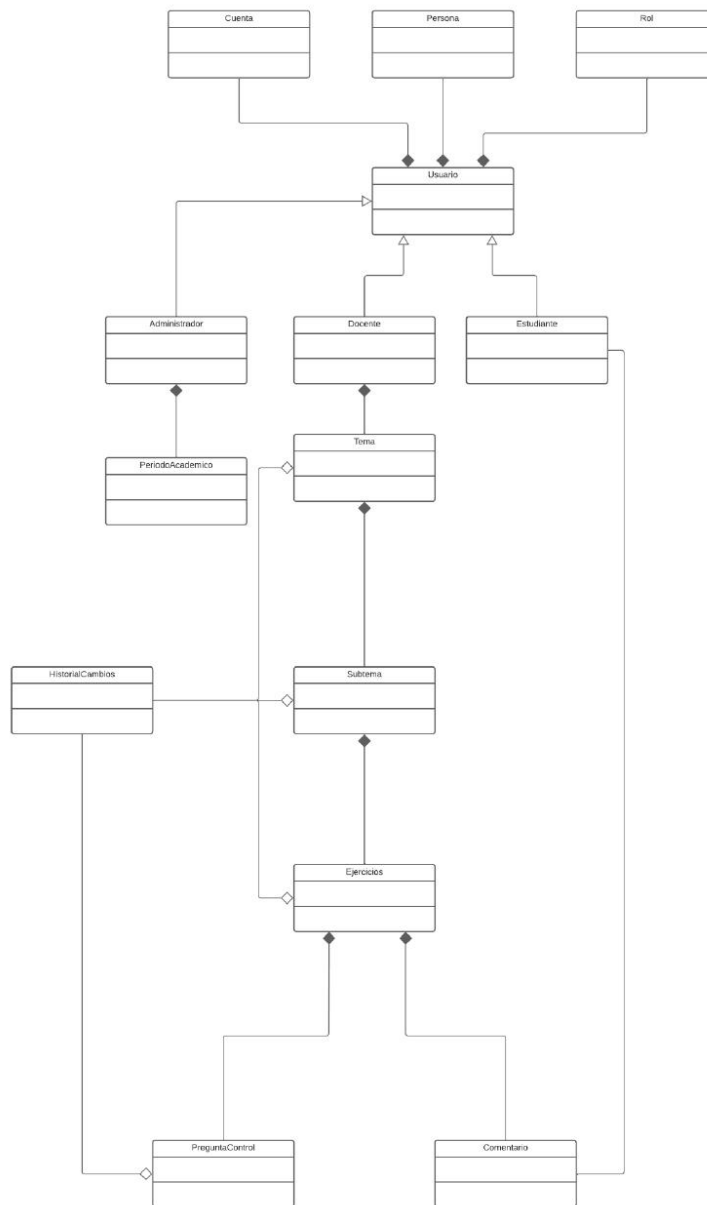
5 Vista lógica

5.1 Diagrama de clases





5.2 Diagrama de dominio

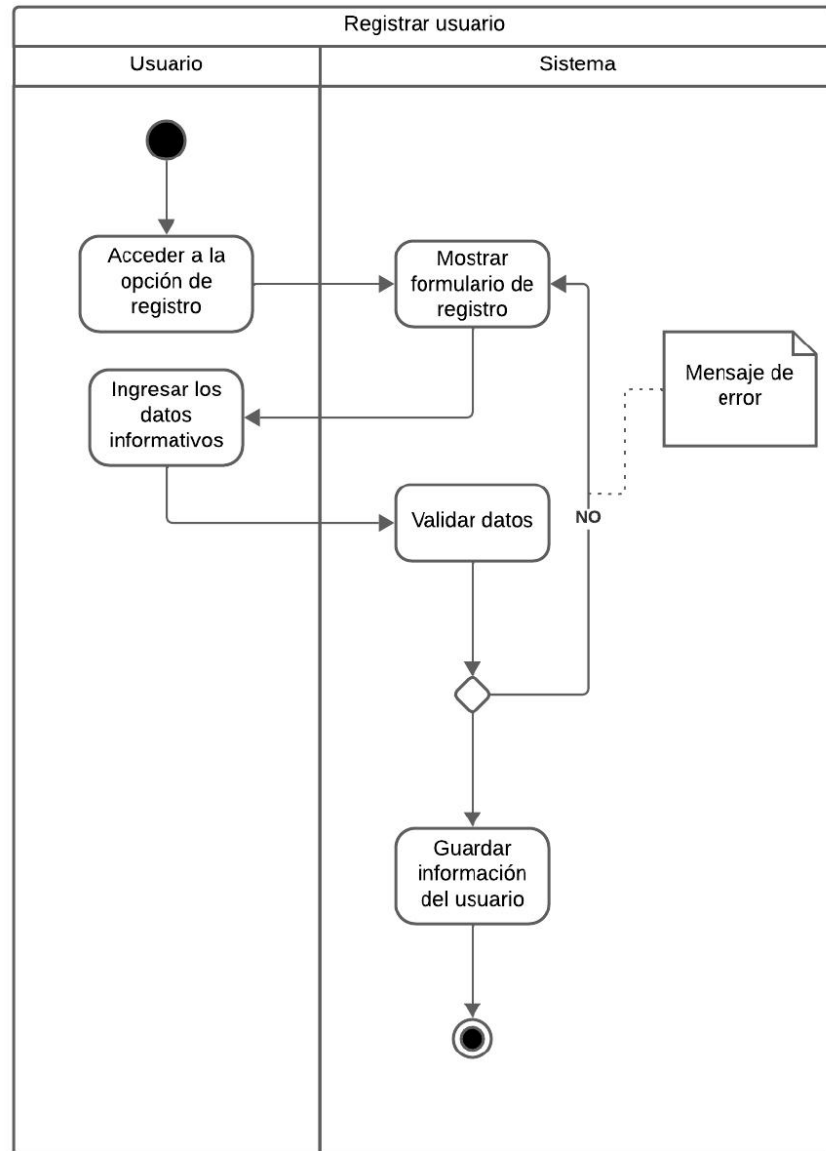


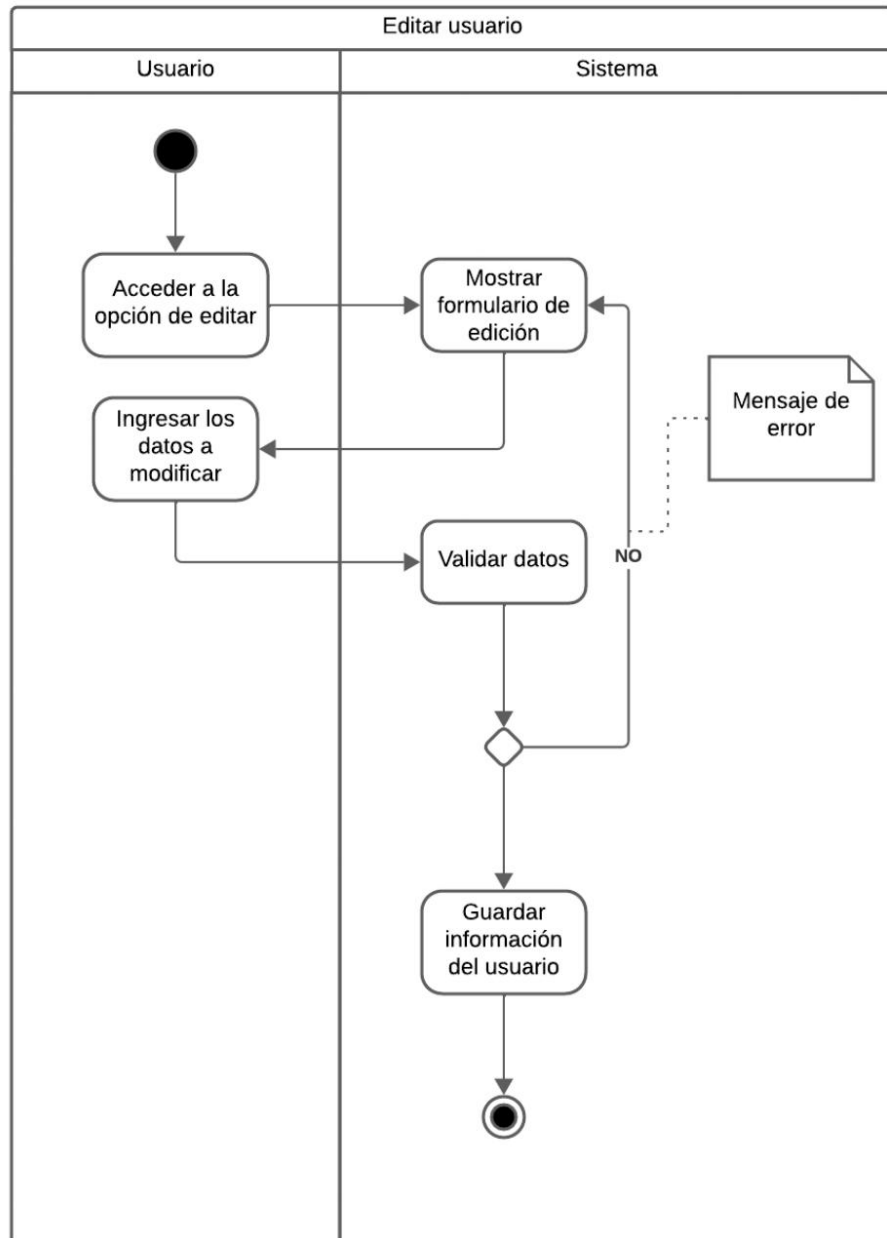


6 Vista de procesos

6.1 Diagrama de actividades

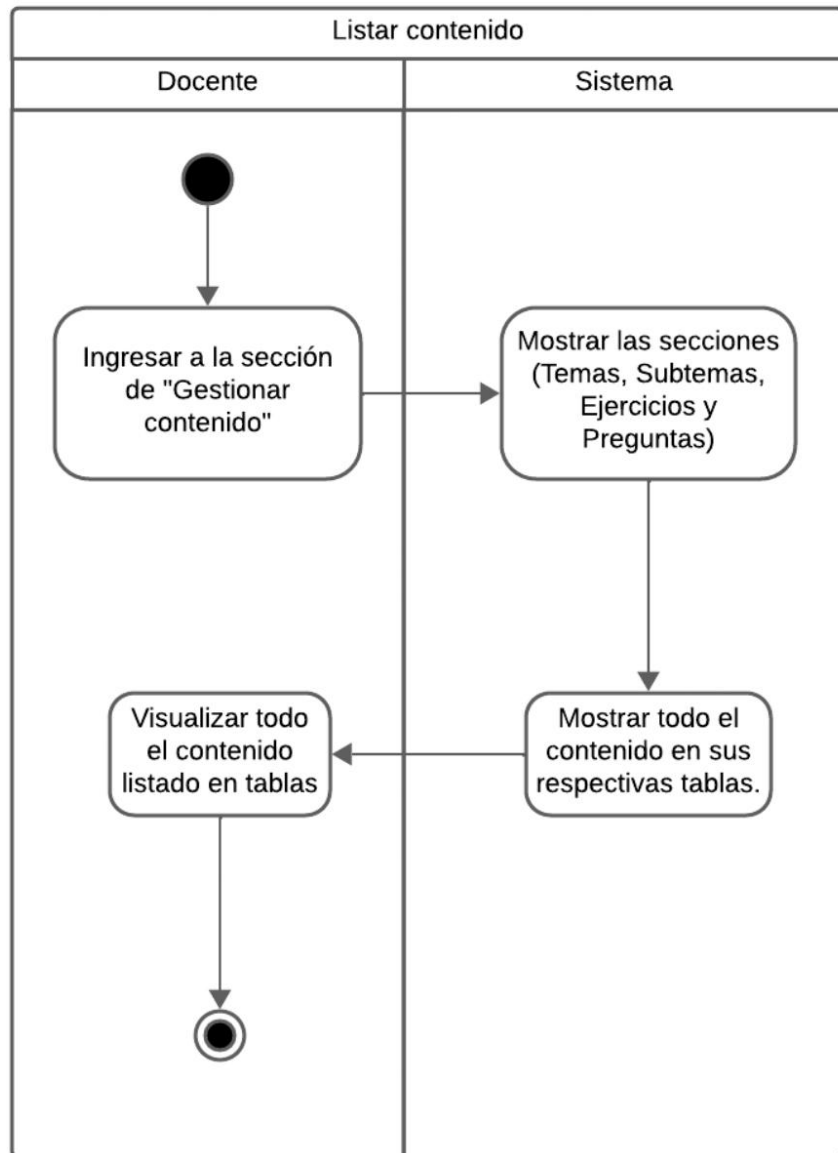
6.1.1 Gestionar usuarios

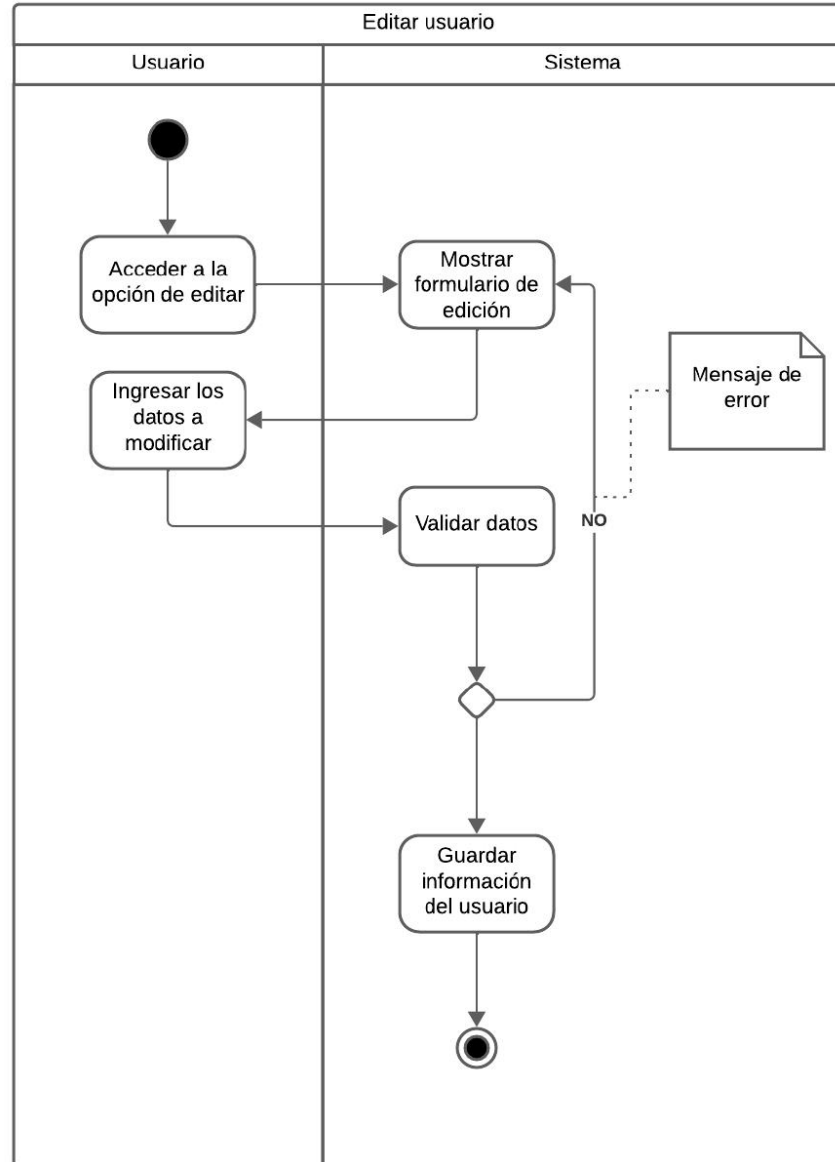


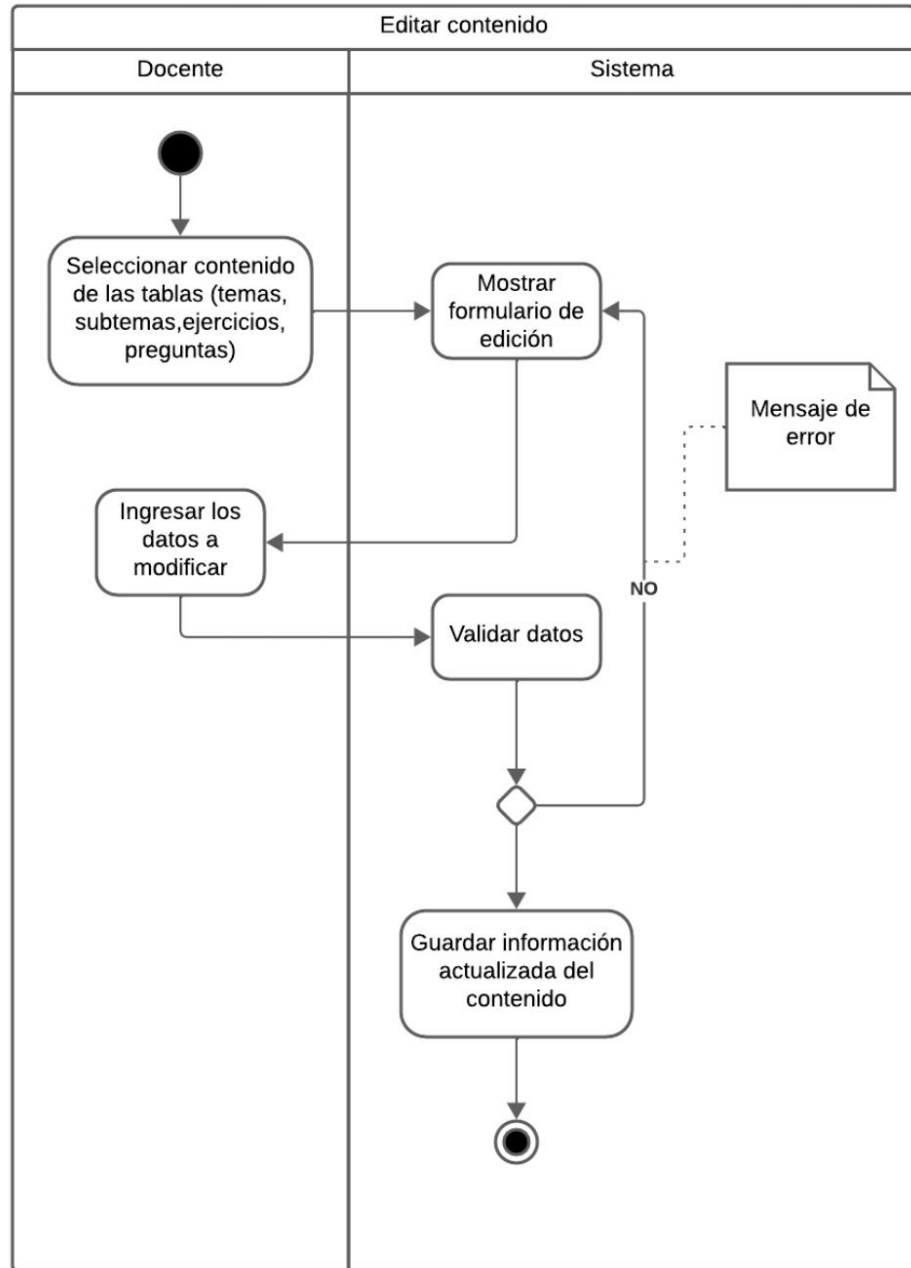


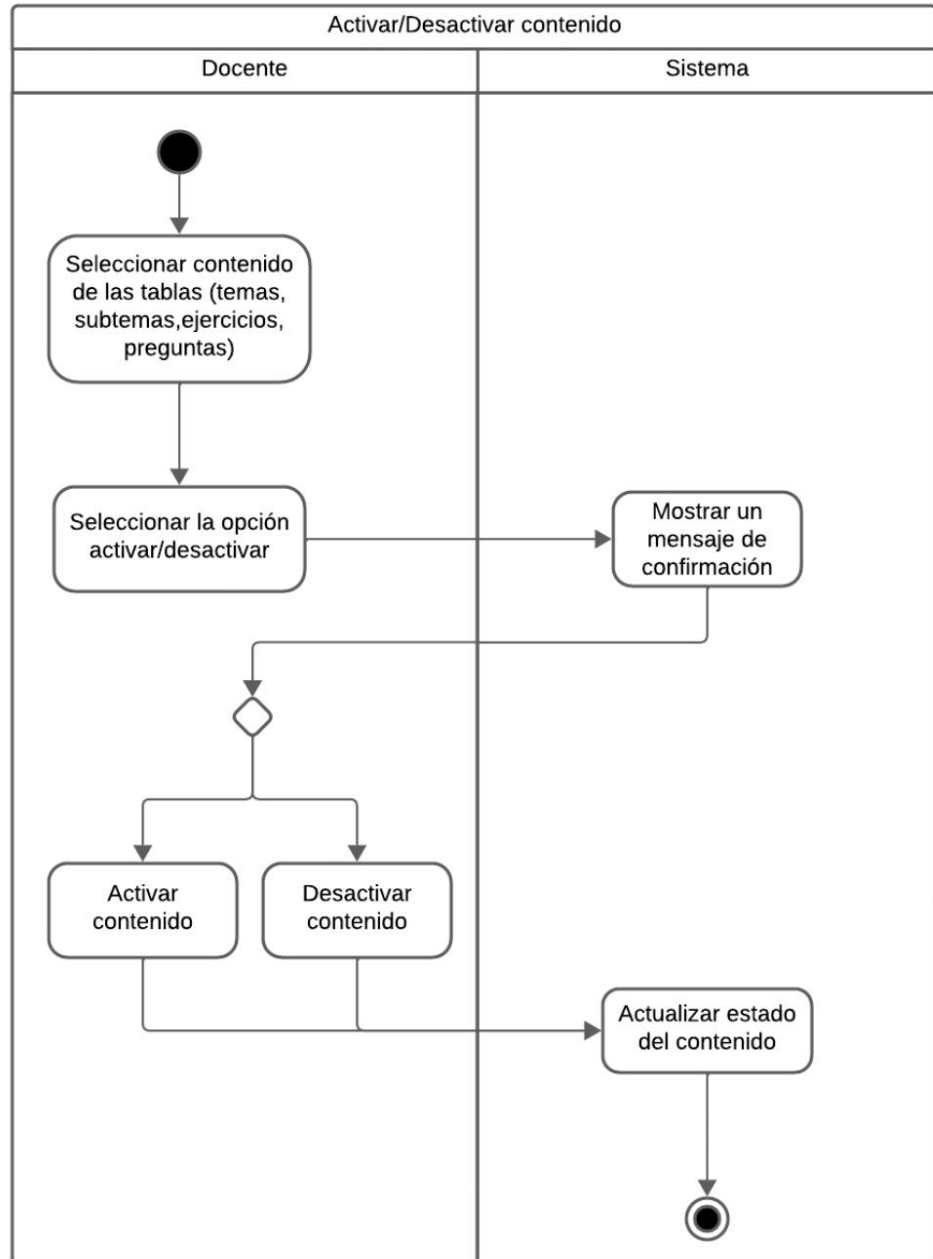


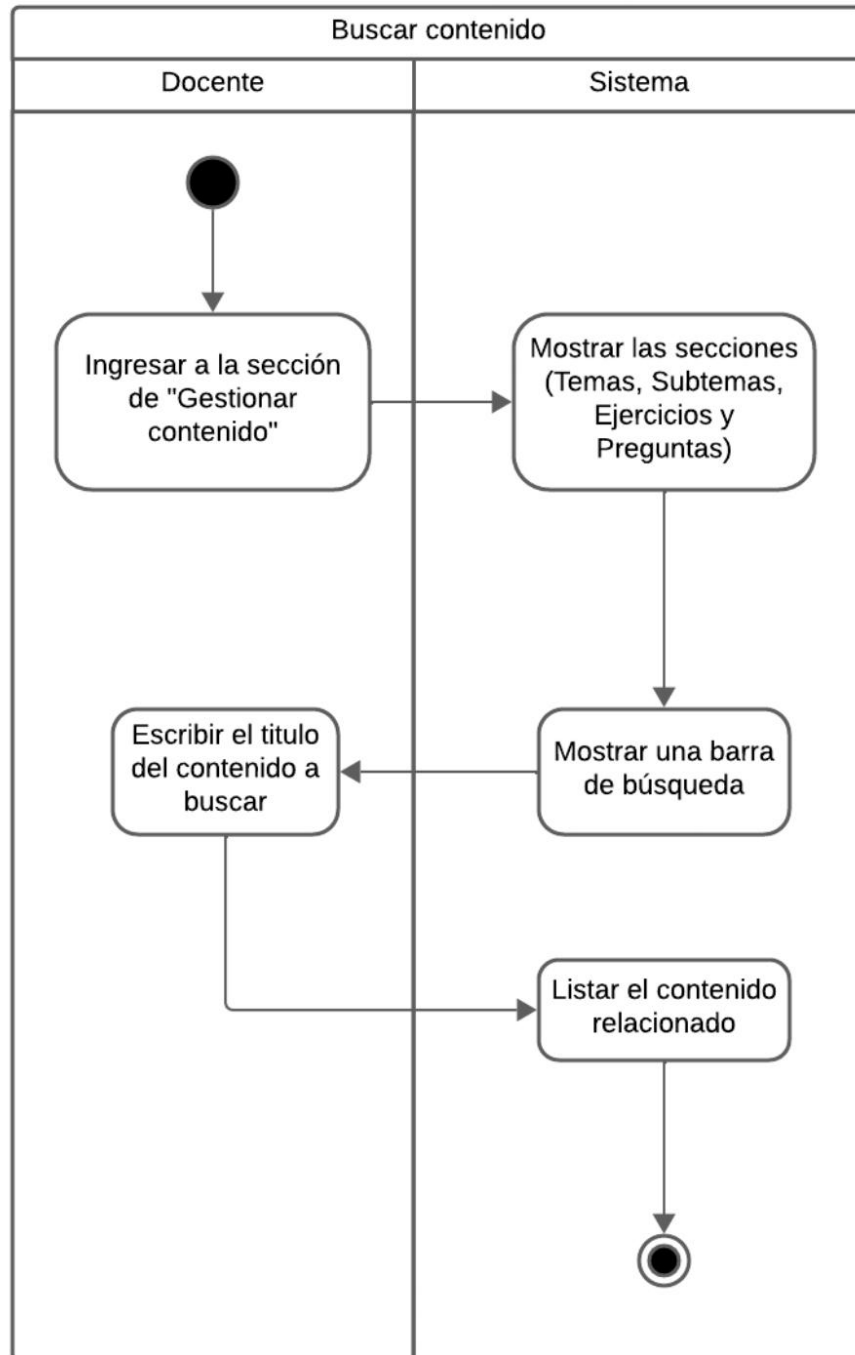
6.1.2 Gestionar contenido (Docente)





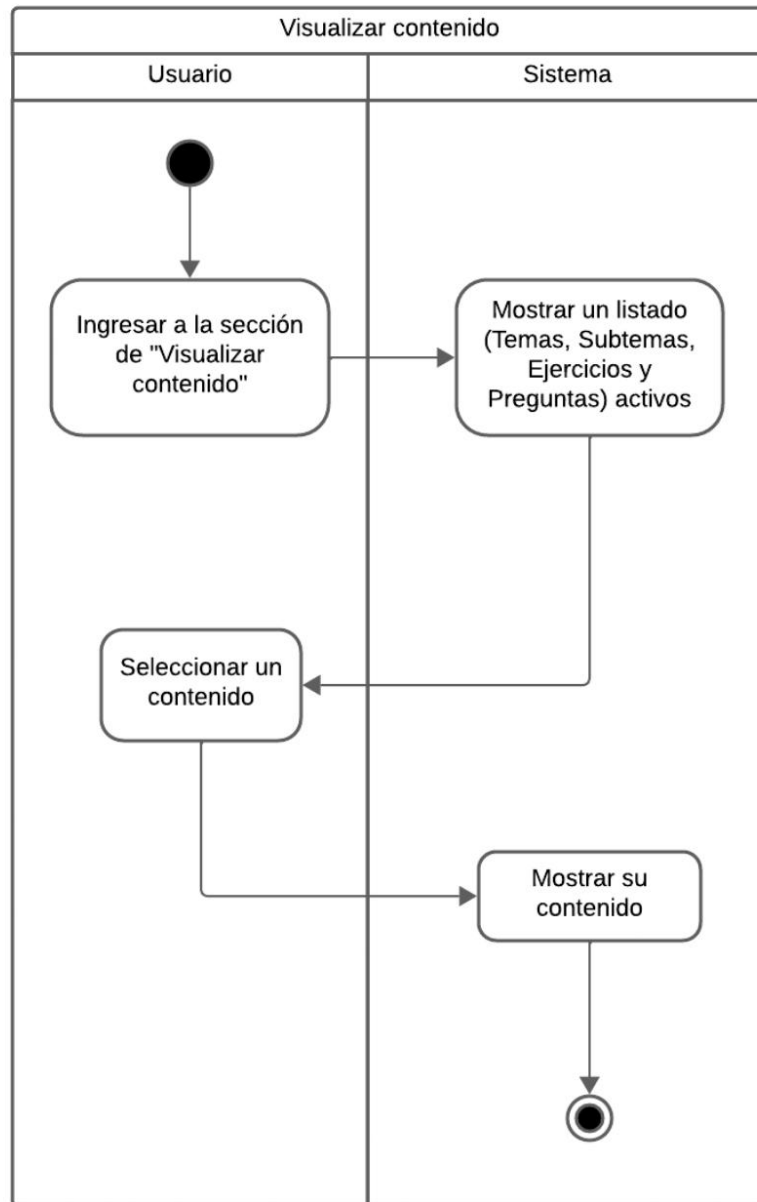






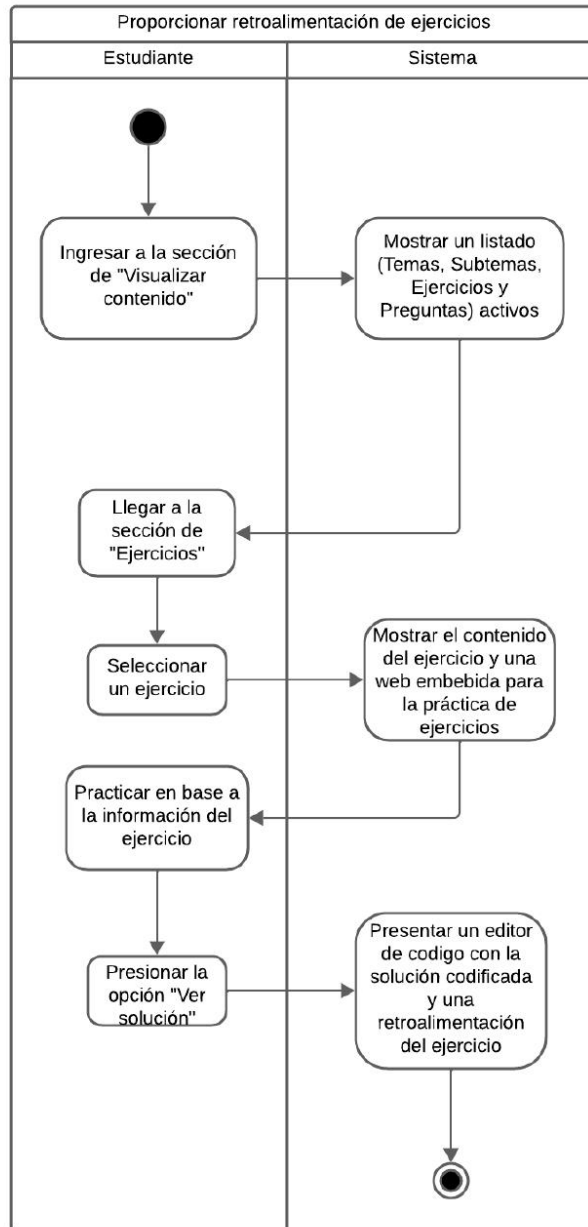


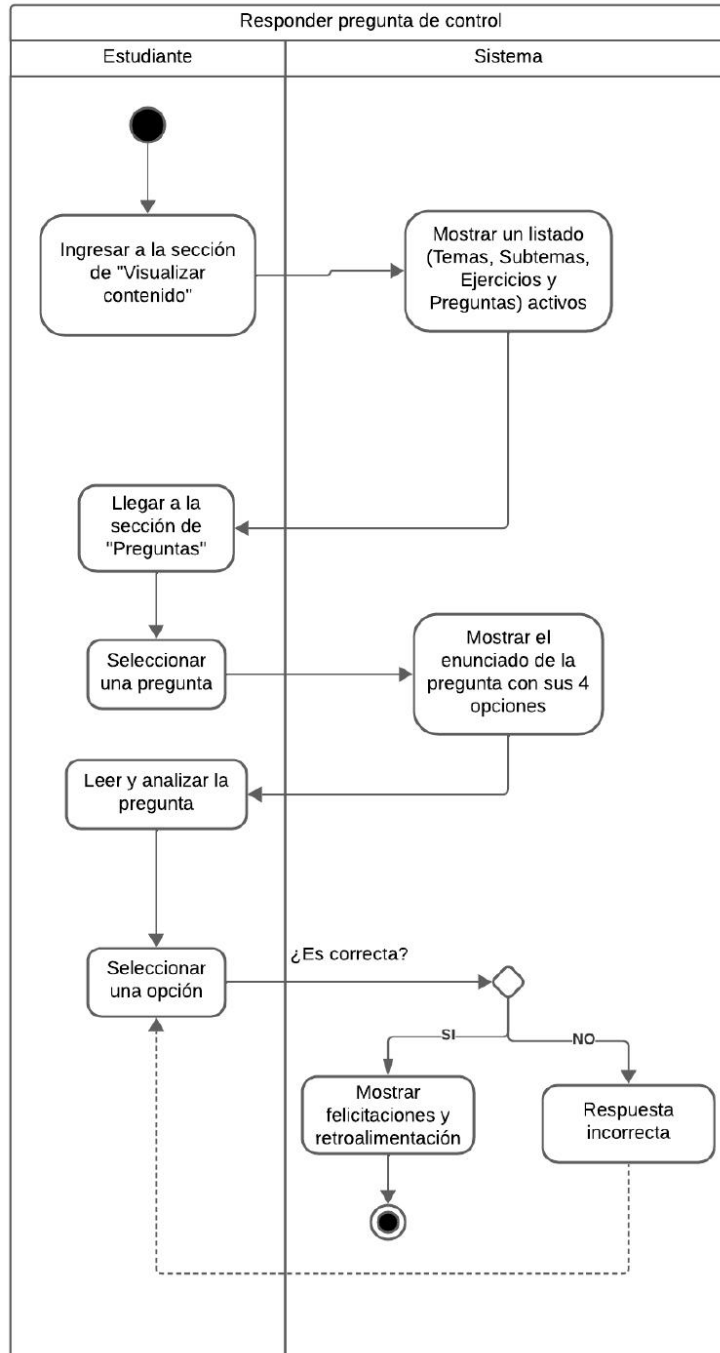
6.1.3 Visualizar contenido





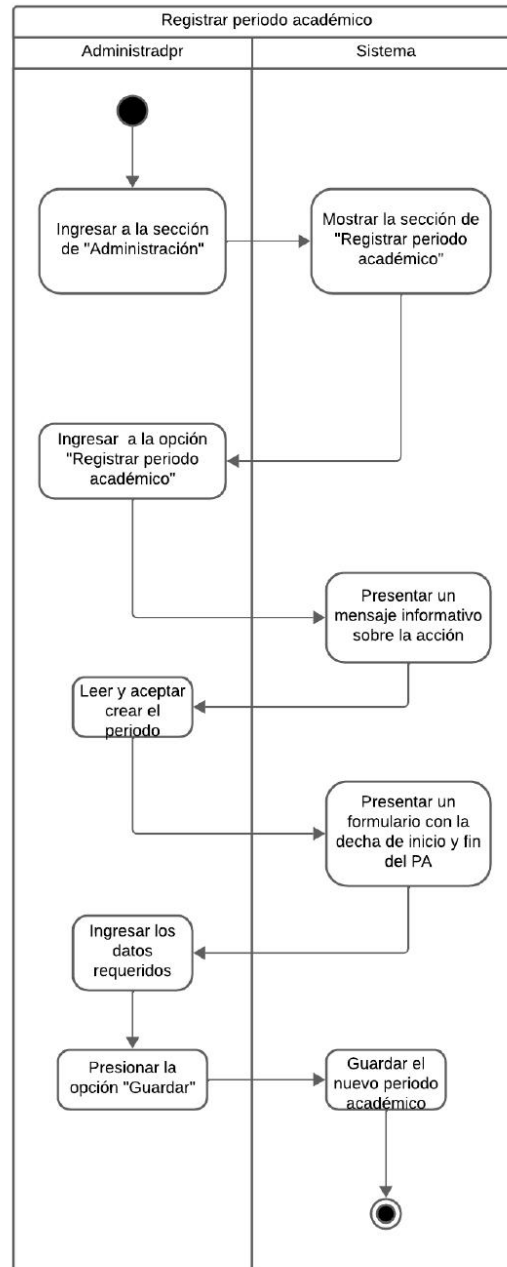
6.1.4 Practica y evaluación (Estudiante)







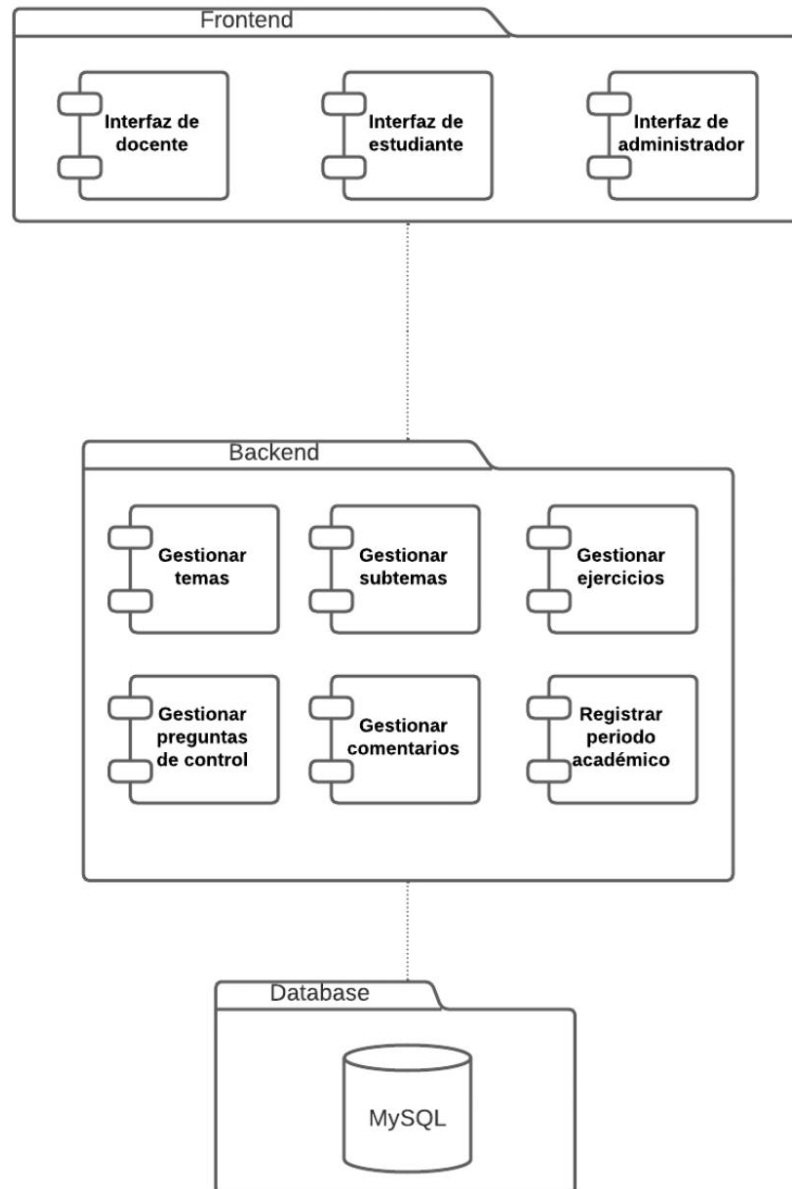
6.1.5 Administración (Administrador)





7 Vista de desarrollo

7.1 Diagrama de componentes

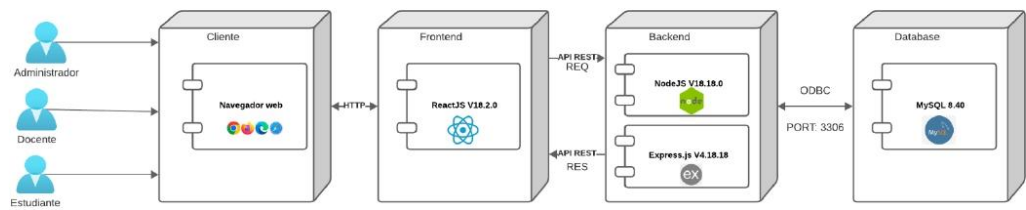




8 Vista física

A continuación, se presenta el **diagrama de despliegue**, el cual ilustra la arquitectura física del sistema, mostrando los componentes de hardware y software involucrados, así como sus relaciones y configuraciones. Este diagrama proporciona una visión clara de cómo se implementa el sistema en su entorno operativo, destacando la distribución de los nodos, procesos y conexiones necesarias para su funcionamiento.

8.1 Diagrama de despliegue



Ing. José Oswaldo Guamán Quinche, Mg. Sc.



Anexo 7: Resultados de la aceptación tecnológica del sistema mediante TAM



unl
Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Revisado y aprobado por:**
 - Ing. José Oswaldo Guamán, director del Trabajo de Integración Curricular.

Descripción: El presente anexo documenta los resultados obtenidos mediante la aplicación de encuestas destinadas a evaluar la aceptación tecnológica del sistema, utilizando como marco de referencia el Modelo de Aceptación Tecnológica (TAM). Este documento acredita la realización de dichas encuestas a estudiantes del primer ciclo académico de la carrera de Computación. En el mismo, se detallan las preguntas formuladas junto con sus correspondientes resultados estadísticos, expresados en términos porcentuales.



RESULTADOS DE LA ACEPTACIÓN TECNOLÓGICA DEL SISTEMA MEDIANTE EL MODELO TAM

Preguntas

La **Tabla 1** presenta el conjunto de preguntas diseñadas para evaluar la aceptación tecnológica mediante el Modelo de Aceptación Tecnológica (TAM). Para este estudio TIC, se han implementado los cuatro constructos fundamentales que se detallan en dicha tabla.

Tabla 1. Pruebas de aceptación finales en base a las Historias de Usuario

Constructo	Identificadores	Preguntas
Utilidad percibida (UP)	UP1	¿Considera que el software de apoyo especializado en el aprendizaje de programación en C facilita el proceso de comprensión de los conceptos de programación en C en comparación con otras herramientas como por ejemplo HackerRank, Khan Academy?
	UP2	¿Considera que el software de apoyo especializado en el aprendizaje de programación en C es práctico de utilizar en comparación con otras herramientas como por ejemplo HackerRank, Khan Academy?
	UP3	¿Considera que es útil el software de apoyo especializado en el aprendizaje de programación en C, como herramienta para el proceso de enseñanza/aprendizaje de programación en C?
Facilidad de uso percibida (FUP)	FUP1	¿Está de acuerdo que es fácil de utilizar el software de apoyo especializado en el aprendizaje de programación en C en el laboratorio de software de la carrera de computación?
	FUP2	¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C es una herramienta que ayuda en su proceso de enseñanza/aprendizaje?
	FUP3	¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C es fácil navegar y encontrar la información que necesita?
Actitud hacia el uso (AU)	AU1	¿Se siente cómodo/a al utilizar el software de apoyo especializado en el aprendizaje de programación en C?
	AU2	¿Considera que el uso del software de apoyo especializado en el aprendizaje de programación en C es beneficioso para agilizar su proceso de enseñanza/aprendizaje?
	AU3	¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C, es una buena idea en general?
	AU4	¿Considera agradable el uso del software de apoyo especializado en el aprendizaje de programación en C?
Intención de uso (IU)	IU1	¿Tiene usted la intención de utilizar el software de apoyo especializado en el aprendizaje de programación en C?



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

IU2	¿Está de acuerdo que el uso del software de apoyo especializado en el aprendizaje de programación en C sería beneficioso en el futuro?
IU3	¿Está de acuerdo que el software de apoyo especializado en el aprendizaje de programación en C mejorará la experiencia académica en el laboratorio?

REPRESENTACIÓN GRÁFICA DE LOS RESULTADOS

Previo a la presentación de los resultados correspondientes a cada constructo, se procede a verificar la autenticidad y validez del proceso de recolección de datos. Como evidencia de la participación estudiantil, se incluye a continuación una la **Fig. 1** que muestra los correos institucionales de los estudiantes que colaboraron en la realización de la encuesta, garantizando así la legitimidad de la muestra obtenida.

Usuarios que han respondido
Correo electrónico
joseph.narvaez@unl.edu.ec
emilio.jaramillo@unl.edu.ec
juan.veintimilla@unl.edu.ec
holger.sanchez@unl.edu.ec
marco.chiliguano@unl.edu.ec
stalin.tapia@unl.edu.ec
camila.chimbo@unl.edu.ec
dilan.chamba@unl.edu.ec
...

Fig. 1. Muestra de la participación estudiantil que formaron parte de la encuesta



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Utilidad Percibida (UP):

Estos resultados se representan de manera grafica en las siguientes figuras.



Fig. 2. Pregunta UP1

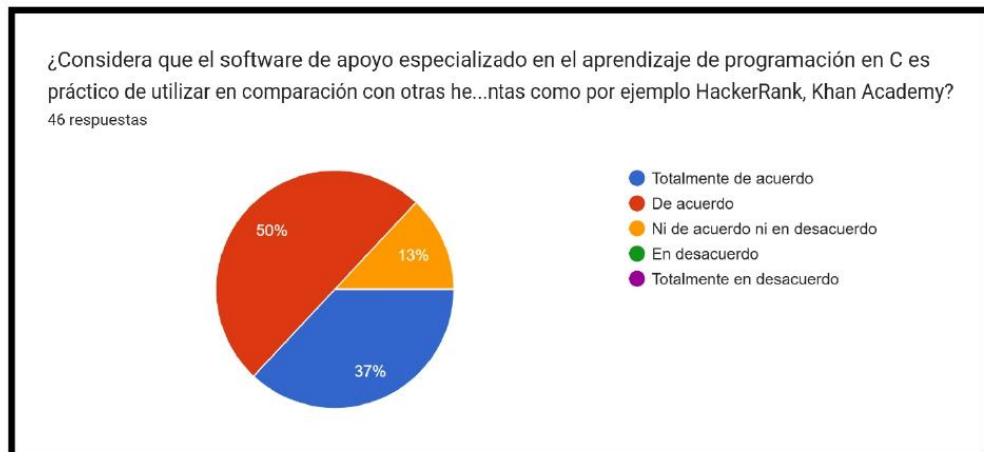


Fig. 3. Pregunta UP2

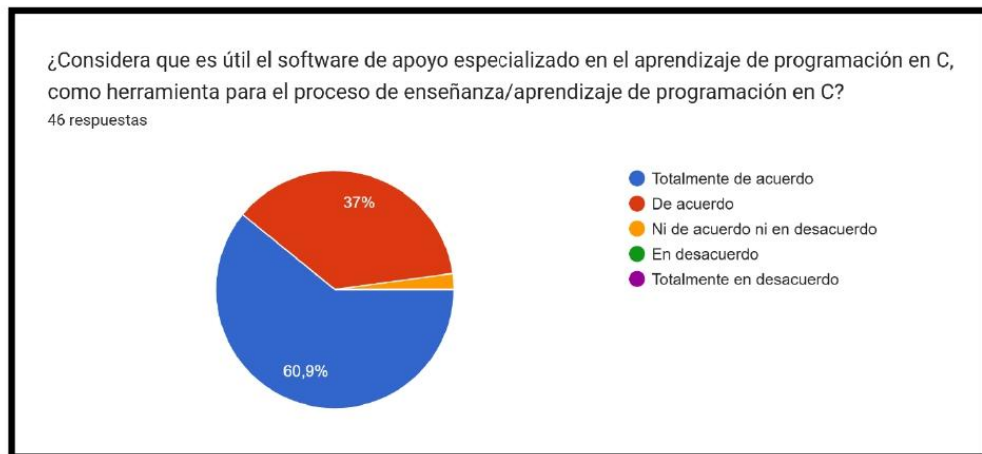


Fig. 4. Pregunta UP3

Facilidad de uso percibida (FUP)

Estos resultados se representan de manera grafica en las siguientes figuras.

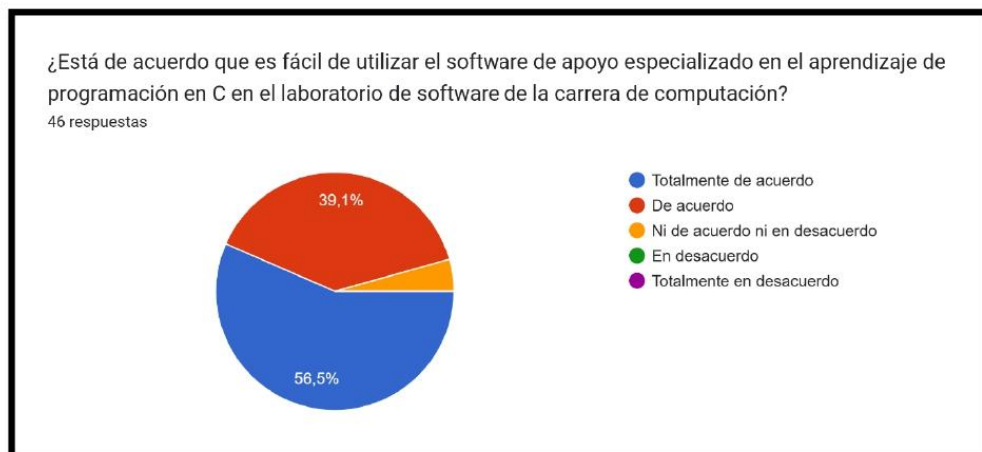


Fig. 5. Pregunta FUP1

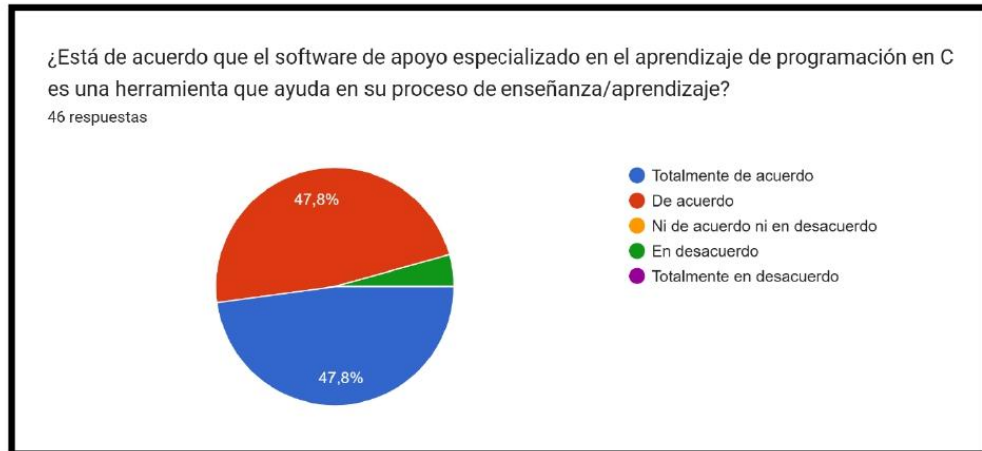


Fig. 6. Pregunta FUP2

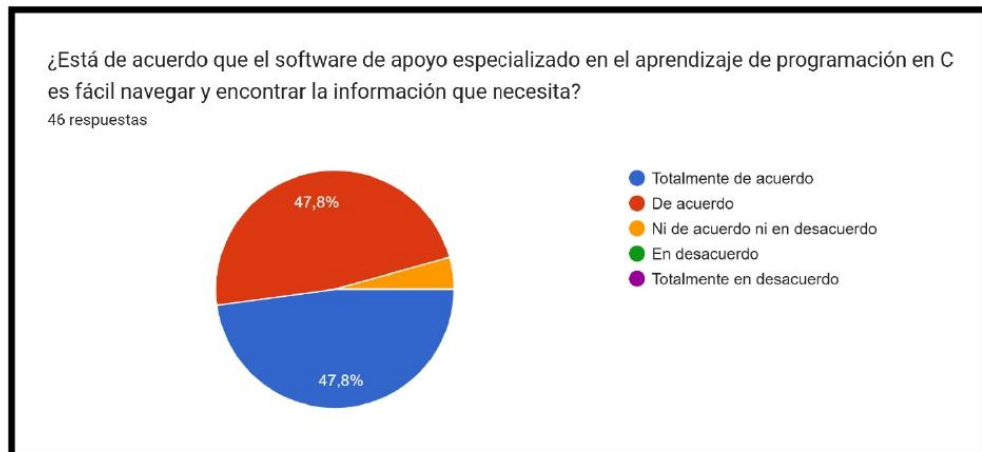


Fig. 7. Pregunta FUP3

Actitud hacia el uso (AU)



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Estos resultados se representan de manera grafica en las siguientes figuras.

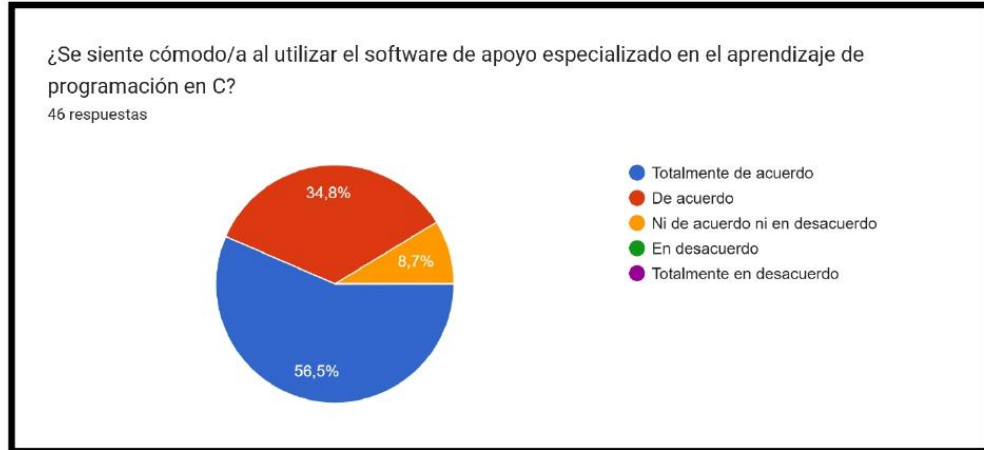


Fig. 8. Pregunta AU1

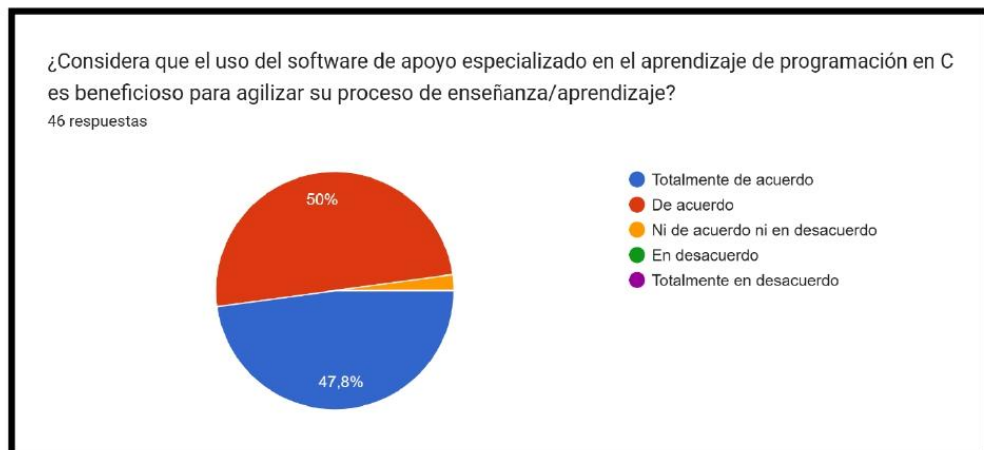


Fig. 9. Pregunta AU2

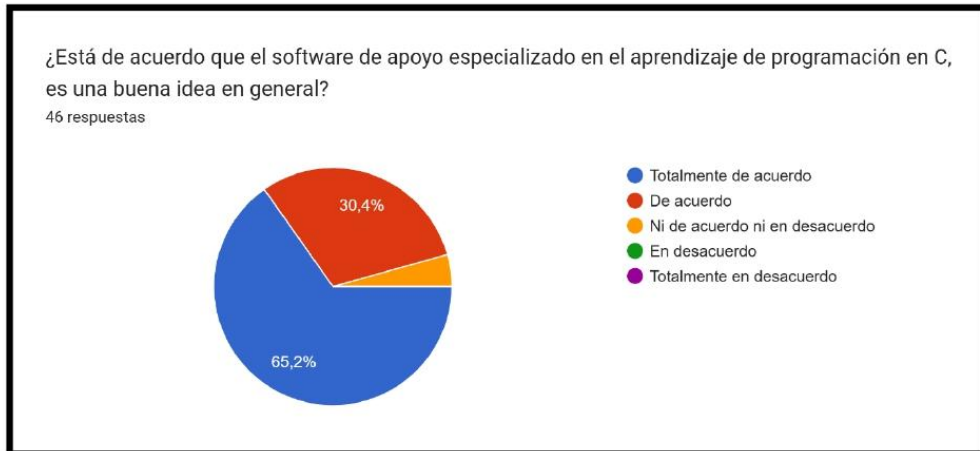


Fig. 10. Pregunta AU3

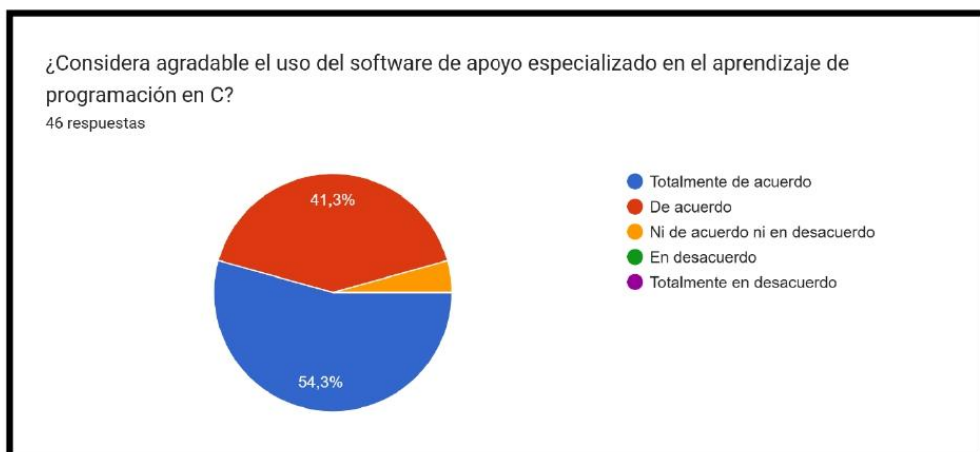


Fig. 11. Pregunta AU4

Intención de uso (IU).



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Estos resultados se representan de manera grafica en las siguientes figuras.

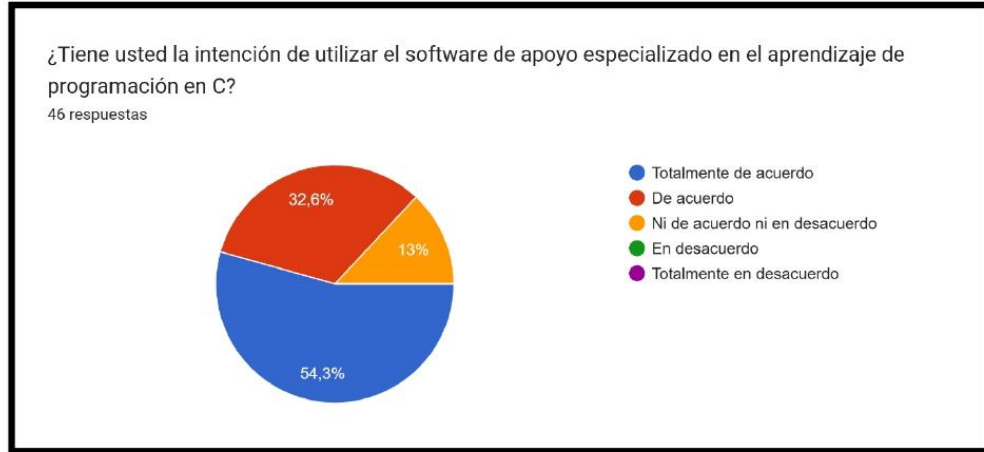


Fig. 12. Pregunta IU1

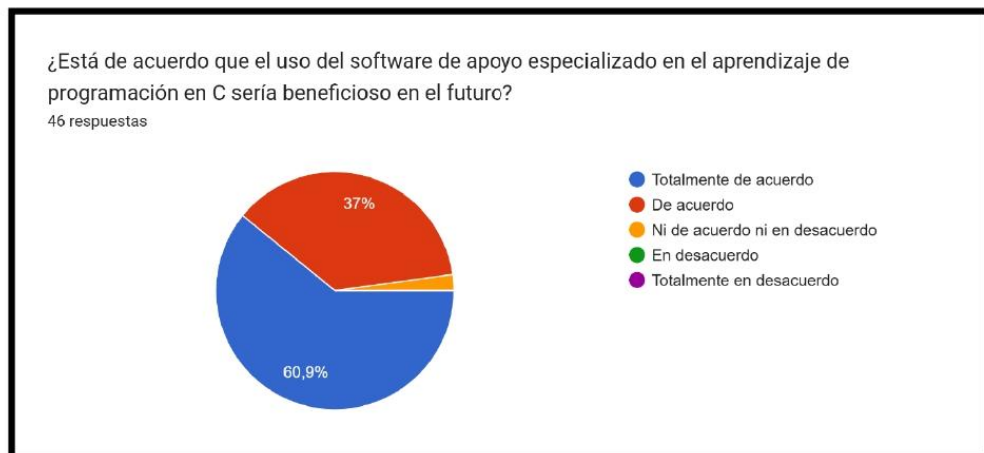


Fig. 13. Pregunta IU2



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación



Fig. 14. Pregunta IU3

<p>Ing. José Oswaldo Guamán Quinche, Mg.Sc.</p>	<p>Escaneado *lectura.combo* por: JOSE OSWALDO GUAMAN QUINCHE</p>
---	---

Anexo 8: Manual de usuario para estudiante



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Revisado y aprobado por:**
 - Ing. José Oswaldo Guamán, director del Trabajo de Integración Curricular.

Descripción: El presente manual de usuario documenta de manera exhaustiva las funciones y responsabilidades asignadas al usuario con rol de Estudiante. Este perfil tiene acceso a diversas funcionalidades del software de apoyo, las cuales han sido específicamente diseñadas para ser una herramienta de apoyo en el proceso de aprendizaje de programación en lenguaje C.



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

MANUAL DE USUARIO: ESTUDIANTE

1 Ingreso a la aplicación web

Para acceder al software de apoyo, el usuario deberá dirigirse a su navegador web e ingresar la URL correspondiente en la barra de direcciones.

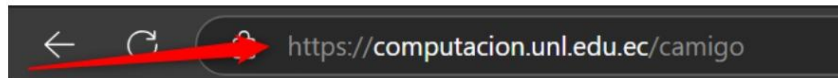


Figura 1. Accediendo al software de apoyo

Luego, se presenta la página web.

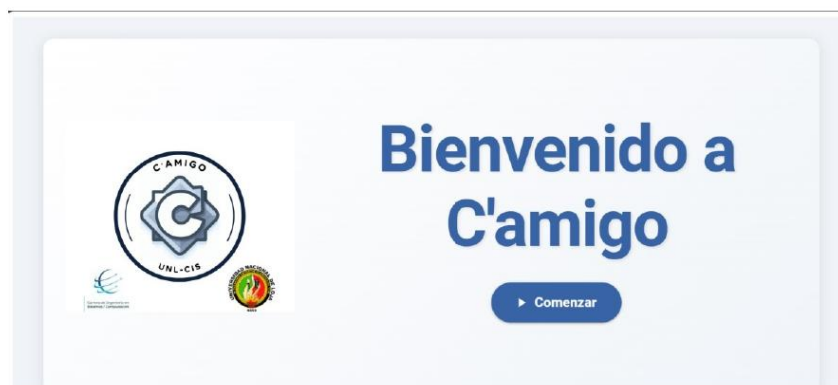


Figura 2. Página de inicio para el software de apoyo

Para hacer uso del software de apoyo se debe seleccionar la opción “Comenzar”, el cual la redirigirá a la sección de **Inicio de sesión**



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

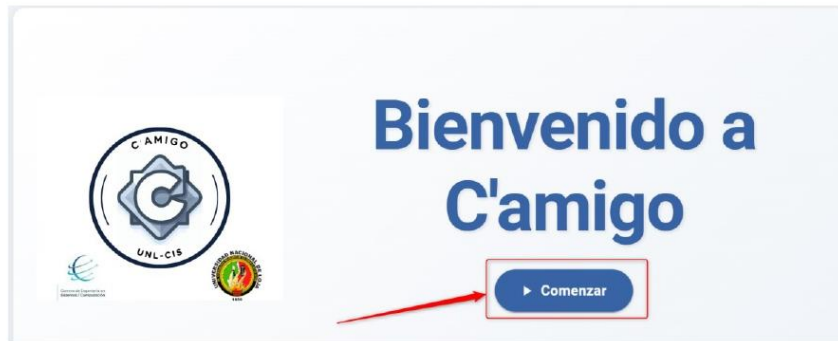


Figura 3. Página de inicio para el software de apoyo

2 Registro de estudiantes

Para acceder al formulario de registro se tiene que dirigir a la opción de “Comenzar”

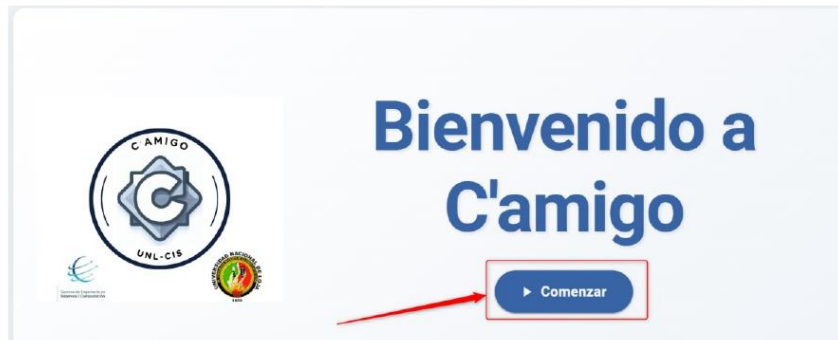


Figura 4. Página de inicio donde se va a redirigir al iniciar sesión.

A continuación, se despliega el formulario de inicio de sesión, en cuya parte inferior se encuentra disponible la opción para registrar una nueva cuenta de usuario.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación



Figura 5. Accediendo al registro de usuarios.

Luego de acceder se visualiza un formulario que contiene 4 campos obligatorios y que son indispensables para el correcto registro del estudiante.



Figura 6 Campos obligatorios del formulario de registro

Una vez ingresados los datos requeridos, el usuario deberá seleccionar el botón "Registrarse". Si la información proporcionada es correcta y cumple con los criterios establecidos, el sistema procederá a crear la cuenta y concederá al usuario el acceso a la aplicación web.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Registro

Ingrese sus dos nombres *

Test Manual

Ingrese sus dos apellidos *

Test Manual

Ingrese su correo electrónico *

manualEstudiante@unl.edu.ec

Ingrese su contraseña *

REGISTRARSE

Figura 7. Registro en el sistema

Una vez completado el registro de la cuenta, el sistema muestra un mensaje de confirmación exitosa. En dicho mensaje, se encuentra disponible un botón que permite al usuario cerrar la notificación.



Figura 8. Mensaje de éxito del registro

Después de cerrar la notificación de registro exitoso, el sistema redirige al formulario de registro inicial. En esta instancia, se despliega un mensaje recordatorio que informa al usuario que la cuenta ya ha sido creada y que puede proceder a iniciar sesión en el sistema.



Figura 9. Recordatorio del registro exitoso

3 Inicio de sesión

3.1 Usuarios externos

Para acceder al formulario de inicio sesión, se debe acceder a la pagina predeterminada de la aplicación, presentada en la sección **Ingreso a la aplicación web**



Figura 10. Formulario de Inicio de sesión

Para acceder al sistema, el usuario deberá ingresar las credenciales correspondientes a la cuenta previamente creada. Los datos obligatorios requeridos para un inicio de sesión exitoso son el correo electrónico y la contraseña.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Iniciar sesión

Ingresar su correo electrónico *

manualEstudiante@uni.edu.ec

Ingresar su contraseña *

.....

INICIAR SESIÓN

IAM COMPUTACIÓN

¿No tienes una cuenta? Regístrate

Copyright © Universidad Nacional de Loja 2024.

Figura 11. Ingreso de datos en el formulario de Inicio de Sesión.

Quando las credenciales sean validadas correctamente, el software de apoyo permitirá el acceso a la ventana de temas destacados. En esta sección, se presentan de manera visual los cinco temas mejor valorados por la comunidad de usuarios. Asimismo, se encuentra disponible el manual de usuario correspondiente al rol de estudiante.



Figura 12. Temas destacados

En la sección correspondiente a los manuales de usuario, se presenta un botón que indica el tipo de usuario. Esta funcionalidad permite al usuario descargar su manual específico cuando lo requiera.

Manuales de Usuario



Figura 13. Manuales de usuario

3.2 Usuarios internos.

El acceso también es posible utilizando el IAM de la carrera de Computación. Para esto, el usuario debe seleccionar esta opción en la pantalla de inicio de sesión.



Figura 14. Inicio de sesión

Luego ingresar las credenciales de acceso al sistema y presionar la opción de “Iniciar Sesión”



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Español v

Iniciar sesión

Usuario o email

Contraseña

[¿Has olvidado tu contraseña?](#)

Iniciar sesión

Figura 15. Inicio de sesión mediante el LAM Computación

4 Editar información de usuario

Para acceder al apartado de perfil de usuario, se debe dirigir a la parte superior derecha y seleccionar la opción "Editar", ubicada junto a "Visualizar Contenido".

Bienvenido, Test Manual
Periodo académico: Septiembre - Febrero 2025

VISUALIZAR CONTENIDO EDITAR SALIR

Temas destacados

¡Bienvenido! Aquí te presentamos los 5 temas más populares basados en los "Me gusta" recibidos. Estos temas reflejan el contenido que más ha resonado con nuestra comunidad. ¡Explora y descubre lo que otros encuentran interesante!

Tema	Número de "Me gusta"
1	5
2	4
3	3
4	2
5	1

Figura 16. Opción "Editar perfil"

En la opción de editar perfil, se despliega un modal que muestra los datos modificables del usuario: nombres, apellidos, correo electrónico y contraseña. Para efectuar cambios en estos campos, se requiere ingresar la contraseña actual y seleccionar "Validar contraseña".



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Editar Usuario

Nombres *

Apellidos *

Email *

Contraseña Actual *

VALIDAR CONTRASEÑA ACTUAL

CERRAR GUARDAR CAMBIOS

Figura 17. Información del usuario a editar

Una vez que el usuario valide su contraseña actual, el sistema le permitirá, si así lo desea, modificar su contraseña de inicio de sesión.

Editar Usuario

manualEstudiante@unl.edu.ec

Contraseña Actual *

VALIDAR CONTRASEÑA ACTUAL

Nueva Contraseña (opcional)

Confirmar Nueva Contraseña

CERRAR **GUARDAR CAMBIOS**

Figura 18. Cambio de contraseña en el editar perfil



5 Visualizar contenido

En la parte superior derecha se encuentra la opción "Visualizar contenido". Al seleccionarla, se muestra al usuario todo el contenido que C'amigo ofrece a los estudiantes, incluyendo temas, subtemas, ejercicios y preguntas de control.



Figura 19. Opción "Visualizar contenido"

Al seleccionar "Visualizar contenido", se presenta una ventana inicial. En la sección izquierda aparece el logo de C'amigo, su slogan y el progreso del usuario en la aplicación. Se muestra también el listado de temas disponibles con su respectivo progreso. En la parte derecha se despliega un desglose del contenido de la página web.

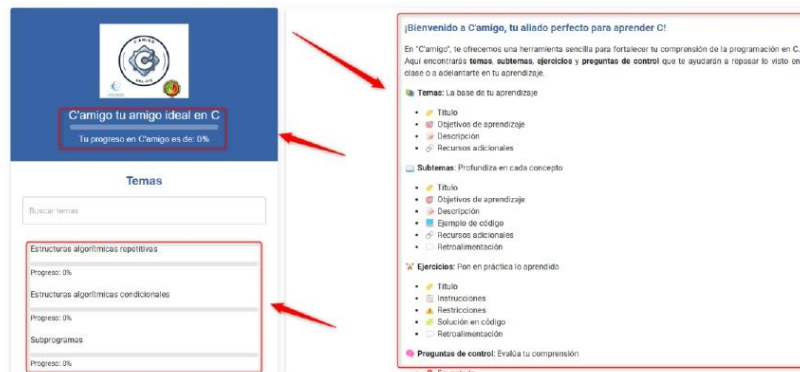


Figura 20. Visualizar contenido



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

En la sección izquierda de la página se aprecia el logo de C'amigo en la parte superior, seguido de su eslogan. Debajo se muestra el progreso del estudiante dentro del software. Posteriormente, se ubica un buscador que permite localizar temas específicos. La parte inferior de esta sección presenta un listado de todos los temas disponibles en la plataforma.

The screenshot displays the C'amigo software interface. At the top, there is a blue header with the C'amigo logo and the text "C'amigo tu amigo ideal en C" and "Tu progreso en C'amigo es de: 0%". Below the header, there is a section titled "Temas" with a search bar labeled "Buscar temas". A list of topics is shown, each with a progress bar indicating 0% completion. The topics listed are: "Estructuras algorítmicas repetitivas", "Estructuras algorítmicas condicionales", and "Subprogramas". Red arrows point to the C'amigo logo, the search bar, and the first topic in the list.

Figura 21. Progreso y listado de contenido

Al seleccionar un tema del listado, la ventana se divide en dos secciones. La parte derecha muestra el contenido del tema, incluyendo título, objetivos de aprendizaje, descripción y recursos adicionales. En la parte izquierda se listan los subtemas correspondientes por título, añadiendo una opción para retornar al contenido anterior. Este patrón se repite en niveles subsiguientes. Al elegir un subtema, la izquierda lista sus ejercicios, mientras la derecha presenta el contenido del subtema: título, objetivos,

descripción, ejemplo de código, recursos adicionales y retroalimentación. La selección de un ejercicio muestra sus preguntas de control a la izquierda, y a la derecha el contenido del ejercicio: título, instrucciones, restricciones, solución en código y retroalimentación. Finalmente, al escoger una pregunta de control, la izquierda mantiene la lista de preguntas, y la derecha exhibe el enunciado, cuatro opciones y justificación. Es importante destacar que en cada nivel existe la posibilidad de retornar al contenido superior, manteniendo la coherencia en la presentación de la información relacionada.



Estructuras algorítmicas repetitivas

Objetivos

- Comprender el concepto de estructuras repetitivas (bucles). Al finalizar, el estudiante será capaz de entender cómo funcionan los ciclos en programación, específicamente en C, para repetir acciones hasta cumplir una condición.
- Diferenciar entre diferentes tipos de bucles. El estudiante aprenderá a identificar y utilizar los tres tipos principales de bucles en C: for, while y do-while.
- Aplicar estructuras repetitivas para resolver problemas. El estudiante podrá implementar bucles en la resolución de problemas que requieran la ejecución repetitiva de código, como cálculos iterativos, recorrido de arreglos y procesamiento de datos.
- Optimizar el uso de bucles para mejorar la eficiencia del código. El estudiante será capaz de analizar cuando usar un tipo de bucle sobre otro y como estructurarlos correctamente para escribir código más eficiente y legible.

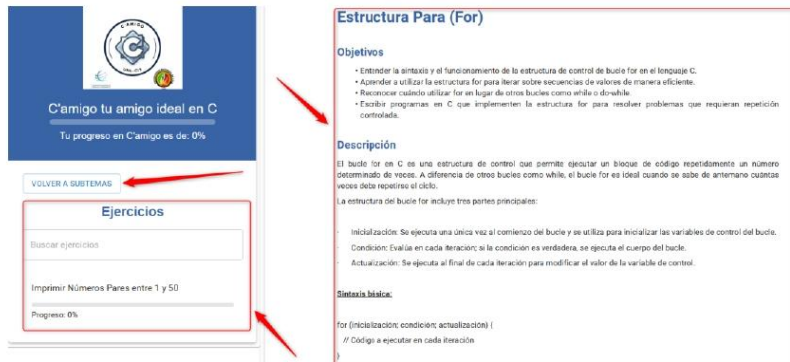
Descripción

Las estructuras algorítmicas repetitivas, también conocidas como bucles o ciclos, son componentes fundamentales en la programación que permiten ejecutar un bloque de código de manera repetida mientras se cumple una condición específica. Estas estructuras son útiles cuando se requiere realizar una acción varias veces sin necesidad de escribir el mismo código en múltiples ocasiones. Los bucles pueden depender de un contador que varía en cada iteración (como en un ciclo for) o de una condición lógica que se evalúa en cada repetición (como en los ciclos while y do-while). Estas estructuras facilitan la automatización de tareas repetitivas, optimizando el código y mejorando su eficiencia.

Existen varios tipos de estructuras repetitivas, cada una con un comportamiento particular. El ciclo for es ideal cuando el número de iteraciones es conocido de antemano, mientras que el ciclo while se utiliza cuando la cantidad de repeticiones depende de una condición lógica que puede cambiar durante la ejecución. El ciclo do-while garantiza que el bloque de código se ejecute al menos una vez antes de verificar la condición. Estas estructuras son esenciales para desarrollar algoritmos que manejen grandes volúmenes de datos, procesen listas o realicen cálculos matemáticos de manera eficiente.

Recursos

Figura 22. Visualizar temas



Estructura Para (For)

Objetivos

- Entender la sintaxis y el funcionamiento de la estructura de control de bucle for en el lenguaje C.
- Aprender a utilizar la estructura for para iterar sobre secuencias de valores de manera eficiente.
- Reconocer cuándo utilizar for en lugar de otros bucles como while o do-while.
- Escribir programas en C que implementen la estructura for para resolver problemas que requieran repetición controlada.

Descripción

El bucle for en C es una estructura de control que permite ejecutar un bloque de código repetidamente un número determinado de veces. A diferencia de otros bucles como while, el bucle for es ideal cuando se sabe de antemano cuántas veces debe repetirse el ciclo.

La estructura del bucle for incluye tres partes principales:

- Inicialización:** Se ejecuta una única vez al comienzo del bucle y se utiliza para inicializar las variables de control del bucle.
- Condición:** Evalúa en cada iteración si la condición es verdadera, se ejecuta el cuerpo del bucle.
- Actualización:** Se ejecuta al final de cada iteración para modificar el valor de la variable de control.

Sintaxis básica:

```
for (Inicialización; condición; actualización) {
    // Código a ejecutar en cada iteración
}
```

Figura 23. Visualizar subtemas



C'amigo tu amigo ideal en C
Tu progreso en C'amigo es de: 0%

[VOLVER A EJERCICIOS](#)

Preguntas

Buscar preguntas

En el programa que imprime los números pares entre 1 y 50, ¿qué condición se utiliza para determinar si un número es par?

Progreso: 0%

¿Cuál es el rango de números que imprime el programa que utiliza un bucle for para mostrar los

Imprimir Números Pares entre 1 y 50

Instrucciones

Escribe un programa en C que utilice un bucle for para imprimir todos los números pares entre 1 y 50 inclusive. El programa debe utilizar la estructura de control for y verificar que los números impresos sean pares.

Restricciones

1. Debes usar la estructura de control for para resolver el problema.
2. El rango de números a imprimir es de 1 a 50.
3. Solo se debe imprimir números pares.

Realiza tu solución:

Programiz
C Online Compiler

[Programiz PRO >](#)

main.c [Run](#) [Output](#) [Clear](#)

```

1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main() {
5     // Write C code here
6     printf("try programiz.pro");
7
8     return 0;

```

Figura 24. Visualizar ejercicios

C'amigo tu amigo ideal en C
Tu progreso en C'amigo es de: 0%

[VOLVER A EJERCICIOS](#)

Preguntas

Buscar preguntas

En el programa que imprime los números pares entre 1 y 50, ¿qué condición se utiliza para determinar si un número es par?

Progreso: 0%

¿Cuál es el rango de números que imprime el programa que utiliza un bucle for para mostrar los

En el programa que imprime los números pares entre 1 y 50, ¿qué condición se utiliza para determinar si un número es par?

Selecciona la respuesta correcta:

- 1 % 2 != 0
- 1 % 2 == 0
- 1 / 2 == 1
- 1 > 2

Figura 25. Visualizar preguntas de control

6 Realizar ejercicios

Al seleccionar un ejercicio de un subtema, el usuario visualiza el contenido completo del mismo, que incluye título, instrucciones y restricciones. Adicionalmente, el software de apoyo integra un editor de código en línea, específicamente el editor de Programiz. Esta



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

funcionalidad permite al estudiante realizar el ejercicio directamente en la plataforma, sin necesidad de abandonar la página web, proporcionando así una experiencia de aprendizaje fluida y conveniente.

Imprimir Números Pares entre 1 y 50

Instrucciones

Escribe un programa en C que utilice un bucle for para imprimir todos los números pares entre 1 y 50 inclusive. El programa debe utilizar la estructura de control for y verificar que los números impresos sean pares.

Restricciones

1. Debes usar la estructura de control for para resolver el problema.
2. El rango de números a imprimir es de 1 a 50.
3. Solo se deben imprimir números pares

Realiza tu solución:

Programiz
C Online
Compiler

Programiz
PRO >

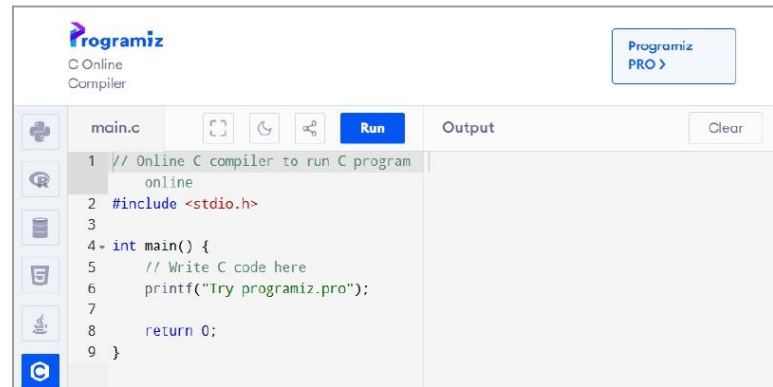
main.c Run Output Clear

```
1 // Online C compiler to run C program
  online
2 #include <stdio.h>
3
4 int main() {
5     // Write C code here
6     printf("Try programiz.pro");
7
8     return 0;
9 }
```

Figura 26. Editor online Programiz

El sistema ofrece asistencia adicional al estudiante que enfrenta dificultades en la resolución del ejercicio. Cuando el usuario selecciona la opción "Mostrar solución", el sistema revela una solución proporcionada por el docente. Esta característica permite al estudiante acceder a una respuesta guía cuando lo considere necesario, facilitando así el proceso de aprendizaje y la comprensión del ejercicio en cuestión.

Realiza tu solución:



Programiz
C Online
Compiler

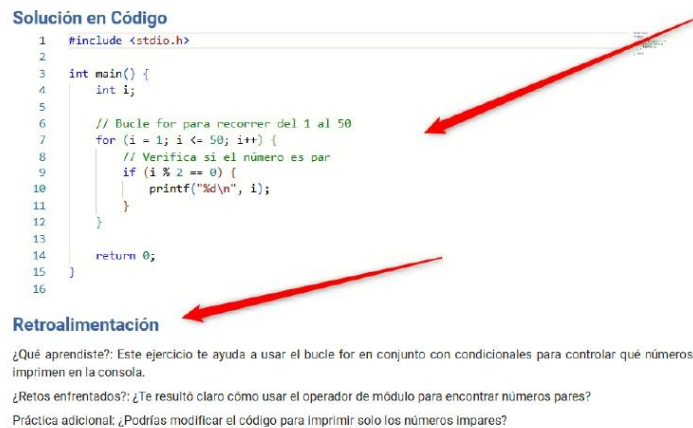
main.c Run Output Clear

```
1 // Online C compiler to run C program  
  online  
2 #include <stdio.h>  
3  
4 int main() {  
5     // Write C code here  
6     printf("Try programiz.pro");  
7  
8     return 0;  
9 }
```

MOSTRAR SOLUCIÓN

Figura 27. Mostrar solución

Al seleccionar la opción "Mostrar solución", el sistema despliega un editor de código integrado en el software de apoyo. Simultáneamente, se presenta una retroalimentación relacionada con el ejercicio. Esta funcionalidad permite al usuario visualizar la solución propuesta y recibir información adicional, enriqueciendo así su comprensión del problema y su proceso de aprendizaje.



Solución en Código

```
1 #include <stdio.h>  
2  
3 int main() {  
4     int i;  
5  
6     // Bucle for para recorrer del 1 al 50  
7     for (i = 1; i <= 50; i++) {  
8         // Verifica si el número es par  
9         if (i % 2 == 0) {  
10            printf("%d\n", i);  
11        }  
12    }  
13  
14    return 0;  
15 }  
16
```

Retroalimentación

¿Qué aprendiste?: Este ejercicio te ayuda a usar el bucle for en conjunto con condicionales para controlar qué números se imprimen en la consola.

¿Retos enfrentados?: ¿Te resultó claro cómo usar el operador de módulo para encontrar números pares?

Práctica adicional: ¿Podrías modificar el código para imprimir solo los números impares?

Figura 28. Solución en código y retroalimentación del ejercicio



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Al realizar un ejercicio, el estudiante tiene la opción de añadir comentarios. Estos pueden ser generales, abarcando todo el contenido, o específicos, enfocándose en aspectos particulares. Esta funcionalidad da paso a una nueva sección, permitiendo una interacción más detallada y personalizada con el material de estudio.

7 Comentarios

7.1 Crear comentario

El estudiante tiene la facultad de realizar comentarios pertinentes a la totalidad del contenido, incluyendo temas, subtemas, ejercicios o preguntas de control. Esta funcionalidad tiene como propósito fomentar la comunicación entre todos los usuarios de la plataforma. Para generar un nuevo comentario, el usuario deberá dirigirse al final de la sección donde se exhibe el contenido del ejercicio correspondiente. En dicha ubicación, encontrará un cuadro de texto destinado a la redacción del comentario. Una vez finalizada la redacción, el estudiante deberá seleccionar la opción "Agregar comentario" para publicar su aportación.

Comentarios

¿Tienes dudas Test Manual?, coméntalas:

Ingrese un nuevo comentario Test Manual

AGREGAR COMENTARIO

Figura 29. Agregar comentario

7.2 Editar comentario

Una vez que se ha añadido un comentario al ejercicio, se desplegará una lista con todos los comentarios realizados por los usuarios. Esta lista se ubicará debajo del recuadro destinado al ingreso de nuevos comentarios. El usuario tendrá la posibilidad de modificar sus propios comentarios mediante la selección de la opción "Editar" asociada a cada uno de ellos.



Figura 30. Editar comentario

Al seleccionar la opción "Editar", el sistema automáticamente cargará el contenido del comentario original en el recuadro de edición. Esta acción permitirá al usuario modificar el texto según lo considere necesario. Una vez realizados los cambios deseados, el usuario podrá confirmar la edición presionando la opción "Editar comentario". Este proceso facilita la actualización y refinamiento de las contribuciones previamente realizadas por el estudiante.

Comentarios

¿Tienes dudas Test Manual?, coméntalas:

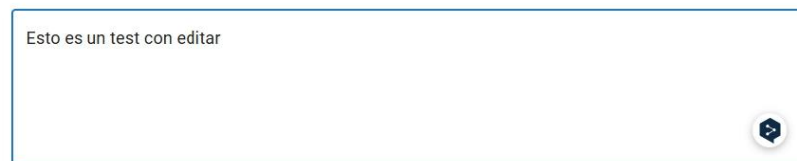
A screenshot of a comment box. The box contains the text "Esto es un test con editar". In the bottom right corner of the box, there is a small circular icon with a pencil, indicating an edit function.

Figura 31. Comentario editado

Al momento de listar el comentario se puede observar de forma inmediata el comentario ya editado en la lista de comentarios de ese contenido.



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Test Manual



Esto es un test con editar

Estudiante Cinco

Excelente contenido

Figura 32. Listado de comentarios

8 Responder preguntas de control

En la sección derecha de la ventana **Visualizar contenido**, el usuario puede revisar las preguntas de control relacionadas con el material previamente estudiado. Para responder, el estudiante selecciona una de las cuatro opciones disponibles y recibe retroalimentación inmediata sobre la respuesta correcta. Este proceso permite al usuario evaluar su aprendizaje y aumentar su progreso en el software de apoyo.

En el programa que imprime los números pares entre 1 y 50, ¿qué condición se utiliza para determinar si un número es par?

Selecciona la respuesta correcta:

- $i \% 2 != 0$
- $i \% 2 == 0$
- $i / 2 == 1$
- $i > 2$



Figura 33. Pregunta de control

Al seleccionar la opción correcta, el sistema desplegará un mensaje de éxito para el estudiante. Simultáneamente, se incrementará el progreso general del contenido. El sistema mostrará la respuesta correcta junto con la justificación correspondiente a la pregunta, proporcionando así una retroalimentación completa y educativa.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Bienvenido, Test Manual
Período académico: Septiembre - Febrero 2026

¡Felicitaciones! Has completado esta pregunta con éxito. Puedes revisarla para reforzar tus conocimientos.

En el programa que imprime los números pares entre 1 y 50, ¿qué condición se utiliza para determinar si un número es par?

Respuesta seleccionada
% 2 == 0

Justificación
La condición % 2 == 0 verifica si un número es divisible por 2 sin dejar un residuo, lo que significa que el número es par. Por lo tanto, la opción B es la correcta.

VOLVER A EJERCICIOS

Figura 34. Pregunta respondida con éxito.

9 Progreso del usuario

El progreso del usuario se actualiza del contenido menor al mayor ósea que debemos completar todas las preguntas de control para que el usuario pueda aumentar su progreso en C'amigo.

C'amigo tu amigo ideal en C
Tu progreso en C'amigo es de: 0%

VOLVER A EJERCICIOS

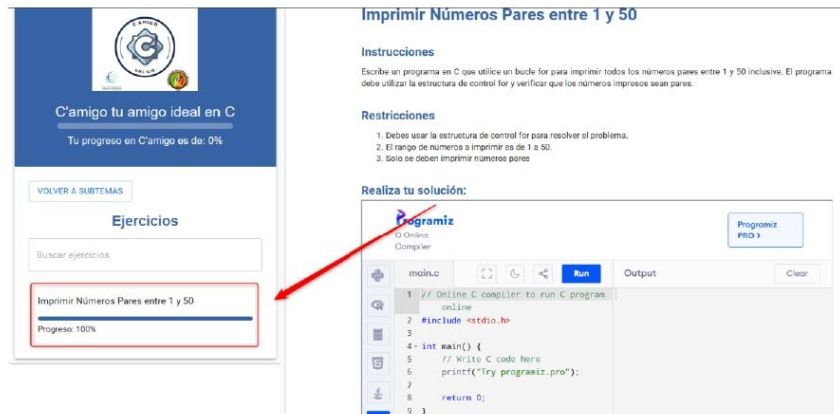
Preguntas

Buscar preguntas

En el programa que imprime los números pares entre 1 y 50, ¿qué condición se utiliza para determinar si un número es par?

Progreso: 100%

Figura 35. Progreso de la pregunta de control actualizado.



Imprimir Números Pares entre 1 y 50

Instrucciones
Escribe un programa en C que utilice un bucle for para imprimir todos los números pares entre 1 y 50 inclusive. El programa debe utilizar la estructura de control for y verificar que los números impresos sean pares.

Restricciones

1. Debes usar la estructura de control for para resolver el problema.
2. El rango de números a imprimir es de 1 a 50.
3. Solo se deben imprimir números pares.

Realiza tu solución:

Programiz
Online Compiler

```
main.c
1 // Online C compiler to run C program
2 online
3 #include <stdio.h>
4 int main() {
5 // Write C code here
6 printf("try programiz.pro");
7
8 return 0;
9 }
```

Run Output Clear

Figura 36. Progreso del ejercicio actualizado



C'amigo tu amigo ideal en C
Tu progreso en C'amigo es de: 0%

VOLVER A TEMAS

Subtemas

Buscar subtemas

Estructura Para (For)
Progreso: 100%

Estructura Mientras (While) en C
Progreso: 0%

Estructura Hacer...Mientras (Do... While) en C
Progreso: 0%

Descripción
El bucle for en C es una estructura de control que permite ejecutar un bloque de código repetidamente un número determinado de veces. A diferencia de otros bucles como while o do-while.
Escribir programas en C que implementen la estructura for para resolver problemas que requieran repetición controlada.

La estructura de bucle for posee tres partes principales:

- Inicialización: Se ejecuta una única vez al comienzo del bucle y se utiliza para inicializar las variables de control del bucle.
- Condición: Evalúa en cada iteración; si la condición es verdadera, se ejecuta el cuerpo del bucle.
- Actualización: Se ejecuta al final de cada iteración para modificar el valor de la variable de control.

Sintaxis básica:

```
for (inicialización; condición; actualización) {
// Código a ejecutar en cada iteración
}
```

Ejemplo de Código

```
1 #include <stdio.h>
2
3 int main() {
4     int i;
```

Figura 37. Progreso del subtema actualizado



Figura 38. Progreso del tema actualizado

El progreso general del usuario se calcula en función de la cantidad de temas completados. En este manual de usuario, se procederá a completar el tema titulado "Estructuras algorítmicas repetitivas". Este proceso permitirá visualizar de manera precisa el avance del usuario en el sistema.



Figura 39. Progreso del usuario actualizado.

10 Volver a Temas destacados.

Para regresar a la sección principal tras iniciar sesión, el usuario deberá seleccionar la opción ubicada en el extremo izquierdo, donde se muestra el logotipo de C'amigo. Al seleccionar dicha opción, el sistema redirigirá al usuario a la vista de "Temas destacados".




Figura 40. Temas destacados

11 Cerrar sesión

Para cerrar sesión en el software de apoyo se debe dirigir a la parte superior derecha y seleccionar la opción "Salir".



Figura 41. Cerrar sesión

Ing. José Oswaldo Guamán Quinche, Mg.Sc.	 <p>Firmado digitalmente por: JOSE OSWALDO GUAMAN QUINCHE</p>
--	---

Anexo 9: Manual de usuario para docente



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Revisado y aprobado por:**
 - Ing. José Oswaldo Guamán, director del Trabajo de Integración Curricular.

Descripción: El presente manual de usuario documenta de manera exhaustiva las funciones y responsabilidades asignadas al usuario con rol Docente. Este perfil tiene acceso a diversas funcionalidades del software de apoyo, las cuales han sido específicamente diseñadas para ser una herramienta de apoyo en el proceso de aprendizaje de programación en lenguaje C.



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

MANUAL DE USUARIO: ADMINISTRADOR

1 Ingreso a la aplicación web

Para acceder al software de apoyo, el usuario deberá dirigirse a su navegador web e ingresar la URL correspondiente en la barra de direcciones.

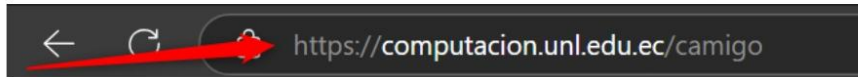


Figura 1. Accediendo al software de apoyo

Luego, se presenta la página web.



Figura 2. Página de inicio para el software de apoyo

Para acceder al software de apoyo, seleccione la opción "Comenzar". Esta acción le redirigirá a la sección de **Inicio de sesión**. Sin embargo, para obtener una cuenta con rol docente, el usuario deberá primero enviar una solicitud formal mediante correo electrónico a la dirección camigounlcis@gmail.com. La cual es enviada a los administradores del sistema para la correcta creación de la cuenta con rol docente.

2 Inicio de sesión

2.1 Usuarios externos

Para acceder al formulario de inicio sesión, se debe acceder a la página predeterminada de la aplicación, presentada en la sección **Ingreso a la aplicación web**

Para acceder al sistema, el usuario deberá ingresar las credenciales correspondientes a la cuenta previamente creada. Los datos obligatorios requeridos para un inicio de sesión exitoso son el correo electrónico y la contraseña.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Figura 3. Ingreso de datos en el formulario de Inicio de Sesión.

Una vez que las credenciales del docente sean verificadas correctamente, el sistema concederá acceso a la interfaz de temas destacados. En esta sección se exhiben visualmente los cinco temas que han recibido la mejor valoración por parte de la comunidad de usuarios.



Figura 4. Opción de administración

Al seleccionar la opción "Gestionar contenido", se presenta una interfaz que proporciona al usuario un conjunto de funcionalidades. Estas incluyen la capacidad de registrar nuevo contenido, como temas, subtemas, ejercicios y preguntas de control, de igual forma el editar dicho contenido, activar o desactivar este contenido y revisar el historial de cambios de cada contenido, todo eso será detallado a lo largo de este manual.



Figura 5. Página de Gestionar Contenido

2.2 Usuarios internos.

El acceso también es posible utilizando el IAM de la carrera de Computación. Para esto, el usuario debe seleccionar esta opción en la pantalla de inicio de sesión.



Figura 6. Inicio de sesión

Luego ingresar las credenciales de acceso al sistema y presionar la opción de “Iniciar Sesión”



Iniciar sesión

Usuario o email

Contraseña

[¿Has olvidado tu contraseña?](#)

Iniciar sesión

Figura 7. Inicio de sesión mediante el IAM Computación

3 Editar información de usuario

Para acceder al apartado de perfil de usuario, se debe dirigir a la parte superior derecha y seleccionar la opción "Editar", ubicada junto a "Administración".

Bienvenido, Docente Camigo

Periodo académico: 2026-02-17 - 2026-07-17

GESTIONAR CONTENIDO EDITAR SALIR

Temas destacados

¡Bienvenido! Aquí te presentamos los 5 temas más populares basados en los "Me gusta" recibidos. Estos temas reflejan el contenido que más ha resonado con nuestra comunidad. ¡Explora y descubre lo que otros encuentran interesante!

Tema	Número de "Me gusta"
1*	2.0
2*	2.0
3*	2.0
4*	1.0
5*	1.0

Figura 8. Opción "Editar perfil"

En la opción de editar perfil, se despliega un modal que muestra los datos modificables del usuario: nombres, apellidos, correo electrónico y contraseña. Para efectuar cambios en estos campos, se requiere ingresar la contraseña actual y seleccionar "Validar contraseña".



Editar Usuario

Nombres *
Test Manual

Apellidos *
Test Manual

Email *
manualEstudiante@unl.edu.ec

Contraseña Actual *
.....

VALIDAR CONTRASEÑA ACTUAL

CERRAR GUARDAR CAMBIOS

Figura 9. Información del usuario a editar

Una vez que el usuario valide su contraseña actual, el sistema le permitirá, si así lo desea, modificar su contraseña de inicio de sesión.

Editar Usuario

manualEstudiante@unl.edu.ec

Contraseña Actual *
.....

VALIDAR CONTRASEÑA ACTUAL

Nueva Contraseña (opcional)

Confirmar Nueva Contraseña

CERRAR **GUARDAR CAMBIOS**

Figura 10. Cambio de contraseña en el editar perfil



4 Seleccionar contenido

El software de apoyo presenta una estructura jerárquica constituida por cuatro niveles fundamentales: temas, subtemas, ejercicios y preguntas de control. Esta organización integral incorpora funcionalidades de gestión comunes para todos los elementos, las cuales comprenden la creación, edición, control de estado mediante activación o desactivación, así como el registro histórico de modificaciones.

La disposición jerárquica implementada permite que cada tema contenga diversos subtemas, los cuales a su vez pueden incorporar múltiples ejercicios, culminando en la inclusión de varias preguntas de control por ejercicio. En cuanto a la interacción con el sistema, la interfaz de gestión de contenido se caracteriza por su naturaleza dinámica y secuencial. El proceso se inicia al acceder a "Gestionar contenido", donde se visualiza la tabla principal de temas. La selección de un tema específico despliega automáticamente la tabla de subtemas asociados en el panel derecho. Subsecuentemente, al elegir un subtema, se genera la visualización de ejercicios correspondientes en el panel inferior izquierdo, y la selección final de un ejercicio resulta en la presentación de la tabla de preguntas de control en el panel derecho.

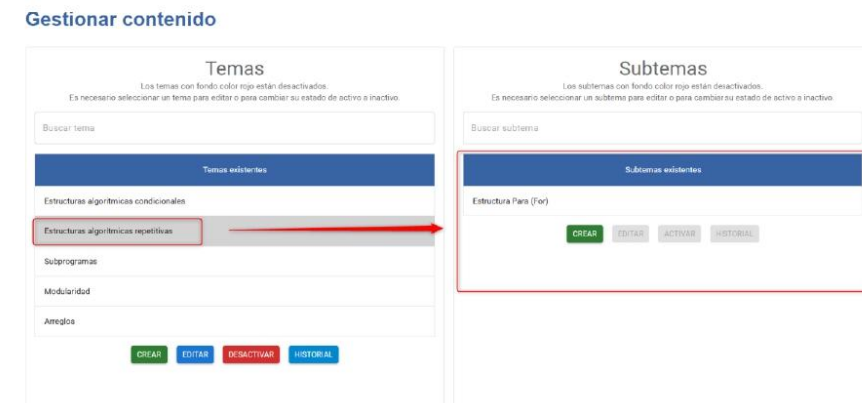


Figura 11. Selección de un tema y despliegue de sus subtemas

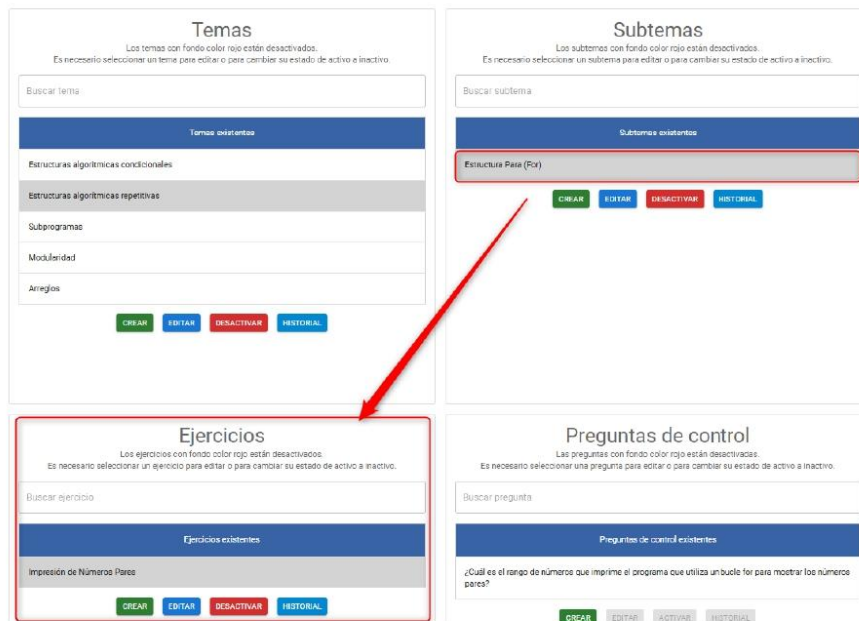


Figura 12. Selección de un subtema y despliegue de sus ejercicios.



Figura 13. Selección de un ejercicio y despliegue de sus preguntas de control

5 Gestionar contenido

El sistema faculta al personal docente para generar todo el contenido necesario en el software de apoyo mediante una estructura organizada. Este proceso de creación se efectúa a través de formularios con capacidades de texto enriquecido, permitiendo la elaboración de diversos elementos interrelacionados. La estructura fundamental comprende temas principales, donde cada uno puede contener múltiples subtemas. A su



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

vez, estos subtemas pueden albergar diversos ejercicios, y cada ejercicio incorpora una o más preguntas de control para la evaluación del aprendizaje.

Para ilustrar el procedimiento de creación de contenido, tomaremos como referencia el proceso de gestión de temas, el cual sirve como modelo representativo para las demás operaciones del sistema en subtemas, ejercicios y preguntas de control.

5.1 Crear contenido

El proceso de creación de contenido está disponible para el docente mediante una opción situada en la sección inferior de cada tabla de visualización. Esta funcionalidad se mantiene consistente a través de todos los niveles de contenido del software de apoyo, incluyendo la gestión de temas, subtemas, ejercicios y preguntas de control. La ubicación estratégica de esta opción, bajo cada tabla de listado correspondiente, facilita una interacción intuitiva con el sistema y mantiene una coherencia en la interfaz de usuario.

Al aplastar la opción de “Crear”, el sistema le presenta al docente un modal donde el usuario debe ingresar todos los datos de forma obligatoria correspondientes a cada contenido.



Figura 14. Opción crear tema.



Crea un nuevo tema

Título (campo obligatorio):

Objetivos de aprendizaje (campo obligatorio):

Descripción (campo obligatorio):

Recursos adicionales (campo obligatorio):

Previsualizar

CANCELAR GUARDAR

Figura 15. Modal crear tema

En el proceso de creación de temas, el docente debe proporcionar información específica que incluye el título, los objetivos de aprendizaje, una descripción detallada y recursos adicionales complementarios. La interfaz presenta una disposición dual: en el panel izquierdo se encuentra el área de ingreso de contenido, mientras que el panel derecho ofrece una vista previa que replica la presentación que posteriormente visualizarán los estudiantes. Esta configuración permite al docente verificar en tiempo real cómo se mostrará el contenido en la interfaz del estudiante.

Crea un nuevo tema

Título (campo obligatorio):

Arreglos

Objetivos de aprendizaje (campo obligatorio):

1. Comprender el concepto de arreglos: Al finalizar, el estudiante será capaz de entender cómo funcionan los arreglos, específicamente en C.
2. Aplicar arreglos para resolver problemas: El estudiante podrá implementar arreglos en la resolución de problemas que requieran arreglos de código, como cálculos iterativos, recorrido de arreglos y procesamiento de datos.
3. Optimizar el uso de arreglos para mejorar la eficiencia del código: El estudiante será capaz de analizar cuándo usar un tipo de arreglo sobre otro y cómo estructurarlos correctamente para escribir código más eficiente y legible.

Previsualizar

Arreglos

1. Comprender el concepto de arreglos: Al finalizar, el estudiante será capaz de entender cómo funcionan los arreglos, específicamente en C.
2. Aplicar arreglos para resolver problemas: El estudiante podrá implementar arreglos en la resolución de problemas que requieran arreglos de código, como cálculos iterativos, recorrido de arreglos y procesamiento de datos.
3. Optimizar el uso de arreglos para mejorar la eficiencia del código: El estudiante será capaz de analizar cuándo usar un tipo de arreglo sobre otro y cómo estructurarlos correctamente para escribir código más eficiente y legible.

CANCELAR GUARDAR

Figura 16. Tema creado

Una vez completado el ingreso de la información, el proceso de creación culmina cuando el docente selecciona la opción "Guardar". Al ejecutar esta acción, el sistema cierra automáticamente la ventana modal, genera una notificación confirmando el éxito de la operación y actualiza la tabla correspondiente, donde se visualiza inmediatamente el nuevo contenido incorporado.



Temas

Los temas con fondo color rojo están desactivados.

Es necesario seleccionar un tema para editar o para cambiar su estado de activo a inactivo.

Temas existentes
Estructuras algorítmicas condicionales
Estructuras algorítmicas repetitivas
Subprogramas
Modularidad
Arreglos

CREAR EDITAR ACTIVAR HISTORIAL

Figura 17. Tabla de temas

5.2 Editar contenido

El sistema proporciona funcionalidades de edición para todos los niveles de contenido, incluyendo temas, subtemas, ejercicios y preguntas de control. Para acceder a estas funciones, el docente debe primero seleccionar el elemento específico en la tabla correspondiente. La selección se indica visualmente mediante un cambio en el color de fondo a gris, señalando inequívocamente el elemento activo.

Una vez realizada la selección, se habilitan múltiples opciones de gestión, entre las que se encuentran la edición, activación o desactivación, y acceso al historial de cambios. Para iniciar el proceso de edición, el docente debe activar el botón "Editar", acción que despliega una interfaz que presenta el contenido completo del elemento seleccionado. Esta interfaz permite la modificación integral de todos los componentes del contenido, facilitando una edición fluida y sin restricciones.



Figura 18. Tema seleccionado con la opción editar

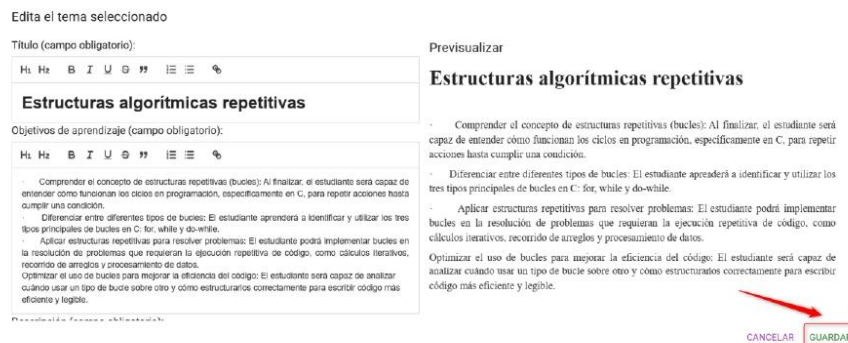


Figura 19. Editar tema.

Al finalizar las modificaciones del contenido, el sistema ejecuta una secuencia de acciones: cierra automáticamente la ventana modal, genera un mensaje de éxito que confirma el éxito de la operación y actualiza la visualización en la tabla correspondiente, donde se refleja inmediatamente el contenido con las modificaciones realizadas.



5.3 Activar o desactivar contenido.

El control del estado de los contenidos (activación o desactivación) sigue un procedimiento similar al de edición. El docente debe seleccionar el elemento específico en la tabla correspondiente, acción que se confirma visualmente mediante el cambio del color de fondo a gris. Una vez realizada la selección, el sistema habilita la opción "Activar o Desactivar", permitiendo al docente modificar el estado del contenido según sus requerimientos.



Figura 20. Tema seleccionado y opción desactivar

Cuando el docente selecciona la opción "Desactivar" en un contenido que se encuentra activo, el sistema presenta un mensaje de confirmación que detalla las implicaciones de esta acción. Este aviso preventivo tiene como propósito asegurar que el docente comprenda las consecuencias de la desactivación del contenido antes de proceder con la acción.



Confirmación

Estimado docente, le informamos que al desactivar un contenido, este dejará de estar visible para los estudiantes y no podrán acceder a los contenidos posteriores relacionados, como subtemas, ejercicios y preguntas.

¿Está seguro de que desea desactivar este tema?

[CANCELAR](#) [CONFIRMAR](#)

Figura 21. Mensaje de aviso.

Al ejecutar la desactivación de un contenido activo, el sistema implementa dos cambios visuales inmediatos: modifica el color de fondo de la celda correspondiente a rojo, indicando su estado inactivo, y actualiza el texto del botón de control a "Activar", permitiendo la posterior reactivación del contenido cuando sea necesario.

Temas existentes	
Estructuras algorítmicas condicionales	
Estructuras algorítmicas repetitivas	
Subprogramas	
Modularidad	
Arreglos	

[CREAR](#) [EDITAR](#) [ACTIVAR](#) [HISTORIAL](#)

Figura 22. Tema desactivado y opción activar



5.4 Historial de cambios

El acceso al historial de cambios sigue el mismo patrón de interacción establecido. El docente debe seleccionar el elemento deseado en la tabla "Temas", acción que se confirma visualmente mediante el cambio del color de fondo a gris. Una vez realizada la selección, el sistema habilita la opción "Historial", permitiendo la visualización del registro de modificaciones realizadas.



Figura 23. Tema seleccionado y opción Historial

Al seleccionar la opción 'Historial', se desplegará una ventana modal que presentará un registro cronológico detallado de las modificaciones realizadas en el tema. Este registro incluirá la fecha y hora de cada operación efectuada, abarcando desde su creación inicial hasta las subsecuentes ediciones, desactivaciones y activaciones del tema en cuestión.



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Fecha	Detalles
24/10/2024, 5:00:44 p. m.	Docente Camigo activó el tema con el título: "Estructuras algorítmicas repetitivas"
22/10/2024, 10:47:25 a. m.	Docente Camigo desactivó el tema con el título: "Estructuras algorítmicas repetitivas"
22/10/2024, 10:41:05 a. m.	Docente Camigo activó el tema con el título: "Estructuras algorítmicas repetitivas"
22/10/2024, 10:41:01 a. m.	Docente Camigo desactivó el tema con el título: "Estructuras algorítmicas repetitivas"
21/10/2024, 11:27:19 a. m.	Docente Camigo activó el tema con el título: "Estructuras algorítmicas repetitivas"
21/10/2024, 11:27:04 a. m.	Docente Camigo desactivó el tema con el título: "Estructuras algorítmicas repetitivas"
21/10/2024, 11:26:39 a. m.	Docente Camigo editó el tema con el título: "Estructuras algorítmicas repetitivas"

CERRAR

Figura 24. Historial de cambios

6 Progreso de los usuarios en el periodo académico actual

El software de apoyo incorpora una funcionalidad para el personal docente que permite monitorear el progreso académico de los estudiantes durante el periodo vigente. Esta herramienta proporciona un análisis tanto del rendimiento general como del avance pormenorizado en temas, subtemas y ejercicios específicos. Para acceder a esta sección de seguimiento, el docente deberá seleccionar la opción 'Simular contenido' ubicada en el área superior derecha de la interfaz.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Bienvenido, Docente Camigo
Periodo académico: 2026-02-17 - 2026-07-17

Gestionar contenido

Temas
Los temas con fondo color rojo están desactivados. Es necesario seleccionar un tema para editar o para cambiar su estado de activo a inactivo.

Buscar tema

Temas existentes

Estructuras algorítmicas condicionales

Estructuras algorítmicas repetitivas

Explora tu interés:
Selecciona un tema.

Figura 25. Opción simular contenido.

Bienvenido, Docente Camigo
Periodo académico: 2026-02-17 - 2026-07-17

Simular contenido

¡Bienvenido a C'amigo, tu aliado perfecto para aprender C!

En "C'amigo", te ofrecemos una herramienta sencilla para fortalecer tu comprensión de la programación en C. Aquí encontrarás **temas**, **subtemas**, **ejercicios** y **preguntas de control** que te ayudarán a repasar lo visto en clase o a adelantarte en tu aprendizaje.

Temas: La base de tu aprendizaje

- Título
- Objetivos de aprendizaje
- Descripción
- Recursos adicionales

Subtemas: Profundiza en cada concepto

- Título
- Objetivos de aprendizaje
- Descripción

C'amigo tu amigo ideal en C
Tu progreso en C'amigo es de: 0%

Temas

Buscar temas

Figura 26. Pagina principal de simular contenido.

Para examinar el progreso de los estudiantes, se debe acceder a la sección 'Visualizar contenido' y dirigirse a la zona inferior derecha del componente principal, donde se encuentra el mensaje de bienvenida y la descripción general del contenido de C'amigo. Al finalizar esta sección introductoria, se despliega la visualización detallada del avance académico de los estudiantes correspondiente al periodo en curso.



Progreso de los estudiantes en Camigo en el periodo académico actual

Usuario	Progreso general	Progreso en temas	Progreso en subtemas	Progreso en ejercicios
Estudiante Uno	0.00%	16.67%	25.00%	25.00%
estudiante dos	0.00%	0.00%	0.00%	0.00%
Estudiante Tres	0.00%	0.00%	0.00%	0.00%
Estudiante Cuatro	0.00%	0.00%	0.00%	0.00%
Estudiante Cinco	0.00%	0.00%	0.00%	0.00%
Juan Carlos	0.00%	0.00%	0.00%	0.00%
Wilman Patricio	0.00%	0.00%	0.00%	0.00%
Edwin René	0.00%	0.00%	0.00%	0.00%
Test Manual	20.00%	50.00%	50.00%	50.00%

Si deseas consultar el progreso de los estudiantes en periodos académicos anteriores, puedes hacerlo aquí:

[VER PERIODOS ACADÉMICOS ANTERIORES](#)

Figura 27. Progreso de los estudiantes en el periodo académico actual

7 Simular contenido

La función de simulación de contenido permite al docente visualizar la interfaz tal como la experimenta el estudiante en el software de apoyo. Para obtener información más detallada sobre esta funcionalidad, se recomienda consultar la sección 5 del manual de usuario del estudiante, el cual se encuentra disponible para su revisión en la sección de anexos o en el documento original en el siguiente enlace: <https://n9.cl/documentosfirmadosdigital>

8 Volver a Temas destacados.

Para regresar a la sección principal tras iniciar sesión, el usuario deberá seleccionar la opción ubicada en el extremo izquierdo, donde se muestra el logotipo de C'amigo. Al seleccionar dicha opción, el sistema redirigirá al usuario a la vista de "Temas destacados".



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación



Figura 28. Temas destacados

9 Cerrar sesión

Para cerrar sesión en el software de apoyo se debe dirigir a la parte superior derecha y seleccionar la opción "Salir".



Figura 29. Cerrar sesión

Ing. José Oswaldo Guamán Quinche, Mg.Sc.	 <p>Firmado electrónicamente por: JOSE OSWALDO GUAMAN QUINCHE</p>
--	---

Anexo 10: Manual de usuario para administrador.



unl
Universidad
Nacional
de Loja

 Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

FACULTAD DE LA ENERGÍA, LAS INDUSTIRAS Y LOS RECURSOS
NATURALES NO RENOVABLES

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Revisado y aprobado por:**
 - Ing. José Oswaldo Guamán, director del Trabajo de Integración Curricular.

Descripción: El presente manual de usuario documenta de manera exhaustiva las funciones y responsabilidades asignadas al usuario con rol Administrador. Este perfil tiene acceso a diversas funcionalidades del software de apoyo, las cuales han sido específicamente diseñadas para ser una herramienta de apoyo en el proceso de aprendizaje de programación en lenguaje C.



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

MANUAL DE USUARIO: ADMINISTRADOR

1 Ingreso a la aplicación web

Para acceder al software de apoyo, el usuario deberá dirigirse a su navegador web e ingresar la URL correspondiente en la barra de direcciones.

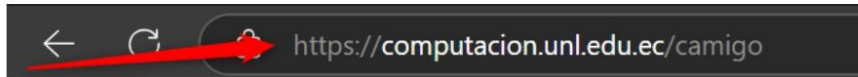


Figura 1. Accediendo al software de apoyo

Luego, se presenta la página web.

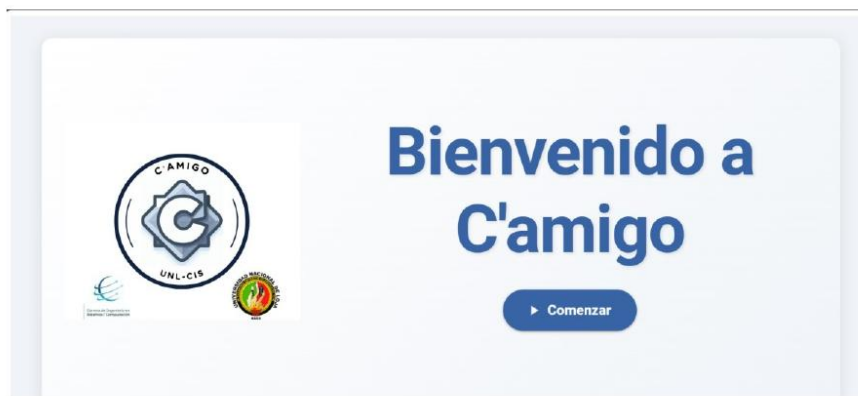


Figura 2. Página de inicio para el software de apoyo

Para acceder al software de apoyo, seleccione la opción "Comenzar". Esta acción le redirigirá a la sección de **Inicio de sesión**. Sin embargo, para obtener una cuenta con privilegios de administrador, el usuario deberá primero dirigirse a la sección **Registro de administradores**.

2 Registro de administradores

Para la creación de una cuenta con rol de administrador, se requiere que el interesado envíe una solicitud formal mediante correo electrónico a la dirección camigounlcis@gmail.com.

Es importante destacar que el sistema generalmente no va a tener un gran número de usuarios con este rol, dado que la responsabilidad del laboratorio de software de la carrera de Computación recae en un número reducido de encargados.



Es preciso señalar que el sistema cuenta con un usuario administrador preestablecido, las credenciales de acceso para esta cuenta predeterminada son las siguientes:

- Correo electrónico: camigoAdmin@unl.edu.ec
- Contraseña: admin123

Para registrar un nuevo administrador en C'amigo por primera vez, se debe iniciar sesión con el administrador preestablecido.



Figura 3. Inicio de sesión con credenciales preestablecidas

Una vez que las credenciales del administrador predeterminado sean verificadas correctamente, el sistema concederá acceso a la interfaz de temas destacados. En esta sección se exhiben visualmente los cinco temas que han recibido la mejor valoración por parte de la comunidad de usuarios. Adicionalmente, se proporciona acceso al manual de usuario específico para el rol de administrador.



Figura 4. Temas destacados

En la sección correspondiente a los manuales de usuario, se presenta un botón que indica el tipo de usuario. Esta funcionalidad permite al usuario descargar su manual específico cuando lo requiera.

Manuales de Usuario



Figura 5. Manuales de usuario

Luego se presiona la opción de "Administración", para ir al panel de administración que nos permite realizar varias acciones entre ellas el registro de administradores.



Figura 6. Opción de administración

Al optar por la sección "Administración", se despliega una interfaz que ofrece al administrador diversas funcionalidades, las cuales se detallarán exhaustivamente en el transcurso de este documento. En la tabla ubicada a la derecha, el administrador tiene la capacidad de visualizar la totalidad de los usuarios con privilegios de administración, así como acceder a una opción para registrar nuevos administradores en el sistema.

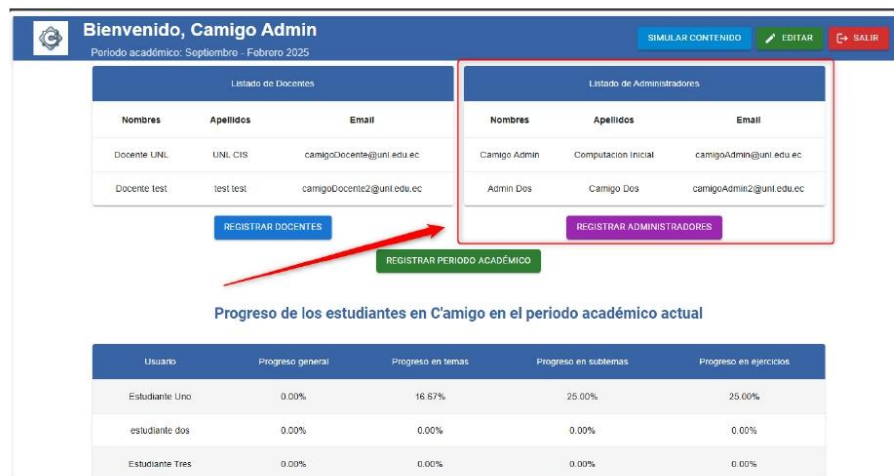


Figura 7. Página de Administración

Al seleccionar la opción "Registrar Administradores", se despliega un modal que contiene un formulario. En este, el usuario debe completar obligatoriamente todos los campos solicitados para proceder con el registro de un nuevo administrador.



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Registrar Administrador

Nombres *

Apellidos *

Email *

Clave *

CANCELAR REGISTRARSE

Figura 8. Formulario de registro de administradores

Cuando la creación de un nuevo usuario se completa correctamente, se muestra un mensaje de éxito en la interfaz. Este mensaje confirma que el proceso de registro se ha llevado a cabo satisfactoriamente.

Registrar Administrador

Nombres *
Admin Manual

Apellidos *
Camigo Manual

Email *
camigoAdminManual@unl.edu.ec

Clave *
Manual123

CANCELAR REGISTRARSE

Figura 9. Formulario de registro completo



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

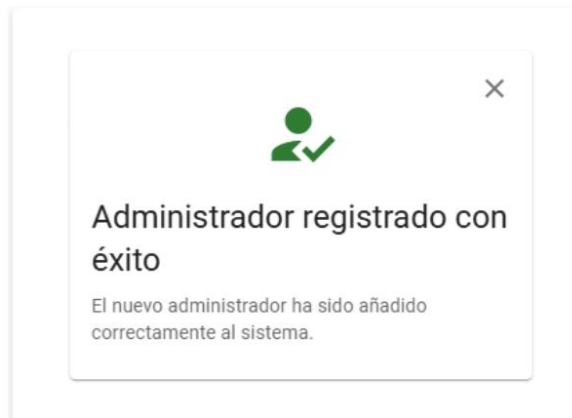


Figura 10. Mensaje de registro exitoso.

Una vez que se cierra el mensaje de confirmación exitosa, el nuevo administrador registrado se visualiza automáticamente en la tabla que enumera a todos los administradores del sistema.

Listado de Administradores		
Nombres	Apellidos	Email
Camigo Admin	Computacion Inicial	camigoAdmin@unl.edu.ec
Admin Dos	Camigo Dos	camigoAdmin2@unl.edu.ec
Admin Manual	Camigo Manual	camigoAdminManual@unl.edu.ec

Figura 11. Tabla de listado de administradores



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

3 Inicio de sesión

3.1 Usuarios externos

Para acceder al formulario de inicio sesión, se debe acceder a la página predeterminada de la aplicación, presentada en la sección **Ingreso a la aplicación web**

Para acceder al sistema, el usuario deberá ingresar las credenciales correspondientes a la cuenta previamente creada. Los datos obligatorios requeridos para un inicio de sesión exitoso son el correo electrónico y la contraseña.

Figura 12. Ingreso de datos en el formulario de Inicio de Sesión.

Una vez que las credenciales del administrador sean verificadas correctamente, el sistema concederá acceso a la interfaz de temas destacados. En esta sección se exhiben visualmente los cinco temas que han recibido la mejor valoración por parte de la comunidad de usuarios.



Figura 13. Opción de administración

Al seleccionar la sección "Administración", se presenta una interfaz que proporciona al usuario con privilegios administrativos un conjunto de funcionalidades. Estas incluyen la capacidad de registrar nuevos docentes, incorporar administradores adicionales, establecer un nuevo periodo académico y acceder a la visualización de periodos académicos previos.



Figura 14. Página de Administración

3.2 Usuarios internos.

El acceso también es posible utilizando el IAM de la carrera de Computación. Para esto, el usuario debe seleccionar esta opción en la pantalla de inicio de sesión.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación



Figura 15. Inicio de sesión

Luego ingresar las credenciales de acceso al sistema y presionar la opción de “Iniciar Sesión”

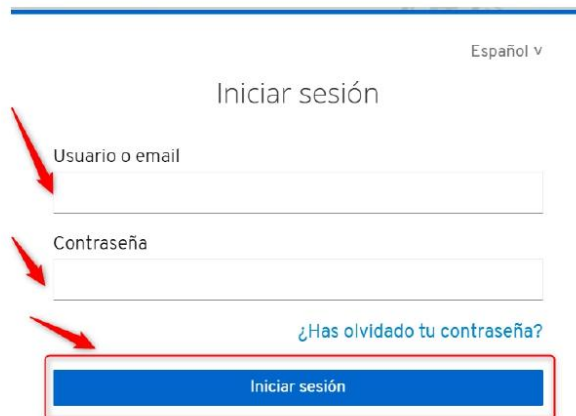


Figura 16. Inicio de sesión mediante el IAM Computación

4 Editar información de usuario

Para acceder al apartado de perfil de usuario, se debe dirigir a la parte superior derecha y seleccionar la opción "Editar", ubicada junto a "Administración".



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Bienvenido, Camigo Admin
Periodo académico: Septiembre - Febrero 2025

ADMINISTRACIÓN EDITAR SALIR

Temas destacados

¡Bienvenido! Aquí te presentamos los 5 temas más populares basados en los "Me gusta" recibidos. Estos temas reflejan el contenido que más ha resonado con nuestra comunidad. ¡Explora y descubre lo que otros encuentran interesante!

Tema	Número de "Me gusta"
1º	5
2º	4
3º	3
4º	2
5º	1

Manuales de Usuario

ADMINISTRADOR

Figura 17. Opción "Editar perfil"

En la opción de editar perfil, se despliega un modal que muestra los datos modificables del usuario: nombres, apellidos, correo electrónico y contraseña. Para efectuar cambios en estos campos, se requiere ingresar la contraseña actual y seleccionar "Validar contraseña".

Editar Usuario

Nombres *
Test Manual

Apellidos *
Test Manual

Email *
manualEstudiante@unl.edu.ec

Contraseña Actual *
.....

VALIDAR CONTRASEÑA ACTUAL

CERRAR GUARDAR CAMBIOS

Figura 18. Información del usuario a editar

Una vez que el usuario valide su contraseña actual, el sistema le permitirá, si así lo desea, modificar su contraseña de inicio de sesión.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Editar Usuario

manualEstudiante@unl.edu.ec

Contraseña Actual *

VALIDAR CONTRASEÑA ACTUAL

Nueva Contraseña (opcional)

Confirmar Nueva Contraseña

CERRAR GUARDAR CAMBIOS

Figura 19. Cambio de contraseña en el editar perfil

5 Administración

En el área superior derecha se ubica la opción "Administración". Al activarla, se despliega ante el usuario el conjunto completo de funciones administrativas disponibles. Estas abarcan la gestión de nuevos docentes y administradores, la creación de periodos académicos, así como la revisión de periodos anteriores, incluyendo el análisis del progreso de los usuarios durante dichos intervalos académicos concluidos.



Figura 20. Opción "Visualizar contenido"

Vamos a detallar cada operación que puede realizar el usuario de tipo administrador en las siguientes secciones.



6 Registrar docentes

Al estar en la ventana de administración podemos observar que podemos ingresar dos clases de usuarios uno es docente y el otro es administrador, el cual ya se detalló en la sección de **Registro de administradores**, ahora al crear docentes es igual presionamos la opción “Registrar Docentes”, el cual nos abrirá un formulario con sus cuatro datos obligatorios que son nombres, apellidos, correo electrónico y clave.

The screenshot shows the 'Camigo Admin' dashboard. At the top, it says 'Bienvenido, Camigo Admin' and 'Periodo académico: Septiembre - Febrero 2025'. There are buttons for 'SIMULAR CONTENIDO', 'EDITAR', and 'SALIR'. Below this, there are two tables: 'Listado de Docentes' and 'Listado de Administradores'. The 'Listado de Docentes' table has columns for 'Nombres', 'Apellidos', and 'Email'. It lists two users: 'Docente UNL' and 'Docente test'. A blue button labeled 'REGISTRAR DOCENTES' is located below this table. A red arrow points to this button. Below the tables, there is a green button 'REGISTRAR PERIODO ACADÉMICO' and a purple button 'REGISTRAR ADMINISTRADORES'. At the bottom, there is a section titled 'Progreso de los estudiantes en Camigo en el periodo académico actual' with a table showing progress for 'Estudiante Uno' and 'estudiante dos' across four categories: 'Progreso general', 'Progreso en temas', 'Progreso en subtemas', and 'Progreso en ejercicios'.

Usuario	Progreso general	Progreso en temas	Progreso en subtemas	Progreso en ejercicios
Estudiante Uno	0.00%	16.67%	25.00%	25.00%
estudiante dos	0.00%	0.00%	0.00%	0.00%

Figura 21. Registrar docentes.

Al seleccionar la opción "Registrar Docentes", se despliega un modal que contiene un formulario. En este, el usuario debe completar obligatoriamente todos los campos solicitados para proceder con el registro de un nuevo docente.



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Registrar Docente

Nombres *

Apellidos *

Email *

Clave *

CANCELAR REGISTRARSE

Figura 22. Formulario de registro de administradores

Cuando la creación de un nuevo docente se completa correctamente, se muestra un mensaje de éxito en la interfaz. Este mensaje confirma que el proceso de registro se ha llevado a cabo satisfactoriamente.

Registrar Docente

Nombres *
Docente Manual

Apellidos *
Manual UNL

Email *
camigoDocenteManual@unl.edu.ec

Clave *
docente123

CANCELAR REGISTRARSE

Figura 23. Formulario de registro completo



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

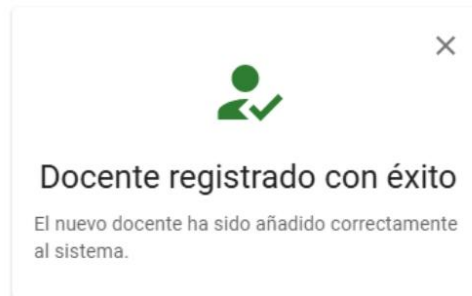


Figura 24. Mensaje de registro exitoso.

Una vez que se cierra el mensaje de confirmación exitosa, el nuevo docente registrado se visualiza automáticamente en la tabla que enumera a todos los docentes del sistema.

Listado de Docentes		
Nombres	Apellidos	Email
Docente UNL	UNL CIS	camigoDocente@unl.edu.ec
Docente test	test test	camigoDocente2@unl.edu.ec
Docente Manual	Manual UNL	camigoDocenteManual@unl.edu.ec

 **REGISTRAR DOCENTES**

Figura 25. Tabla de listado de administradores

7 Registrar periodo académico.

Para registrar un nuevo periodo académico debemos seleccionar la opción de “Registrar periodo académico”.



UNL

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Bienvenido, Camigo Admin
Periodo académico: Septiembre - Febrero 2025

SIMULAR CONTENIDO EDITAR SALIR

Listado de Docentes			Listado de Administradores		
Nombres	Apellidos	Email	Nombres	Apellidos	Email
Docente UNL	UNL CIS	camigoDocente@unl.edu.ec	Camigo Admin	Computacion Inicial	camigoAdmin@unl.edu.ec
Docente test	test test	camigoDocente2@unl.edu.ec	Admin Dos	Camigo Dos	camigoAdmin2@unl.edu.ec
Docente Manual	Manual UNL	camigoDocenteManual@unl.edu.ec	Admin Manual	Camigo Manual	camigoAdminManual@unl.edu.ec

REGISTRAR DOCENTES REGISTRAR PERIODO ACADÉMICO REGISTRAR ADMINISTRADORES

Figura 26. Opción Registrar periodo académico

Al seleccionar la opción de registrar un nuevo periodo académico, se muestra un mensaje informativo al administrador. Este detalla las implicaciones de iniciar un nuevo ciclo, que incluyen: el reinicio del progreso de todos los usuarios del sistema, el almacenamiento del progreso actual en la sección **Periodos académicos anteriores**, y la confirmación de que los usuarios mantendrán acceso a este contenido. El administrador debe indicar que ha leído y comprendido estas condiciones, así como confirmar su decisión de proceder con la creación del nuevo periodo académico.

Nuevo Periodo Académico

Estimado administrador, antes de crear el periodo académico debe tener en cuenta los siguientes puntos:

- El progreso de todos los usuarios se volverá a reiniciar
- El progreso de los usuarios será guardado en su periodo académico correspondiente
- Usted podrá revisar el contenido en la sección de administración

He leído los puntos anteriores y quiero crear un nuevo periodo académico

CANCELAR CREAR PERIODO ACADÉMICO


Figura 27. Mensaje de confirmación de creación de un nuevo periodo

Al seleccionar la opción "Crear periodo académico", se despliega el formulario correspondiente para el ingreso de los datos del nuevo periodo académico.



Registrar Período Académico

Fecha de inicio *
dd/mm/aaaa 

Fecha de fin *
dd/mm/aaaa 

[CANCELAR](#) [REGISTRAR](#)

Figura 28. Registrar nuevo periodo académico

En caso de que la creación del nuevo periodo académico se efectúe correctamente, el sistema procede a archivar el periodo académico anterior en la sección denominada **Periodos académicos anteriores**. Simultáneamente, se visualiza el progreso de los estudiantes reiniciado para el nuevo ciclo.

8 Progreso de los usuarios en el periodo académico actual

El administrador dispone de una funcionalidad adicional dentro del software de apoyo que le permite visualizar el progreso de los estudiantes en el periodo académico vigente. Esta sección ofrece una visión detallada del avance general, así como el progreso específico en temas, subtemas y ejercicios.

Progreso de los estudiantes en C'amigo en el periodo académico actual

Usuario	Progreso general	Progreso en temas	Progreso en subtemas	Progreso en ejercicios
Estudiante Uno	0.00%	16.67%	25.00%	25.00%
estudiante dos	0.00%	0.00%	0.00%	0.00%
Estudiante Tres	0.00%	0.00%	0.00%	0.00%
Estudiante Cuatro	0.00%	0.00%	0.00%	0.00%
Estudiante Cinco	0.00%	0.00%	0.00%	0.00%
Juan Carlos	0.00%	0.00%	0.00%	0.00%
Wilmar Patricio	0.00%	0.00%	0.00%	0.00%
Edwin René	0.00%	0.00%	0.00%	0.00%
Test Manual	20.00%	50.00%	50.00%	50.00%

Si deseas consultar el progreso de los estudiantes en periodos académicos anteriores, puedes hacerlo aquí:

[VER PERIODOS ACADÉMICOS ANTERIORES](#)

Figura 29. Progreso de los estudiantes en el periodo académico actual



9 Periodos académicos anteriores

El administrador, como se mencionó previamente, tiene la capacidad de acceder a los periodos académicos anteriores y examinar el progreso tanto de manera general como detallada. Para efectuar esta acción, es necesario seleccionar inicialmente la opción "Ver periodos académicos anteriores".

Progreso de los estudiantes en Camigo en el periodo académico actual

Usuario	Progreso general	Progreso en temas	Progreso en subtemas	Progreso en ejercicios
Estudiante Uno	0.00%	16.67%	25.00%	25.00%
estudiante dos	0.00%	0.00%	0.00%	0.00%
Estudiante Tres	0.00%	0.00%	0.00%	0.00%
Estudiante Cuatro	0.00%	0.00%	0.00%	0.00%
Estudiante Cinco	0.00%	0.00%	0.00%	0.00%
Juan Carlos	0.00%	0.00%	0.00%	0.00%
Wilman Patricio	0.00%	0.00%	0.00%	0.00%
Edwin René	0.00%	0.00%	0.00%	0.00%
Test Manual	20.00%	50.00%	50.00%	50.00%

Si deseas consultar el progreso de los estudiantes en periodos académicos anteriores, puedes hacerlo aquí:

[VER PERIODOS ACADÉMICOS ANTERIORES](#)

Figura 30. Opción periodos académicos anteriores

Al acceder a esta sección, el sistema despliega una nueva ventana que exhibe la totalidad de los periodos académicos precedentes.



Figura 31. Periodos académicos anteriores

Al seleccionarse un periodo académico específico, el sistema presenta a continuación una vista general de los progresos correspondientes a todos los estudiantes inscritos durante dicho ciclo lectivo.



Períodos académicos anteriores

VOLVER A ADMINISTRACIÓN

Periodo	Año
Marzo - Agosto	2024

Progreso de los usuarios

Usuario	Progreso general	Progreso en temas	Progreso en subtemas	Progreso en ejercicios
Estudiante Uno	50.00%	50.00%	50.00%	50.00%
estudiante dos	0.00%	0.00%	0.00%	0.00%
Estudiante Tres	0.00%	0.00%	0.00%	0.00%
Estudiante Cuatro	0.00%	0.00%	0.00%	0.00%
Estudiante Cinco	0.00%	0.00%	0.00%	0.00%
Juan Carlos	0.00%	0.00%	0.00%	0.00%
Wilman Patricio	0.00%	0.00%	0.00%	0.00%
Edwin René	0.00%	0.00%	0.00%	0.00%

Figura 32. Progreso general de los usuarios

Al seleccionarse un estudiante en particular, se despliega una ventana modal que detalla exhaustivamente el progreso del usuario seleccionado. Esta presentación permite al administrador examinar el avance desglosado por tema, subtema y ejercicio.

Progreso detallado del estudiante. X

Progreso detallado					
Tema	Progreso	Subtema	Progreso	Ejercicio	Progreso
Estructuras algorítmicas repetitivas	0%	Estructura Para (For)	0%	Imprimir Números Pares entre 1 y 50	0%
Estructuras algorítmicas repetitivas	0%	Estructura Mientras (While) en C	0%	Contar hasta un Número Específico	0%
Estructuras algorítmicas repetitivas	0%	Estructura Hacer...Mientras (DO...While) en C	0%	Menú de Opciones	0%
Estructuras algorítmicas condicionales	100%	Estructura condicional simple	100%	Verificador de Número Par	100%
Estructuras algorítmicas condicionales	100%	Estructura condicional doble	100%	Clasificador de Temperatura	100%
Estructuras					

Figura 33. Progreso detallado del estudiante



10 Simular contenido

La función de simulación de contenido permite al administrador visualizar la interfaz tal como la experimenta el estudiante en el software de apoyo. Para obtener información más detallada sobre esta funcionalidad, se recomienda consultar la sección 5 del manual de usuario del estudiante, el cual se encuentra disponible para su revisión en la sección de anexos o en el documento original en el siguiente enlace: <https://n9.cl/documentosfirmadosdigital>

11 Volver a Temas destacados.

Para regresar a la sección principal tras iniciar sesión, el usuario deberá seleccionar la opción ubicada en el extremo izquierdo, donde se muestra el logotipo de C'amigo. Al seleccionar dicha opción, el sistema redirigirá al usuario a la vista de "Temas destacados".



Figura 34. Temas destacados

12 Cerrar sesión

Para cerrar sesión en el software de apoyo se debe dirigir a la parte superior derecha y seleccionar la opción "Salir".



Figura 35. Cerrar sesión



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

<p>Ing. José Oswaldo Guamán Quinche, Mg.Sc.</p>	 <p>Escanea este documento del: JOSE OSWALDO GUAMAN QUINCHE</p>
---	---

Anexo 11: Pruebas de aceptación realizadas a los estudiantes del primer ciclo de la carrera de Computación de la materia “Teoría de la programación”



UNL

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Revisado y aprobado por:**
 - Ing. José Guamán Mg. Sc. director del Trabajo de Integración Curricular.

Descripción: El presente anexo documenta los resultados de las pruebas de aceptación finales del sistema, las cuales verifican el cumplimiento íntegro de las historias de usuario establecidas. Cada historia corresponde a un requisito esencial del sistema, detallando su implementación. Estas pruebas son críticas para el proceso de validación, garantizando que el sistema proporcione todas las funcionalidades requeridas por el cliente. La culminación exitosa de este proceso certifica que el sistema cumple plenamente con los requerimientos iniciales, previamente establecidos y validados con el cliente.



PRUEBAS DE ACEPTACIÓN FINALES DEL SISTEMA

La **Tabla 1** sintetiza las historias de usuario, detallando identificadores, nombres y estados, facilitando una visión global de las pruebas de aceptación del sistema.

Tabla 1. Pruebas de aceptación finales en base a las Historias de Usuario

HU #	Historia de usuario	Estado
HU001	Crear usuario	Aprobado
HU002	Editar usuario	Aprobado
HU003	Listar temas	Aprobado
HU004	Registrar temas	Aprobado
HU005	Editar temas	Aprobado
HU006	Historial de cambios por tema	Aprobado
HU007	Activar/Desactivar temas	Aprobado
HU008	Buscar temas	Aprobado
HU009	Visualizar temas	Aprobado
HU010	Listar subtemas	Aprobado
HU011	Registrar subtemas	Aprobado
HU012	Editar subtemas	Aprobado
HU013	Historial de cambios por subtema	Aprobado
HU014	Activar/Desactivar subtemas	Aprobado
HU015	Buscar subtemas	Aprobado
HU016	Visualizar subtemas	Aprobado
HU017	Listar ejercicios	Aprobado
HU018	Registrar ejercicios	Aprobado
HU019	Editar ejercicios	Aprobado
HU020	Historial de cambios por ejercicio	Aprobado
HU021	Activar/Desactivar ejercicios	Aprobado
HU022	Buscar ejercicios	Aprobado
HU023	Visualizar ejercicios	Aprobado
HU024	Proporcionar retroalimentación de ejercicios	Aprobado
HU025	Listar preguntas de control	Aprobado
HU026	Registrar preguntas de control	Aprobado
HU027	Editar preguntas de control	Aprobado
HU028	Historial de cambios por pregunta de control	Aprobado
HU029	Activar/Desactivar preguntas de control	Aprobado
HU030	Buscar preguntas de control	Aprobado
HU031	Visualizar preguntas de control	Aprobado
HU032	Responder pregunta de control	Aprobado
HU033	Listar comentarios	Aprobado
HU034	Registrar comentarios	Aprobado
HU035	Editar comentarios	Aprobado
HU036	Visualizar comentarios	Aprobado
HU037	Visualizar progreso	Aprobado
HU038	Cambiar periodo académico	Aprobado
HU039	Temas destacados	Aprobado
HU040	Notificaciones por el email de nuevo contenido	Aprobado



UNL

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Para constancia y validación de la información expuesta, firman a continuación estudiantes del primer ciclo de la carrera de Computación, en su calidad de clientes del TIC y firmado y aprobado por el director del TIC, Ing. José Guamán:

Nombres	Firma	Curso
Emilio José Gálvez Merchán		1
Steven Alexander Jumbo Jaramila		1
Loandro David Cevallos Villavicencio		1
Victor David Urcus Montano		1
Eduardo Daniel Moreno Cuenca		1
Janileth Catherine Gaona Abrigo		1
Pedro Luis Jimenez Mora		1
Gyna Magerly Yupanqui Romero		1
Johanel Alexei Castillo Asanza		1
Alexis Emmanuel Renteria Torres		1
Abraham David Moracho Grageda		1
Luis Humberto Rios Barrera		1
Miguel Atejandro Armas Ordóñez		1
Steven Raúl Pérez Ríos		1



UNL

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Nombres	Firma	Curso
Joseph Adrian Aguilar Asanza		Computación
Richard Ariel Chamba Santin		Computación.
Gabriel Alejandro Suárez Poma		Computación.
Yandri Alexander Piscocoma Jaramillo		computación
Paul Alejandro Soto Lastillos		computación
Darina Alejandra Maldonado Pacheco		Computación
Jaime Alejandro Landóuri Román		Computación
María Soledad Buit Camacho		Computación
Mark Anthony González Jaramillo		Computación
Lucas Sebastián Orellana M		Computación
Wilson Patricio Palma Saco		computación.
Odalys Camila Rosillo G.		Computación
Alex Francisco Sogho Serrano		1 Computación.



UNL

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación

Nombres	Firma	Curso
Ivett Anahi Zaragocin. Mora		Primer ciclo
Doménica Anahi Rojas Curimilma		Primer ciclo.
Victor fernando Roa Carrión		Primer Ciclo
Alan Sebastian Montesinos Pachar		Primer ciclo.
Fernando Sebastian Patiño Diaz		Primer ciclo
Justin Xavier Vasquez Calderón		Primer ciclo.
Jahiri Alexander Velásquez Cuenca		Primer ciclo
Emilo José Flores Gollardo		Primer ciclo
Francis Jasve Velázquez Jimenez		Primer ciclo
Andesson Mateo Coello Jasomillo		Primer ciclo
Ronal Patriuo Jumbo Abad		Primer Ciclo
Devick Paul Vargas Vasquez		Primer ciclo
Katheryn Melissa Contento Maldonado		1er ciclo
Alejandro Emanuel Padilla Espinoza		1er ciclo

Ing. José Guamán, Mg. Sc.



JOSE OSWALDO GUAMAN QUINCHE

Anexo 12: Pruebas de aceptación realizadas a los docentes del área de programación



UNIVERSIDAD NACIONAL DE LOJA



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Revisado y aprobado por:**
 - Ing. Wilman Chamba Mg. Sc. Docente de la materia “Teoría de la programación”.
 - Ing. Edison Coronel Mg. Sc. Docente de la materia “Programación Orientada a Objetos”.
 - Ing. René Guamán Mg. Sc. Docente de la materia “Programación Paralela”.
 - Ing. José Guamán Mg. Sc. Docente de la materia “Plataformas web”.

Descripción: El presente anexo documenta los resultados de las pruebas de aceptación finales del sistema, las cuales verifican el cumplimiento íntegro de las historias de usuario establecidas. Cada historia corresponde a un requisito esencial del sistema, detallando su implementación. Estas pruebas son críticas para el proceso de validación, garantizando que el sistema proporcione todas las funcionalidades requeridas por el cliente. La culminación exitosa de este proceso certifica que el sistema cumple plenamente con los requerimientos iniciales, previamente establecidos y validados con el cliente.




PRUEBAS DE ACEPTACIÓN FINALES DEL SISTEMA

La **Tabla 1** sintetiza las historias de usuario, detallando identificadores, nombres y estados, facilitando una visión global de las pruebas de aceptación del sistema.

Tabla 1. Pruebas de aceptación finales en base a las Historias de Usuario

HU #	Historia de usuario	Estado
HU001	Crear usuario	Aprobado
HU002	Editar usuario	Aprobado
HU003	Listar temas	Aprobado
HU004	Registrar temas	Aprobado
HU005	Editar temas	Aprobado
HU006	Historial de cambios por tema	Aprobado
HU007	Activar/Desactivar temas	Aprobado
HU008	Buscar temas	Aprobado
HU009	Visualizar temas	Aprobado
HU010	Listar subtemas	Aprobado
HU011	Registrar subtemas	Aprobado
HU012	Editar subtemas	Aprobado
HU013	Historial de cambios por subtema	Aprobado
HU014	Activar/Desactivar subtemas	Aprobado
HU015	Buscar subtemas	Aprobado
HU016	Visualizar subtemas	Aprobado
HU017	Listar ejercicios	Aprobado
HU018	Registrar ejercicios	Aprobado
HU019	Editar ejercicios	Aprobado
HU020	Historial de cambios por ejercicio	Aprobado
HU021	Activar/Desactivar ejercicios	Aprobado
HU022	Buscar ejercicios	Aprobado
HU023	Visualizar ejercicios	Aprobado
HU024	Proporcionar retroalimentación de ejercicios	Aprobado
HU025	Listar preguntas de control	Aprobado
HU026	Registrar preguntas de control	Aprobado
HU027	Editar preguntas de control	Aprobado
HU028	Historial de cambios por pregunta de control	Aprobado
HU029	Activar/Desactivar preguntas de control	Aprobado
HU030	Buscar preguntas de control	Aprobado
HU031	Visualizar preguntas de control	Aprobado
HU032	Responder pregunta de control	Aprobado
HU033	Listar comentarios	Aprobado
HU034	Registrar comentarios	Aprobado
HU035	Editar comentarios	Aprobado
HU036	Visualizar comentarios	Aprobado
HU037	Visualizar progreso	Aprobado
HU038	Cambiar periodo académico	Aprobado
HU039	Temas destacados	Aprobado
HU040	Notificaciones por el email de nuevo contenido	Aprobado

Para constancia y validación de la información expuesta, firman a continuación docentes del área de programación de la carrera de Computación, en su calidad de clientes del TIC:

<p>Ing. Wilman Chamba, Mg. Sc.</p>	 <p>Firmado electrónicamente por: WILMAN PATRICIO CHAMBA ZARAGOCIN</p>
<p>Ing. Edison Coronel, Mg. Sc.</p>	 <p>Firmado electrónicamente por: EDISON LEONARDO CORONEL ROMERO</p>
<p>Ing. René Guamán, Mg. Sc.</p>	 <p>Firmado electrónicamente por: EDWIN RENE GUAMAN QUINCHE</p>
<p>Ing. José Guamán, Mg. Sc.</p>	 <p>Firmado electrónicamente por: JOSE OSWALDO GUAMAN QUINCHE</p>

Anexo 13: Pruebas de aceptación realizadas al encargado de los laboratorios de software de la carrera de computación, Ing. Luis Sinche



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**

- Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.

- **Revisado y aprobado por:**

- Ing. Luis Darío Sinche Cueva, Mg. Sc., técnico docente del laboratorio de software de la Universidad Nacional de Loja.

Descripción: El presente anexo documenta los resultados de las pruebas de aceptación finales del sistema, las cuales verifican el cumplimiento íntegro de las historias de usuario establecidas. Cada historia corresponde a un requisito esencial del sistema, detallando su implementación. Estas pruebas son críticas para el proceso de validación, garantizando que el sistema proporcione todas las funcionalidades requeridas por el cliente. La culminación exitosa de este proceso certifica que el sistema cumple plenamente con los requerimientos iniciales, previamente establecidos y validados con el cliente.



PRUEBAS DE ACEPTACIÓN FINALES DEL SISTEMA

La **Tabla 1** sintetiza las historias de usuario, detallando identificadores, nombres y estados, facilitando una visión global de las pruebas de aceptación del sistema.

Tabla 1. Pruebas de aceptación finales en base a las Historias de Usuario

HU #	Historia de usuario	Estado
HU001	Crear usuario	Aprobado
HU002	Editar usuario	Aprobado
HU003	Listar temas	Aprobado
HU004	Registrar temas	Aprobado
HU005	Editar temas	Aprobado
HU006	Historial de cambios por tema	Aprobado
HU007	Activar/Desactivar temas	Aprobado
HU008	Buscar temas	Aprobado
HU009	Visualizar temas	Aprobado
HU010	Listar subtemas	Aprobado
HU011	Registrar subtemas	Aprobado
HU012	Editar subtemas	Aprobado
HU013	Historial de cambios por subtema	Aprobado
HU014	Activar/Desactivar subtemas	Aprobado
HU015	Buscar subtemas	Aprobado
HU016	Visualizar subtemas	Aprobado
HU017	Listar ejercicios	Aprobado
HU018	Registrar ejercicios	Aprobado
HU019	Editar ejercicios	Aprobado
HU020	Historial de cambios por ejercicio	Aprobado
HU021	Activar/Desactivar ejercicios	Aprobado
HU022	Buscar ejercicios	Aprobado
HU023	Visualizar ejercicios	Aprobado
HU024	Proporcionar retroalimentación de ejercicios	Aprobado
HU025	Listar preguntas de control	Aprobado
HU026	Registrar preguntas de control	Aprobado
HU027	Editar preguntas de control	Aprobado
HU028	Historial de cambios por pregunta de control	Aprobado
HU029	Activar/Desactivar preguntas de control	Aprobado
HU030	Buscar preguntas de control	Aprobado
HU031	Visualizar preguntas de control	Aprobado
HU032	Responder pregunta de control	Aprobado
HU033	Listar comentarios	Aprobado
HU034	Registrar comentarios	Aprobado
HU035	Editar comentarios	Aprobado
HU036	Visualizar comentarios	Aprobado
HU037	Visualizar progreso	Aprobado
HU038	Cambiar periodo académico	Aprobado
HU039	Temas destacados	Aprobado
HU040	Notificaciones por el email de nuevo contenido	Aprobado




Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Para constancia y validación de la información expuesta, firma a continuación el técnico docente del laboratorio de software de la Universidad Nacional de Loja, en su calidad de cliente del proyecto:

<p>Ing. Luis Darío Sinche Cueva, Mg. Sc.</p>	 <p>Firma electrónica por: LUIS DARIO SINCHE CUEVA</p>
--	---

Anexo 14: Pruebas de aceptación realizadas al director de la carrera de computación, Ing. Pablo Ordoñez



Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.

- **Revisado y aprobado por:**
 - Ing. Pablo Fernando Ordoñez Ordoñez, Mg. Sc., director de la carrera de Computación y encargado principal del laboratorio de software de la Universidad Nacional de Loja.

Descripción: El presente anexo documenta los resultados de las pruebas de aceptación finales del sistema, las cuales verifican el cumplimiento íntegro de las historias de usuario establecidas. Cada historia corresponde a un requisito esencial del sistema, detallando su implementación. Estas pruebas son críticas para el proceso de validación, garantizando que el sistema proporcione todas las funcionalidades requeridas por el cliente. La culminación exitosa de este proceso certifica que el sistema cumple plenamente con los requerimientos iniciales, previamente establecidos y validados con el cliente.



PRUEBAS DE ACEPTACIÓN FINALES DEL SISTEMA

La **Tabla 1** sintetiza las historias de usuario, detallando identificadores, nombres y estados, facilitando una visión global de las pruebas de aceptación del sistema.

Tabla 1. Pruebas de aceptación finales en base a las Historias de Usuario

HU #	Historia de usuario	Estado
HU001	Crear usuario	Aprobado
HU002	Editar usuario	Aprobado
HU003	Listar temas	Aprobado
HU004	Registrar temas	Aprobado
HU005	Editar temas	Aprobado
HU006	Historial de cambios por tema	Aprobado
HU007	Activar/Desactivar temas	Aprobado
HU008	Buscar temas	Aprobado
HU009	Visualizar temas	Aprobado
HU010	Listar subtemas	Aprobado
HU011	Registrar subtemas	Aprobado
HU012	Editar subtemas	Aprobado
HU013	Historial de cambios por subtema	Aprobado
HU014	Activar/Desactivar subtemas	Aprobado
HU015	Buscar subtemas	Aprobado
HU016	Visualizar subtemas	Aprobado
HU017	Listar ejercicios	Aprobado
HU018	Registrar ejercicios	Aprobado
HU019	Editar ejercicios	Aprobado
HU020	Historial de cambios por ejercicio	Aprobado
HU021	Activar/Desactivar ejercicios	Aprobado
HU022	Buscar ejercicios	Aprobado
HU023	Visualizar ejercicios	Aprobado
HU024	Proporcionar retroalimentación de ejercicios	Aprobado
HU025	Listar preguntas de control	Aprobado
HU026	Registrar preguntas de control	Aprobado
HU027	Editar preguntas de control	Aprobado
HU028	Historial de cambios por pregunta de control	Aprobado
HU029	Activar/Desactivar preguntas de control	Aprobado
HU030	Buscar preguntas de control	Aprobado
HU031	Visualizar preguntas de control	Aprobado
HU032	Responder pregunta de control	Aprobado
HU033	Listar comentarios	Aprobado
HU034	Registrar comentarios	Aprobado
HU035	Editar comentarios	Aprobado
HU036	Visualizar comentarios	Aprobado
HU037	Visualizar progreso	Aprobado
HU038	Cambiar periodo académico	Aprobado
HU039	Temas destacados	Aprobado
HU040	Notificaciones por el email de nuevo contenido	Aprobado




unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Para constancia y validación de la información expuesta, firma a continuación el director de la carrera de Computación y encargado principal del laboratorio de software de la Universidad Nacional de Loja, en su calidad de cliente del proyecto:

Ing. Pablo Fernando Ordoñez Ordoñez, Mg. Sc.	 <p>Firmado electrónicamente por: PABLO FERNANDO ORDONEZ ORDONEZ</p>
--	--

Anexo 15: Reporte de plagio en la herramienta Turnitin



Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**
 - Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.
- **Revisado y aprobado por:**
 - Ing. José Oswaldo Guamán, director del Trabajo de Integración Curricular.

Descripción: El presente anexo incorpora el informe de originalidad del Trabajo de Integración Curricular (TIC), generado mediante la herramienta especializada Turnitin. Este reporte se presenta como evidencia para certificar el nivel de originalidad del documento, demostrando así la autoría propia y el rigor académico aplicado en la investigación desarrollada.



unl

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación



Página 1 of 69 - Portada

Identificador de la entrega trn:oid::1:3118555096

Juan Armijos

Plagio TIC - Juan Carlos Armijos

- Ver tesis
- Tesis
- Universidad Nacional de Loja

Detalles del documento

Identificador de la entrega
trn:oid::1:3118555096

Fecha de entrega
18 dic 2024, 9:48 a.m. GMT-5

Fecha de descarga
18 dic 2024, 10:03 a.m. GMT-5

Nombre de archivo
Documento_para_revision_de_plagio_-_TIC.pdf

Tamaño de archivo
1.7 MB

66 Páginas

17,910 Palabras

97,643 Caracteres



Página 1 of 69 - Portada

Identificador de la entrega trn:oid::1:3118555096



3% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Coincidencias menores (menos de 30 palabras)

Exclusiones

- N.º de fuente excluida
- N.º de coincidencias excluidas

Fuentes principales

- 3% Fuentes de Internet
- 0% Publicaciones
- 2% Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alerta de integridad para revisión

- Caracteres reemplazados**
 52 caracteres sospechosos en N.º de páginas
 Las letras son intercambiadas por caracteres similares de otro alfabeto.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.





UNL

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación



Página 3 of 59 - Descripción general de integridad

Identificador de la entrega tm:oid::1:3118555096

Fuentes principales

- 3% Fuentes de Internet
- 0% Publicaciones
- 2% Trabajos entregados (trabajos del estudiante)

Fuentes principales

Las fuentes con el mayor número de coincidencias dentro de la entrega. Las fuentes superpuestas no se mostrarán.

1	Internet	es.scribd.com	1%
2	Internet	es.slideshare.net	0%
3	Internet	core.ac.uk	0%
4	Internet	dspace.unl.edu.ec	0%
5	Internet	repositorio.uesiglo21.edu.ar	0%
6	Internet	repositorio.ucv.edu.pe	0%
7	Trabajos del estudiante	udes-virtual	0%
8	Trabajos del estudiante	Universidad Tecnológica Centroamericana UNITEC	0%



Página 3 of 59 - Descripción general de integridad

Identificador de la entrega tm:oid::1:3118555096



Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Para constancia y validación de la información expuesta, firma a continuación el tutor del Trabajo de Integración Curricular, en su calidad de docente guía en el desarrollo del proyecto:

Ing. José Oswaldo Guamán Quinche, Mg.Sc.	 <p>Firmado electrónicamente por: JOSE OSWALDO GUAMAN QUINCHE</p>
--	--

Anexo 16: Documentos firmados electrónicamente



unl

Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

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

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Software de apoyo especializado en el aprendizaje de programación en C para los laboratorios de la carrera de Computación en la Universidad Nacional de Loja.

- **Elaborado por:**

- Juan Carlos Armijos Sánchez, autor del Trabajo de Integración Curricular.

Descripción: El presente anexo contiene los documentos con firma electrónica correspondientes a los anexos presentados en el Trabajo de Integración Curricular. Debido a que en el cuerpo del trabajo estos se muestran como imágenes, se ha habilitado una carpeta en Google Drive que aloja los archivos originales en formato PDF con sus respectivas firmas electrónicas. Este repositorio permite verificar la autenticidad de cada documento, garantizando así la validez de todos los anexos mediante la comprobación electrónica de sus firmas.

DOCUMENTOS FIRMADOS ELECTRÓNICAMENTE

En virtud de que los anexos que contienen firmas electrónicas se han incorporado en formato de imagen al presente Trabajo de Integración Curricular, se ha habilitado el acceso a una carpeta compartida en Google Drive. En dicha carpeta se encuentran alojados todos los anexos mencionados en su formato original PDF. Esta disposición permite la verificación de la autenticidad de los documentos, dado que cada archivo conserva su correspondiente firma electrónica, garantizando así la integridad y validez de la información presentada.

Link a la carpeta compartida de Google Drive:

<https://n9.cl/documentosfirmadosdigital>

Anexo 17: Certificado de traducción del resumen

Loja, 27 de noviembre de 2024

A quien corresponda. -

De mi consideración,

La presente traducción de español a inglés del resumen de la tesis denominada “Software de Apoyo Especializado en el Aprendizaje de Programación en C para los Laboratorios de la Carrera de Computación en la Universidad Nacional de Loja” de autoría de Juan Carlos Armijos Sánchez, C.I.: 1150329884, fue realizada y revisada por María Gabriela Jiménez con título de Licenciada en Ciencias de la Educación, mención Idioma Inglés, con número de registro en Senescyt 1008-2016-1754550. En consecuencia, esta traducción se considera válida para ser utilizada con fines académicos.

Particular que informo para los fines pertinentes.

Atentamente,

MARIA
GABRIELA
JIMENEZ
CARRION



Digitally signed by
MARIA GABRIELA
JIMENEZ CARRION

María Gabriela Jiménez

Licenciada en Ciencias de la Educación, mención Idioma Inglés.

Registro en Senescyt: 1008-2016-1754550