



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD DE TECNOLOGÍAS INTERACTIVAS

PLATAFORMA WEB DEL COMITÉ CUBANO DE PRUEBAS DE SOFTWARE

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Ibrahim Márquez Sánchez

Tutores: M. Sc. Yadira Ramírez Rodríguez

Dr. C. Yaimí Trujillo Casañola

La Habana, 2024

"El aspecto más triste de la vida en este momento es que la ciencia reúne el conocimiento más rápidamente que la sociedad reúne la sabiduría."

- Isaac Asimov

Dedico este trabajo a mis padres, cuyo amor incondicional, apoyo inquebrantable, dedicación y sacrificio fueron los pilares fundamentales que me permitieron alcanzar este logro.

Agradecimientos

En primer lugar, expreso mi más sincera gratitud a mis padres, Milagros y Adel, por su amor incondicional, apoyo constante y sacrificio inagotable. Su creencia en mí y dedicación incansable han sido la fuerza que necesitaba para conseguir este logro. Sin su ayuda y guía, este trabajo no habría sido posible.

También quiero agradecer profundamente a mi tío Rey y a mi madrina Yakelín por todo el apoyo, consejos y sobre todo por estar siempre para mí.

A mis primos, mis abuelos y a toda mi familia, por su cariño y apoyo incondicional a lo largo de los años.

De forma especial, quiero expresar mi gratitud a mis amistades de la UCI, que se han convertido en una segunda familia. Gracias por todo Brito, Gaby, Nelsito, "Orlian profe", la traicionera Susel y Albita, que me han tenido paciencia y apoyado, además de alegrar mis estancia en la UCI.

Gracias a mi novia Sarah por tenerme paciencia, acompañarme y brindarme su apoyo incondicional a lo largo de mi proceso de tesis.

Agradezco también a mis tutoras Yadira y Yaimí, por guiarme y acompañarme en este proceso.

También agradezco a todos esos profesores que han contribuido a mi formación como profesional.

Finalmente, mi reconocimiento a todas aquellas personas que, de una u otra manera, me han brindado su apoyo a lo largo de este proceso. Su confianza y aliento han sido fundamentales para alcanzar este objetivo. Muchas gracias a

todos.

Declaración de autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los 10 días del mes de diciembre del año 2024.



Ibrahim Márquez Sánchez
Autor



M. Sc. Yadira Ramírez Rodríguez
Tutora



Dr. C. Yaimí Trujillo Casañola
Tutora

La asociación cubana Cuban Testing Board (CuTB), acreditada por el International Software Testing Qualifications Board (ISTQB), requiere la creación de un sitio web para divulgar información sobre sus cursos y certificaciones, cumpliendo con los estándares internacionales establecidos por el ISTQB. El objetivo principal de esta investigación es desarrollar un portal web que facilite la gestión y acceso a la información relacionada con la formación en pruebas de software en Cuba.

El desarrollo del sitio web sigue un enfoque basado en tecnologías web modernas, integrando herramientas de gestión de contenido y seguridad. Se establecen los fundamentos teóricos, se realiza un análisis y diseño de la solución propuesta, y se aplican pruebas funcionales y de aceptación para asegurar el correcto desempeño del sitio. El proceso de desarrollo sigue las pautas de la metodología ágil SCRUM, priorizando la escalabilidad, eficiencia y la satisfacción de los usuarios finales, promoviendo un entorno digital que fomente la innovación y facilite la adopción de soluciones tecnológicas en el sector de las pruebas de software en Cuba.

Palabras clave: Certificación, CuTB, ISTQB, pruebas, software.

Introducción	1
1 Fundamentación teórica.	3
1.1 Evolución de la certificación de software en Cuba	3
1.2 Análisis de soluciones existentes	4
1.2.1 Funcionalidades clave y secciones comunes	4
1.2.2 Funcionalidades diferenciadas a evaluar	5
1.2.3 Conclusiones derivadas del análisis de la tabla	5
1.3 Herramientas y tecnologías para el desarrollo	6
1.3.1 Tecnologías y lenguajes:	6
1.3.2 Herramientas y entornos de desarrollo:	7
1.4 Metodología de desarrollo de software:	8
2 Análisis y diseño de la solución	10
2.1 Descripción de la propuesta de solución	10
2.2 Historias de Usuario	11
2.3 Requisitos no funcionales	13
2.4 Plan de Actividades	14
2.5 Representación arquitectónica de la solución	17
2.6 Diseño de la base de dato	19
3 Implementación y prueba del sistema	22
3.1 Pruebas de rendimiento	22
3.1.1 Herramienta de prueba	23
3.1.2 Resultados de las pruebas	24
3.2 Casos de Pruebas Funcionales	26
3.3 Pruebas de aceptación	27
3.4 Resultado de las pruebas	30
Conclusiones	31

Recomendaciones	32
Referencias bibliográficas	33
Apéndices	35
A Entrevista	36
B Historias de Usuario	37
C Pruebas de Aceptación	48

Índice de figuras

2.1	Esquema general del funcionamiento de la solución (Elaboración propia).	11
2.2	Patrón de Arquitectura en Capas (Elaboración propia).	18
2.3	Diagrama de Clases para el Crud Gestionar Curso (Elaboración propia).	19
2.4	Modelo de datos	20
3.1	Test	25
3.2	Resultado de pruebas	30

Índice de tablas

1.1	Análisis de sistemas homólogos de certificación de pruebas de software	5
2.1	Historia de usuario # 1	12
2.2	Historia de usuario # 2	12
2.3	Historia de usuario # 3	12
2.4	Plan de lanzamiento	14
3.1	Casos de prueba	26
3.2	Prueba de aceptación # 1	28
3.3	Prueba de aceptación # 2	28
3.4	Prueba de aceptación # 3	29
3.5	Prueba de aceptación # 4	29
3.6	Prueba de aceptación # 5	29
B.1	Historia de usuario # 4	37
B.2	Historia de usuario # 5	37
B.3	Historia de usuario # 6	37
B.4	Historia de usuario # 7	38
B.5	Historia de usuario # 8	38
B.6	Historia de usuario # 9	38
B.7	Historia de usuario # 10	39
B.8	Historia de usuario # 11	39
B.9	Historia de usuario # 12	39
B.10	Historia de usuario # 13	40
B.11	Historia de usuario # 14	40
B.12	Historia de usuario # 15	40
B.13	Historia de usuario # 16	40
B.14	Historia de usuario # 17	41
B.15	Historia de usuario # 18	41
B.16	Historia de usuario # 19	41
B.17	Historia de usuario # 20	42

B.18 Historia de usuario # 21	42
B.19 Historia de usuario # 22	42
B.20 Historia de usuario # 23	42
B.21 Historia de usuario # 24	43
B.22 Historia de usuario # 25	43
B.23 Historia de usuario # 26	43
B.24 Historia de usuario # 27	44
B.25 Historia de usuario # 28	44
B.26 Historia de usuario # 29	44
B.27 Historia de usuario # 30	44
B.28 Historia de usuario # 31	45
B.29 Historia de usuario # 32	45
B.30 Historia de usuario # 33	45
B.31 Historia de usuario # 34	46
B.32 Historia de usuario # 35	46
B.33 Historia de usuario # 36	46
B.34 Historia de usuario # 37	46
B.35 Historia de usuario # 38	47
 C.1 Prueba de aceptación # 6	 48
C.2 Prueba de aceptación # 7	48
C.3 Prueba de aceptación # 8	49
C.4 Prueba de aceptación # 9	49
C.5 Prueba de aceptación # 10	49
C.6 Prueba de aceptación # 11	50
C.7 Prueba de aceptación # 12	50
C.8 Prueba de aceptación # 13	50
C.9 Prueba de aceptación # 14	51
C.10 Prueba de aceptación # 15	51
C.11 Prueba de aceptación # 16	51
C.12 Prueba de aceptación # 17	52
C.13 Prueba de aceptación # 18	52
C.14 Prueba de aceptación # 19	53
C.15 Prueba de aceptación # 20	53
C.16 Prueba de aceptación # 21	53
C.17 Prueba de aceptación # 22	54
C.18 Prueba de aceptación # 23	54
C.19 Prueba de aceptación # 24	55

C.20 Prueba de aceptación # 25	55
C.21 Prueba de aceptación # 26	55
C.22 Prueba de aceptación # 27	56
C.23 Prueba de aceptación # 28	56
C.24 Prueba de aceptación # 29	56
C.25 Prueba de aceptación # 30	57
C.26 Prueba de aceptación # 31	57

La tecnología de la información ha desempeñado un papel crucial en el desarrollo de diversos sectores en Cuba, incluida la industria del software. En particular, las certificaciones internacionales como las del International Software Testing Qualifications Board son altamente valoradas en la comunidad de pruebas de software. La asociación cubana CuTB, que gestiona la formación y certificación de probadores en el país, está acreditada por el ISTQB para ofrecer certificaciones internacionales. Sin embargo, para mantener esta acreditación, el ISTQB requiere que CuTB cumpla con una serie de normas, entre ellas la creación de un sitio web oficial donde se divulgue información relevante sobre los cursos y las certificaciones disponibles. El problema principal radica en que CuTB no cuenta actualmente con un sitio web para mostrar la información de los miembros, los cursos, los exámenes, los proveedores y las noticias puesto que esto se realiza por redes sociales. Esto provoca que no se cuente con una vía oficial y centralizada para brindar la información a la comunidad cubana de prueba además de que la documentación del comité cubano de pruebas se maneja con documentos digitales. Con esto se pone en riesgo la acreditación de la junta en las auditorías bienales realizadas por el ISTQB. Estas auditorías son cruciales para evaluar si la asociación continúa cumpliendo con los requisitos establecidos por el ISTQB, y la falta de un sitio web podría comprometer su capacidad para seguir operando como entidad acreditada.

A partir de esta problemática se enuncia como **problema científico**: ¿Cómo gestionar y publicar la información del comité cubano de pruebas de software a los profesionales de las Tecnologías de Información de la Comunicación de forma centralizada?

Se define como **objeto de estudio**: Los Sistemas de Gestión de Información, precisando como **campo de acción**: La gestión y publicación de la información del comité cubano de pruebas.

Para resolver la situación problemática establecida se plantea como **objetivo general**: Desarrollar un sitio web para el comité cubano de pruebas de software para gestionar y publicar la información en correspondencia con los requerimientos del ISTQB.

Objetivos específicos:

- Construir el marco teórico referencial, con énfasis en los sistemas de gestión de la información y el análisis de sistemas homólogos.
- Identificar tecnologías, metodologías y herramientas a utilizar para el desarrollo del sitio web.
- Analizar los requisitos de software para gestionar y publicar la información del comité cubano de pruebas.

- Diseñar el sitio web para gestionar y publicar la información del comité cubano de pruebas.
- Implementar el sitio web para gestionar y publicar la información del comité cubano de pruebas.
- Realizar pruebas de software para verificar y validar el sitio web.

Métodos y técnicas de investigación científica

Métodos teóricos:

- Histórico-lógico: revisar sistemas previos utilizados (sistemas homólogos) al CUTB para identificar tendencias, fortalezas y debilidades.
- Analítico-sintético: separar las funciones del sistema para luego integrarlas en una sola plataforma coherente.
- Modelación: Diseñar diagramas visuales (UML) del sistema para previsualizar cómo funcionará.

Métodos empíricos:

- Observación: observar cómo se llevan a cabo actualmente los procesos en el CUTB para identificar áreas de mejora.
- Entrevista: realizar entrevistas a los miembros de la junta del CUTB para entender los requisitos específicos. Las preguntas realizadas se pueden encontrar en el apéndice A de los anexos.

Técnicas basadas en juicio de experto:

- Criterio de especialistas: pedir a desarrolladores con experiencia que revisen la arquitectura del sistema propuesto para detectar posibles fallos o mejoras.

Estructura del documento:

- **Capítulo 1: Fundamentación teórica.** En este capítulo se analizan los conceptos clave relacionados con el problema en cuestión. Además, se hace un análisis de los sistemas similares y se seleccionan las herramientas, técnicas, tecnologías y metodologías a utilizar para lograr el objetivo general.
- **Capítulo 2: Análisis y diseño de la solución.** Se describe en términos generales el proceso de desarrollo del portal, comenzando por la planificación de los componentes esenciales y la organización del contenido. Para ello se hace uso de los artefactos ingenieriles acordes a la metodología de desarrollo de software seleccionada.
- **Capítulo 3: Implementación y prueba del sistema.** Se lleva a cabo la implementación de las funcionalidades del sistema y se realizan pruebas para asegurar el correcto funcionamiento del sistema y la satisfacción del cliente.

Fundamentación teórica.

La creación de plataformas web especializadas requiere un análisis profundo de los conceptos y tecnologías involucradas, especialmente cuando se trata de cumplir con estándares internacionales, como los establecidos por el ISTQB. Es esencial comprender el estado del arte en el desarrollo de portales web enfocados en la gestión de certificaciones de software a nivel nacional e internacional. Esto permite identificar enfoques exitosos y lecciones aprendidas que puedan aplicarse al caso de estudio.

Además, se evaluarán las tecnologías y herramientas seleccionadas para la implementación del portal web de CuTB, considerando su capacidad para garantizar accesibilidad, escalabilidad y cumplimiento con los requisitos del ISTQB. Este análisis proporcionará un marco teórico sólido para la construcción de una solución eficiente y alineada con las necesidades de la comunidad cubana de pruebas de software.

1.1. Evolución de la certificación de software en Cuba

La evolución de la certificación de pruebas de software en Cuba ha sido impulsada en gran medida por la colaboración de organizaciones como ISTQB, que es la mayor entidad de certificación en el ámbito del software testing a nivel mundial. La ISTQB ofrece un esquema de certificación que valida y desarrolla las habilidades de los profesionales de pruebas de software, con módulos que abarcan desde el nivel de fundamentos hasta especializaciones más avanzadas como Agile y Security Testing. Estas certificaciones, que están alineadas con estándares internacionales, se adaptan constantemente para seguir siendo relevantes y accesibles en todo el mundo, incluida Cuba.

El surgimiento de la CuTB (Cuban Testing Board) ha sido fundamental en la implementación de estas certificaciones dentro de la comunidad cubana, permitiendo a los profesionales de pruebas en el país acceder a certificaciones globales sin tener que depender de instancias internacionales. Este desarrollo ha sido clave para fomentar la calidad en los procesos de software, facilitando la inserción de Cuba en la industria global de pruebas de software. Con más de 500,000 certificaciones emitidas a nivel mundial, la ISTQB permite que los profesionales cubanos no solo se beneficien de mejores prácticas, sino que también puedan integrarse en proyectos internacionales.

La creación de un portal web para el CuTB se convierte en una herramienta crucial para consolidar estos avances. Este sitio reunirá en un solo lugar toda la información necesaria para que los testers cubanos accedan a la certificación, ofreciendo un entorno más organizado y accesible para la comunidad local.

1.2. Análisis de soluciones existentes

Al analizar los sistemas homologados de certificación de pruebas de software seleccionados para este estudio - **ISTQB** (International Software Testing Qualifications Board), **ASTQB** (American Software Testing Qualifications Board), **GTB** (German Testing Board), **SSTQB** (Spanish Software Testing Qualifications Board), y **IST** (Indian Software Testing Board) - se busca identificar las funcionalidades y secciones comunes que caracterizan a estos organismos internacionales. Este análisis es crucial para el Cuban Testing Board, ya que permitirá determinar qué prácticas y estructuras pueden ser adaptadas eficazmente para satisfacer las necesidades específicas del contexto cubano. El objetivo principal es diseñar un portal cubano de certificación de pruebas de software que no solo cumpla con estándares globales, sino que también mejore la accesibilidad y calidad de las certificaciones en Cuba.

1.2.1. Funcionalidades clave y secciones comunes

Los cinco sistemas homologados, incluyendo el ISTQB, ASTQB, GTB, IST y SSTQB, comparten la estructura básica de certificaciones con módulos que van desde los fundamentos hasta especializaciones en pruebas avanzadas. Estos módulos son fundamentales en cada sistema y permiten la actualización constante de habilidades. El CuTB debe replicar esta estructura de manera clara y accesible, con especial énfasis en los módulos de certificación básica y especializaciones relevantes para el contexto cubano.

Una característica que comparten todos estos sistemas es la presencia de materiales educativos, ya sea en forma de guías, webinars, o recursos descargables. Por ejemplo, el ASTQB y el GTB ofrecen un acceso a recursos educativos bien organizados, que ayudan a los profesionales a prepararse para los exámenes. El CuTB también debería incorporar una sección similar, asegurando la disponibilidad de materiales tanto en español como en inglés para facilitar el estudio.

Otra funcionalidad importante es la facilidad para registrarse en los exámenes y acceder al soporte técnico. ISTQB, ASTQB y IST brindan interfaces muy intuitivas para el registro de certificaciones, lo que reduce las barreras para los usuarios. En el CuTB, esto puede implementarse con formularios accesibles en línea, que también brinden información detallada sobre fechas, costos y opciones de pago, acompañado de soporte técnico ágil.

El GTB y el SSTQB promueven un sentido de comunidad a través de foros de discusión o secciones de noticias donde se abordan las tendencias del software testing. Para el CuTB, crear una comunidad local sería invaluable para fomentar el crecimiento del sector en Cuba, conectando a profesionales locales e internacionales.

1.2.2. Funcionalidades diferenciadas a evaluar

Algunos sistemas, como el SSTQB, ofrecen recursos limitados o menos completos comparados con el ASTQB o GTB. Para el CuTB, es fundamental garantizar una amplia base de recursos y asegurarse de que el contenido sea accesible en diferentes dispositivos y plataformas.

En países como India y Alemania, los sistemas de certificación locales (por ejemplo, IST y GTB) ofrecen módulos específicos ajustados a las necesidades locales, algo que el CuTB debería evaluar para que la certificación en Cuba responda a las particularidades del mercado laboral cubano.

A continuación se muestra un resumen de la comparación entre los sistemas analizados:

Tabla 1.1. Análisis de sistemas homólogos de certificación de pruebas de software

Sistema	Facilidad de navegación	Recursos educativos	Soporte técnico	Idiomas	Comunidad	Costo y accesibilidad
ISTQB	Muy intuitiva	Extensos, guías y webinars	Soporte técnico y foros	Inglés	Internacional	Varía por región
ASTQB	Intuitiva, fácil registro online	Muy completos, webinars regulares	Soporte técnico, comunidad local	Inglés	Fuerte presencia en EE.UU.	Costo accesible
GTB	Intuitiva, enfoque local en Alemania	Múltiples recursos, enfoque ágil	Atención por correo, webinars	Inglés, Alemán	Comunidad activa	Varía por tipo de examen
SSTQB	Funcional, menos recursos disponibles	Recursos limitados	Soporte por correo, menos inmediato	Español, Inglés	Comunidad pequeña	Costos moderados
IST	Registro directo, adecuado al mercado local	Buen nivel de recursos adaptados	Webinars y foros locales	Inglés	Comunidad creciente	Costos accesibles

1.2.3. Conclusiones derivadas del análisis de la tabla

Todos los sistemas homólogos tienen interfaces intuitivas que permiten una **fácil navegación** y registro de exámenes. El CuTB debería tomar como modelo el enfoque simplificado del ASTQB, donde el registro y la obtención de información es ágil y clara, sin barreras innecesarias.

El ASTQB y el GTB destacan por ofrecer una amplia gama de **recursos educativos**, lo que les permite garantizar que los candidatos estén bien preparados para las certificaciones. En contraste, el SSTQB tiene una oferta más limitada en este sentido. El CuTB debería priorizar la creación de una base de recursos educativos extensa para facilitar el aprendizaje de los testers cubanos.

La atención al cliente es clave en cualquier sistema de certificación. El ISTQB y el GTB ofrecen un **soporte técnico eficiente**, mientras que el SSTQB se queda atrás en este aspecto. El CuTB deberá implementar un

sistema de atención inmediato y accesible para resolver problemas técnicos y dudas, con soporte tanto en español como en inglés.

El ASTQB y el IST son conocidos por mantener un **costo accesible** para sus exámenes, algo crucial para facilitar la certificación de profesionales. El CuTB debe asegurarse de que el costo de las certificaciones sea viable para los profesionales cubanos, especialmente considerando las restricciones económicas del país.

En resumen, el CuTB puede beneficiarse enormemente al adoptar las mejores prácticas de estos sistemas homologados, priorizando la accesibilidad, la amplitud de los recursos educativos y la atención al cliente, mientras adapta el contenido y los exámenes a las necesidades del mercado cubano de pruebas de software.

1.3. Herramientas y tecnologías para el desarrollo

En este epígrafe, se describen las principales tecnologías adoptadas para la construcción del sistema, abarcando desde frameworks de desarrollo web hasta bases de datos y herramientas de colaboración. Además, se analiza cómo estas tecnologías se integran para ofrecer una solución robusta y moderna.

1.3.1. Tecnologías y lenguajes:

JavaScript v.ES13, comúnmente abreviado como JS, es un lenguaje de programación del tipo interpretado. Está basado en el estándar ECMAScript y, a pesar de ser débilmente tipado y dinámico, también se define como un lenguaje orientado a objetos. Se ha consolidado como uno de los lenguajes de programación más utilizados en el desarrollo web por su versatilidad y capacidad de funcionar tanto en el frontend como en el backend. Su capacidad para interactuar con APIs y generar aplicaciones dinámicas ha sido crucial para este proyecto, proporcionando una experiencia fluida y en tiempo real para los usuarios (Luna, 2019).

HTML 5, es un lenguaje de marcado, que desde hace varios años se considera el estándar para la creación de sitios web, sobre todo a la hora de crear interfaces visuales. Sus siglas, por su nombre en inglés HyperText Markup Language, hacen referencia al tipo de lenguaje ante el cual te encuentras, un estándar que utiliza etiquetas de apertura y cierre, orientado a almacenar información y a mostrarla (Aguirre, 2021).

CSS 3, un lenguaje de hojas de estilo que se utiliza para describir la presentación de un documento escrito en HTML o XML (incluidos los dialectos XML como SVG, MathML o XHTML). CSS describe cómo deben representarse los elementos en la pantalla, en papel, en el habla o en otros medios (CSS 2024).

UML, ML es una notación de modelado visual, que utiliza diagramas para mostrar distintos aspectos de un sistema. Si bien muchos destacan que UML es apto para modelar cualquier sistema, su mayor difusión y sus principales virtudes se advierten en el campo de los sistemas de software. Esto no obstaculiza para que muchos profesionales intenten usar UML en situaciones diversas, haciendo uso de esa máxima que dice que "cuando la única herramienta que conocemos es el martillo, aun los tornillos nos parecen clavos". (Fontela,

2012).

Node.js v.20.9.0, es un entorno que trabaja en tiempo de ejecución, de código abierto, multi-plataforma, que permite a los desarrolladores crear toda clase de herramientas de lado servidor y aplicaciones en JavaScript. La ejecución en tiempo real está pensada para usarse fuera del contexto de un explorador web (es decir, ejecutarse directamente en una computadora o sistema operativo de servidor). Como tal, el entorno omite las APIs de JavaScript específicas del explorador web y añade soporte para APIs de sistema operativo más tradicionales que incluyen HTTP y bibliotecas de sistemas de archivos (*Introducción a Express/Node - Aprende desarrollo web* | MDN 2024).

Next.js v.14.2.14 es un framework React para crear aplicaciones web de pila completa. Se utilizan componentes React para crear interfaces de usuario y Next.js para funciones y optimizaciones adicionales. En segundo plano, Next.js también abstrae y configura automáticamente las herramientas necesarias para React, como la agrupación, la compilación y más. Esto te permite concentrarte en crear tu aplicación en lugar de perder tiempo con la configuración. Ya sea que sea un desarrollador individual o forme parte de un equipo más grande, Next.js puede ayudarte a crear aplicaciones React interactivas, dinámicas y rápidas. (*Vercel | Next.js* 2024).

Tailwind CSS, es un framework de estilos que funciona escaneando todos sus archivos HTML, componentes JavaScript y cualquier otra plantilla en busca de nombres de clase, generando los estilos correspondientes y luego escribiéndolos en un archivo CSS estático. Es rápido, flexible y confiable, con tiempo de ejecución cero (*Tailwind - Tailwind CSS* 2024).

Como sistema de gestión de bases de datos, **PgAdmin 4** puesto que permite gestionar bases de datos de **PostgreSQL** que es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (Ordóñez; Ríos y Castillo, 2017).

1.3.2. Herramientas y entornos de desarrollo:

Para el entorno de desarrollo se ha seleccionado **Visual Studio Code v.1.95.3**, un editor de código fuente desarrollado por Microsoft, diseñado para ser ligero, rápido y altamente personalizable. A diferencia de otros editores de código, como los IDE completos (Entornos de Desarrollo Integrados), VS Code se enfoca en ofrecer una experiencia ágil para la escritura y edición de código, sin sacrificar características avanzadas (*Visual Studio Code* 2024).

Para el control de versiones y la colaboración en equipo, se ha utilizado **GitHub v.2.39.2.windows.1**, una plataforma Web de desarrollo colaborativo basada en Git. GitHub ofrece toda la funcionalidad de Git e integra diversas herramientas de control de acceso, colaboración, trazabilidad, gestión de tareas y control de proyectos (Lopez-Pellicer; Béjar; Latre; Noguera-Iso y Zarazaga-Soria, 2015).

Postman, Postman es una popular herramienta utilizada para probar APIs, permitiendo a los desarrolladores enviar peticiones a servicios web y ver respuestas. Exploraremos cómo Postman puede ser una herramienta esencial para probar APIs eficientemente (Izertis, 2024).

Visual Paradigm v.8.0, es una herramienta UML CASE compatible con UML 2, SysML y la notación de modelado de procesos empresariales del grupo de gestión de objetos. Además del soporte de modelado, proporciona capacidades de ingeniería de código y generación de informes, incluida la generación de código (*Visual Paradigm | Edutools* 2024) .

1.4. Metodología de desarrollo de software:

La metodología ágil **SCRUM** ha sido utilizada como marco de trabajo para gestionar el desarrollo del portal web del CuTB. SCRUM permite organizar equipos y procesos para garantizar un flujo continuo de entregas en cada iteración o sprint, adaptándose a las necesidades del proyecto.

En SCRUM, el proceso comienza con la **planificación del sprint**, en la cual el equipo revisa el backlog del producto y selecciona las tareas que se abordarán durante la iteración. El backlog incluye historias de usuario, que son una herramienta esencial para definir los requisitos de una funcionalidad específica desde el punto de vista del usuario final. En el caso del portal CuTB, las historias de usuario se centraron en las necesidades del cliente, como el despliegue de la información de cursos de certificación y noticias del ISTQB, así como la inscripción de usuarios. Estas historias ayudaron a priorizar el trabajo, asegurando que los requerimientos más críticos fueran abordados primero.

La **fase de desarrollo del sprint** es donde se ejecutan las tareas seleccionadas. Durante esta fase, el equipo trabaja colaborativamente, utilizando un enfoque de desarrollo incremental, lo que permite entregar funcionalidades en cada sprint. En el contexto del portal CuTB, esto incluyó el desarrollo de las secciones para la información de los cursos, la creación de formularios de inscripción y el manejo del contenido que debía estar disponible para la comunidad de pruebas de software en Cuba. Este enfoque ágil permitió recibir retroalimentación rápida y realizar ajustes constantes.

Luego, se lleva a cabo la **revisión del sprint**, donde se presenta el incremento del producto al cliente y otros interesados para recibir comentarios. En el desarrollo del portal del CuTB, estas revisiones permitieron al equipo ajustar elementos como la interfaz de usuario y las funcionalidades del backend en base al feedback, lo que contribuyó a mejorar la experiencia general del usuario y cumplir con las expectativas del ISTQB.

Uno de los artefactos fundamentales en SCRUM son las **historias de usuario**. Estas descripciones detallan, desde la perspectiva del usuario, los resultados esperados de una funcionalidad específica. Para el portal CuTB, las historias de usuario abarcaron tareas como la visualización de información sobre certificaciones, la inscripción en cursos, y la gestión de usuarios, todas ellas alineadas con los objetivos del cliente de mejorar la accesibilidad a la certificación en Cuba.

Otro artefacto importante son los **diagramas de arquitectura del sistema**, los cuales se usaron para planificar la infraestructura técnica del portal, asegurando que la plataforma fuese escalable y segura. Los **maquetados visuales** también jugaron un papel clave, ayudando al equipo a definir la estructura visual del portal, alineando la interfaz de usuario con los estándares de diseño del cliente.

Finalmente, el **plan de actividades** sirvió como una guía detallada para la ejecución del proyecto, describiendo las tareas de cada sprint, los responsables de su ejecución y el tiempo estimado para su culminación, asegurando así una gestión efectiva del tiempo y los recursos en el proyecto del CuTB.

Conclusiones Parciales

Del análisis realizado en este capítulo se concluye:

- La creación del portal web para el CuTB es crucial para consolidar los avances en la certificación de pruebas de software en Cuba y garantizar el cumplimiento de los estándares del ISTQB.
- El análisis de sistemas homologados permitió identificar prácticas clave que mejorarán la accesibilidad y calidad de las certificaciones en el contexto cubano.
- Las tecnologías seleccionadas para el desarrollo del portal, como Next.js, Express y PostgreSQL como gestor de bases de datos, aseguran una plataforma robusta, escalable y adaptable a las necesidades locales.
- Se seleccionó SCRUM como metodología de desarrollo puesto que permite organizar equipos y procesos para garantizar un flujo continuo de entregas en cada iteración o sprint, adaptándose a las necesidades del proyecto.

Análisis y diseño de la solución

La creación del portal web de CuTB implica un diseño cuidadoso enfocado en la estructura y organización de la información, aspectos que resultan fundamentales para cumplir con los requisitos de visibilidad y accesibilidad del ISTQB. La solución propuesta se fundamenta en la metodología SCRUM, lo cual permite un desarrollo ágil y adaptativo, organizado en fases de trabajo secuenciales.

En este capítulo, se describe en términos generales el proceso de desarrollo del portal, comenzando por la planificación de los componentes esenciales y la organización del contenido. La solución se organiza en fases clave, que permiten el avance gradual de cada sección y funcionalidad. Esta estructura de desarrollo busca optimizar tanto la disposición de la información relevante sobre certificaciones y cursos como la experiencia de los usuarios en el sitio web.

2.1. Descripción de la propuesta de solución

Para responder a las exigencias del ISTQB y facilitar el acceso a la información de certificación en Cuba, se ha realizado un diseño previo de la estructura del portal web de la Junta Cubana de Pruebas de Software, alineado con los requisitos y normativas del ISTQB. Este diseño ofrece acceso centralizado a información clave sobre cursos, exámenes, proveedores y noticias, entre otras necesidades del cliente, asegurando que el contenido esté siempre actualizado y conforme a los estándares de la asociación. La estructura del portal se organiza en dos secciones principales, garantizando el fácil acceso a la información para los usuarios y permitiendo una gestión eficiente del contenido del sitio.

El sistema consta de dos roles: administrador y gestor de actividades, los cuales acceden al panel administrativo a través de un inicio de sesión (login). Estos usuarios podrán actualizar y gestionar el contenido según los permisos asociados a cada rol, sin necesidad de conocimientos técnicos.

Finalmente, la propuesta se complementa con un esquema general de las funcionalidades principales, que

ilustra la navegación y conexión entre las secciones del portal. Este diagrama facilita la comprensión de la estructura y muestra cómo el portal integra las áreas clave de información y servicios para ofrecer una experiencia de usuario cohesiva y alineada con los objetivos de la CuTB y los estándares del ISTQB. Ver Figura 2.1.

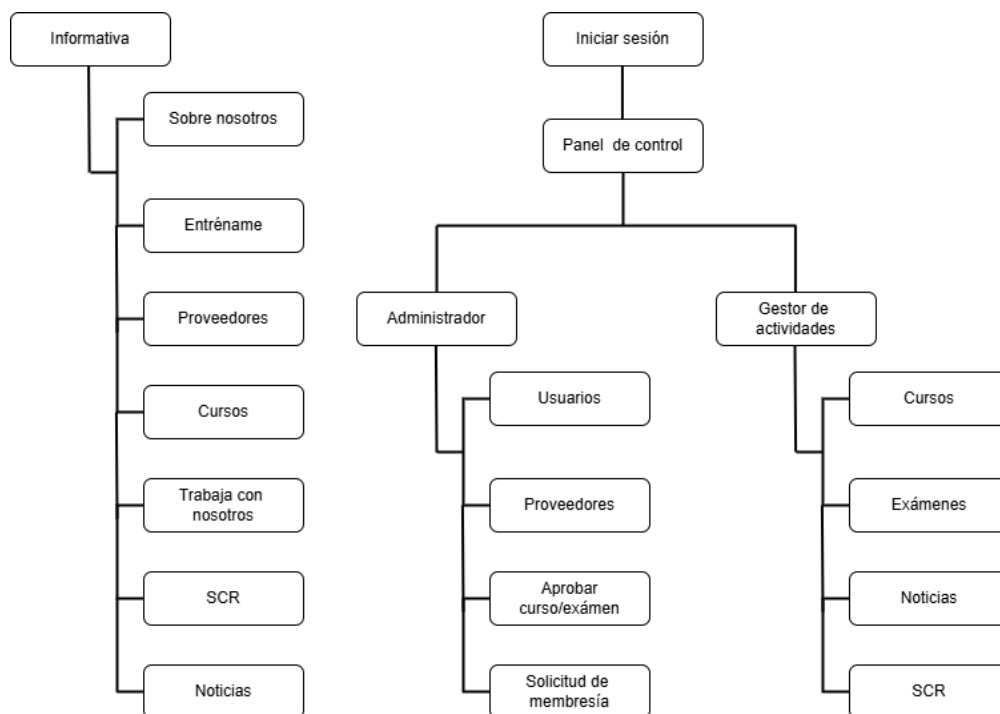


Figura 2.1. Esquema general del funcionamiento de la solución (Elaboración propia).

2.2. Historias de Usuario

Las historias de usuario son una notación concisa para expresar requisitos que se emplean cada vez más en la ingeniería de requisitos ágil y en el desarrollo ágil. De hecho, se han convertido en la notación de requisitos más utilizada en proyectos ágiles y su adopción ha sido fomentada por su uso en numerosos libros sobre desarrollo ágil. La literatura sugiere los mismos tres componentes básicos de una HU (Durán Rodríguez, s.f.)

Las historias de usuario se plantean de forma sencilla y breve, con el objetivo de ser fácilmente comprendidas y realizables en iteraciones cortas, generalmente de una a dos semanas. Esto facilita que el equipo técnico mantenga un enfoque claro en la implementación de funcionalidades específicas que aporten valor al sistema. A continuación, se presentan las historias de usuario para el portal web de CuTB, que abarcan tanto requisitos funcionales como aspectos esenciales para la experiencia del usuario final.

A continuación se muestran las historias de usuario definidas:

Tabla 2.1. Historia de usuario # 1

Historia de usuario	
Número: 1	Nombre: Mostrar información general
Usuario: Visitante	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 2
Programador responsable: Ibrahim Márquez Sánchez	
Descripción: Mostrar información relevante sobre el ISTQB, CuTB y las certificaciones disponibles.	
Observaciones: Verificar que el contenido se actualice fácilmente desde el panel administrativo. Garantizar diseño responsivo.	

Tabla 2.2. Historia de usuario # 2

Historia de usuario	
Número: 2	Nombre: Mostrar proveedores
Usuario: Visitante	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.4	Iteración asignada: 2
Programador responsable: Ibrahim Márquez Sánchez	
Descripción: Permitir a los usuarios visualizar los proveedores acreditados, con un botón para ver detalles específicos.	
Observaciones: Diseñar un modal emergente para ver los detalles.	

Tabla 2.3. Historia de usuario # 3

Historia de usuario	
Número: 3	Nombre: Mostrar cursos
Usuario: Visitante	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.4	Iteración asignada: 2
Programador responsable: Ibrahim Márquez Sánchez	
Descripción: Listar los cursos disponibles, con la opción de copiar el correo del proveedor para solicitar más información.	
Observaciones: Implementar una funcionalidad que facilite copiar el correo de contacto	

El resto de las historias de usuario se muestran en el apéndice B de los anexos.

2.3. Requisitos no funcionales

Los requisitos no funcionales son características que definen la calidad y el rendimiento del sistema, y son cruciales para garantizar una experiencia de usuario satisfactoria y la eficacia operativa del portal web de CuTB. Estos requisitos abarcan aspectos de usabilidad, rendimiento, seguridad, accesibilidad y soporte.

Requisitos de usabilidad y apariencia:

RNF1: La plataforma debe tener una interfaz de usuario que permita realizar las tareas principales (como acceder a cursos, ver certificaciones y enviar solicitudes) de manera rápida y sin complicaciones, con una navegación estructurada y lógica que guíe al usuario de forma clara y eficiente.

RNF2: La plataforma debe ser accesible y funcional en dispositivos móviles, tabletas y computadoras.

RNF3 Sistema operativo: Implementar teclas de atajo para facilitar la navegación.

Requisitos de hardware:

RNF4: Requisitos mínimos del cliente:

- Procesador: 1 GHz o superior.
- RAM: 2 GB o más.
- Espacio en Disco: 100 MB disponibles.

RNF5: Requisitos del servidor:

- Procesador: 4 núcleos a 2.5 GHz o superior.
- RAM: 8 GB o más.
- Espacio en Disco: 500 GB disponibles.
- Conectividad: Conexión a internet de alta velocidad (mínimo 10 Mbps).

Requisitos no funcionales de software:

RNF6: La plataforma debe ser compatible con los navegadores más populares: Google Chrome, Mozilla Firefox, Microsoft Edge y Safari.

RNF7: Compatible con los sistemas operativos Windows 10 y superiores, macOS 10.14 y superiores, y distribuciones recientes de Linux (e.g., Ubuntu 20.04 LTS y superiores).

RNF8: Utilizar una base de datos robusta y escalable como PostgreSQL.

RNF9: Implementación utilizando frameworks modernos y lenguajes de programación robustos como Next.js y Node.js.

Pruebas y calidad:

RNF10: Realizar pruebas de funcionalidad y aceptación por parte del cliente.

Accesibilidad:

RNF11: La plataforma debe ser compatible con lectores de pantalla para usuarios con discapacidades visuales.

RNF12: Usar combinaciones de colores que sean accesibles para personas con discapacidades visuales.

RNF13: Cumplir con los estándares de accesibilidad WCAG 2.1.

Seguridad:

RNF14: Implementar autenticación robusta y autorización basada en roles (RBAC).

RNF15: Todos los datos sensibles deben ser cifrados en tránsito (TLS/SSL) y en reposo.

RNF16: Implementar medidas de seguridad para proteger contra ataques comunes como SQL Injection, Cross-Site Scripting (XSS), y Cross-Site Request Forgery (CSRF).

2.4. Plan de Actividades

El plan de actividades es fundamental para guiar el desarrollo del portal web de CuTB, asegurando que cada funcionalidad se desarrolle de manera organizada y con un enfoque centrado en el usuario final. A partir del análisis exhaustivo de los requisitos y tomando en cuenta la distribución de tareas en ciclos iterativos, se ha estructurado un plan detallado bajo la metodología Scrum. Este enfoque permite adaptar las actividades a cambios en los requisitos o a nuevos descubrimientos, promoviendo la colaboración continua entre el equipo y el cliente.

El objetivo principal de este plan es optimizar el desarrollo de las funcionalidades clave del sistema, garantizando tanto la eficiencia en cada etapa como la calidad del producto final. La planificación detallada de cada sprint y la asignación clara de tareas permiten que el equipo mantenga un enfoque ágil y orientado a cumplir con los objetivos del proyecto.

Tabla 2.4. Plan de lanzamiento

Orden	Descripción	Tiempo en semanas
Sprint 1		
1	Crear el backlog del producto y definir historias de usuario.	0.5
2	Establecer el entorno de desarrollo y configurar herramientas necesarias.	0.5
Sprint 2		
3	Realizar la planificación del sprint.	0.2
4	Mostrar información general.	0.3
5	Mostrar listado de proveedores.	0.4
6	Mostrar listado de cursos.	0.4
7	Descargar información de exámenes de certificación (PDF).	0.3
8	Formulario de "Trabaja con Nosotros".	0.3
9	Mostrar noticias.	0.4
10	Visualizar usuarios certificados.	0.4
11	Filtrar usuarios certificados.	0.4
Continúa en la siguiente página		

Tabla 2.4. (continuación de la página anterior)

Orden	Descripción	Tiempo en semanas
12	Buscar usuarios certificados .	0.3
13	Realizar pruebas unitarias y correcciones.	0.5
Sprint 3		
14	Realizar la planificación del sprint.	0.2
15	Iniciar sesión.	0.2
16	Cerrar sesión.	0.1
17	Regresar al panel de administración.	0.1
18	Realizar pruebas unitarias y correcciones.	0.5
Sprint 4		
19	Realizar la planificación del sprint.	0.2
20	Listar Usuarios.	0.2
21	Añadir Usuario.	0.3
22	Editar Usuario.	0.3
23	Eliminar Usuario.	0.2
24	Realizar pruebas unitarias y correcciones.	0.5
Sprint 5		
25	Realizar la planificación del sprint.	0.2
26	Listar Proveedores.	0.3
27	Añadir Proveedor.	0.3
28	Editar Proveedor.	0.3
29	Eliminar Proveedor.	0.2
30	Aceptar solicitud de membresía o proveedor.	0.3
31	Denegar solicitud de membresía o proveedor.	0.3
32	Realizar pruebas unitarias y correcciones.	0.5
Sprint 6		
33	Realizar la planificación del sprint.	0.2
34	Listar Noticias.	0.3
35	Añadir Noticia.	0.3
36	Editar Noticia.	0.3
37	Eliminar Noticia.	0.2
38	Listar Cursos.	0.3
39	Añadir Curso.	0.3
40	Editar Curso.	0.3
41	Eliminar Curso.	0.2
42	Listar Exámenes.	0.3
Continúa en la siguiente página		

Tabla 2.4. (continuación de la página anterior)

Orden	Descripción	Tiempo en semanas
43	Añadir Examen.	0.3
44	Editar Examen.	0.3
45	Eliminar Examen.	0.2
46	Realizar pruebas unitarias y correcciones.	0.5
Sprint 7		
47	Realizar la planificación del sprint.	0.2
48	Listar Usuarios Certificados.	0.2
49	Añadir Usuario Certificado.	0.3
50	Editar Usuario Certificado.	0.3
51	Eliminar Usuario Certificado.	0.2
52	Realizar pruebas unitarias y correcciones.	0.5
Sprint 8		
53	Realizar la planificación del sprint.	0.1
54	Realizar pruebas de integración del sistema.	0.3
55	Realizar pruebas de aceptación y correcciones finales.	0.5
56	Preparar la documentación técnica y de usuario.	0.3
57	Finalizar el proyecto y entregar el producto.	0.2

Se estima que el proyecto finalizará en aproximadamente **13 semanas**, distribuidas en **8 sprints**. Dado que las tareas son incrementales y dependen unas de otras, no es posible realizarlas simultáneamente.

La estimación de tiempo para cada actividad se realizó considerando factores como la complejidad de las funcionalidades, el número de tareas involucradas, y el tiempo promedio necesario para completarlas. Además, se asignó un tiempo específico al inicio de cada sprint para la planificación de las actividades y al final para pruebas unitarias y correcciones, asegurando un desarrollo iterativo y de calidad.

Es importante señalar que estas estimaciones son aproximadas y pueden variar debido a factores como la disponibilidad de recursos, la experiencia del equipo y posibles imprevistos durante la ejecución. Por lo tanto, se implementará un monitoreo constante del progreso del proyecto, permitiendo ajustes necesarios para cumplir con los objetivos y garantizar la entrega exitosa del producto final.

2.5. Representación arquitectónica de la solución

La arquitectura de software, en un sentido estricto, se define como el conjunto de estructuras que componen el sistema, lo que incluye elementos de software, las relaciones entre los mismos y las propiedades tanto de los elementos como de sus relaciones Ardis; Budgen; Hislop; Offutt; Sebern y Visser, 2015. En palabras simples, la arquitectura de software son patrones o lineamientos que ayudan a la construcción de un programa (aplicación). Estos patrones permiten tener una guía para los desarrolladores, analistas y todos los cargos relacionados, para lograr cumplir con los requerimientos de la aplicación.

Dado que la solución se desarrolla como una plataforma web, la arquitectura de la solución se ajusta a una arquitectura en capas. A continuación, se detallan las capas de la arquitectura y las tecnologías utilizadas en cada una:

Capa de presentación: Esta capa es la encargada de visualizar en su interfaz gráfica los elementos de la plataforma web, permitiendo la interacción del usuario con ellos. Se utiliza Next.js, un framework basado en React, que permite la creación de interfaces de usuario interactivas y dinámicas. Además, se emplea Tailwind CSS para el diseño, facilitando la aplicación de estilos de forma eficiente y coherente en toda la plataforma.

Capa de negocio: En esta capa se maneja la lógica de negocio de la plataforma. Se usa Node.js con el framework Express para gestionar las solicitudes del servidor. Esta capa incluye funcionalidades esenciales como la autenticación de usuarios, gestión de sesiones y procesamiento de datos. La lógica de negocio se implementa en controladores y servicios, encargados de validar y procesar la información recibida, así como de interactuar con la capa de datos.

Capa de Comunicación: Se utiliza Axios para manejar la comunicación entre el cliente y el servidor, facilitando las solicitudes HTTP y la integración con servicios externos. Esto proporciona una interfaz consistente para las operaciones de la plataforma y simplifica la gestión de las solicitudes asíncronas.

Capa de datos: La capa de datos es responsable de la gestión y almacenamiento de la información de la plataforma. Se utiliza una base de datos relacional MySQL para almacenar datos estructurados de manera eficiente. Para interactuar con la base de datos, se emplea el ORM Sequelize, que permite manejar las operaciones CRUD de manera más sencilla y con menos errores.

El flujo de la arquitectura está diseñado para gestionar la interacción entre el cliente, las vistas de la aplicación, la lógica de negocio y la base de datos de manera eficiente. El flujo comienza cuando el usuario realiza una solicitud inicial desde el navegador. Esta solicitud se dirige al servidor de aplicaciones, que sirve como punto de entrada para procesar las vistas y gestionar las peticiones. El servidor de aplicaciones de Next.js

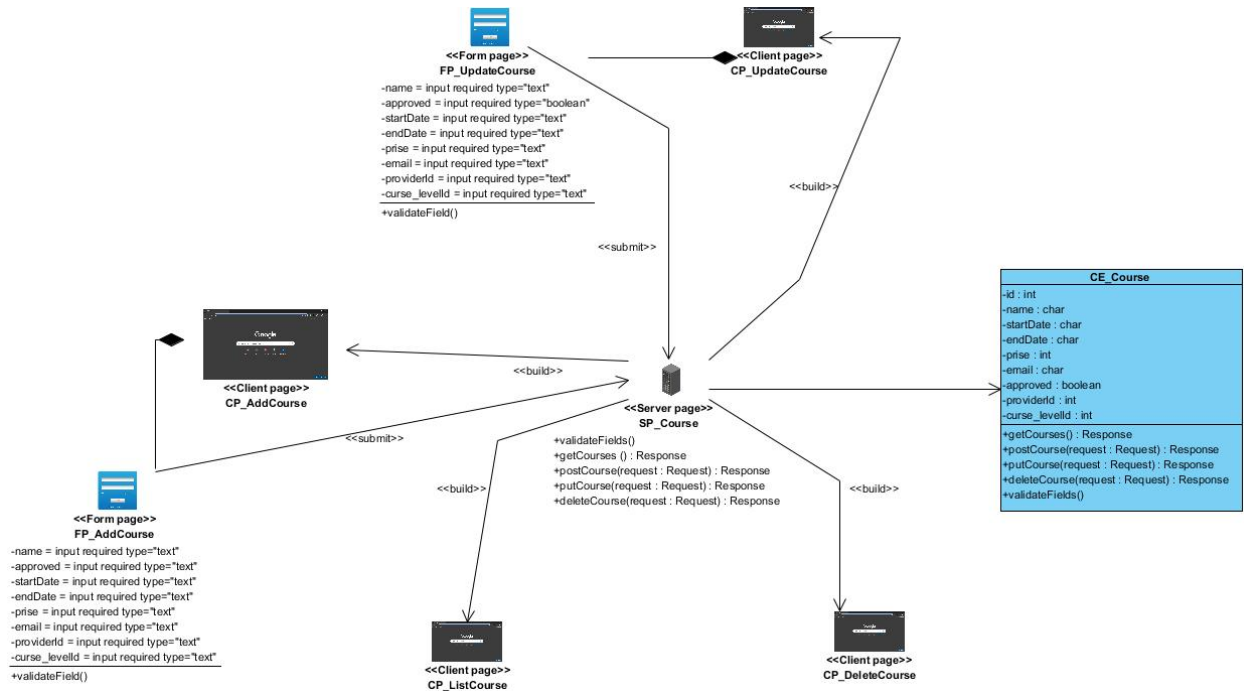


Figura 2.3. Diagrama de Clases para el Crud Gestionar Curso (Elaboración propia).

2.6. Diseño de la base de dato

El modelo entidad-relación, en su forma más simple, implica identificar los asuntos de importancia dentro de una organización (entidades), las propiedades de esos asuntos (atributos) y cómo se relacionan entre sí (relación). Pero esto tiene valor solamente dentro del contexto de lo que se realiza en la empresa y en la forma de actuar de estas funciones de gestión sobre el modelo de información. El modelo de datos tiene como objetivos proporcionar un modelo preciso de las necesidades de información de la organización, que actuará como un marco de trabajo para el desarrollo de sistemas nuevos o mejorados y proporcionar un modelo independiente de cualquier almacenamiento de datos y método de acceso, que permita tomar decisiones objetivas de las técnicas de implementación y la coexistencia con sistemas antiguos (Barker, 1994).

Un diagrama de base de datos es una herramienta esencial para representar visualmente la estructura de una base de datos y las relaciones entre sus diferentes entidades. Este tipo de diagrama facilita la comprensión de cómo se organizan las tablas y cómo interactúan entre sí dentro del sistema, permitiendo una mejor planificación y optimización del diseño de la base de datos (*Modelado de datos* 2024). En el contexto del proyecto del portal web de CuTB, un diagrama de base de datos ayuda a identificar y estructurar eficientemente las tablas necesarias para gestionar cursos, certificaciones, usuarios y otras funcionalidades clave.

A continuación, se presenta un ejemplo del diagrama de base de datos para este proyecto, ilustrando las

interrelaciones entre las entidades principales. Ver Figura 2.4:

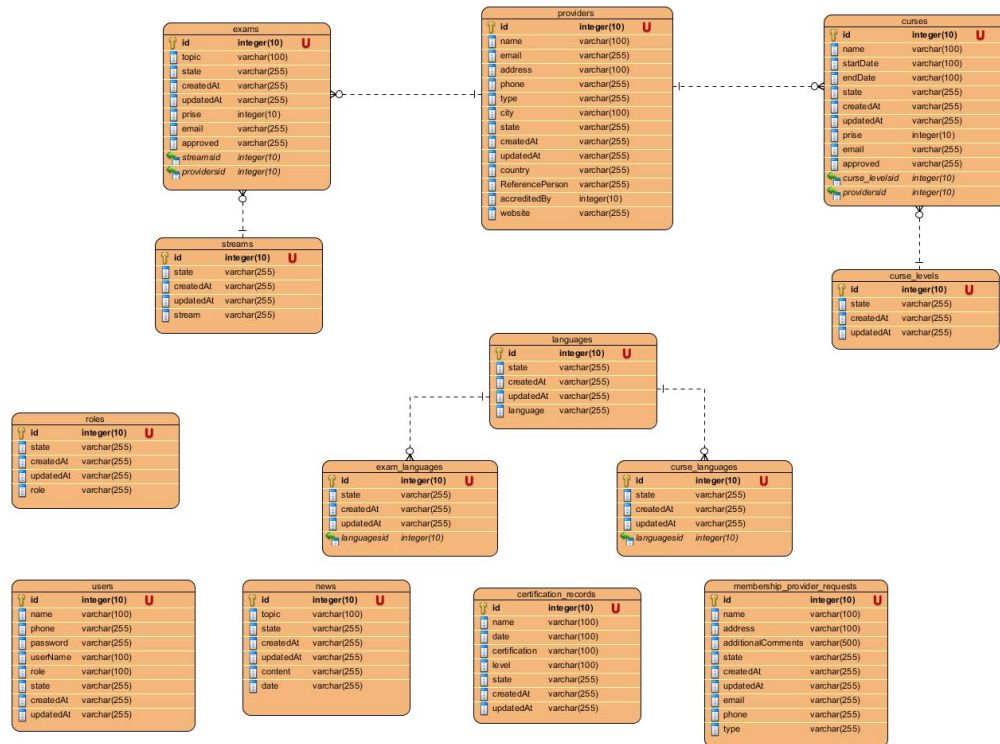


Figura 2.4. Modelo de datos

Conclusiones del capítulo

En este segundo capítulo, se ha trazado el camino para desarrollar la solución al problema científico. A continuación, se destacan las conclusiones más relevantes:

- Se realizó un análisis para definir la propuesta de solución.
- Se definieron 38 historias de usuario que fueron clave para establecer los requisitos del sistema, ya que ofrecen una descripción clara de las funcionalidades que debe tener el sistema para cumplir con las expectativas del cliente.
- Se planificaron las iteraciones de trabajo en función del cronograma de implementación, utilizando como base las historias de usuario previamente definidas.
- El uso de una arquitectura por capas permitió una clara separación de responsabilidades, facilitando la gestión y escalabilidad del proyecto.

Implementación y prueba del sistema

El proceso de implementación del portal web de CuTB se basa en el análisis y diseño establecidos en la propuesta de solución. Durante el desarrollo, se priorizan tanto los requisitos funcionales como los no funcionales para garantizar un producto eficiente y alineado con las necesidades del cliente. Una vez finalizado el sistema, se realizarán las pruebas necesarias para validar su funcionamiento y asegurar que los resultados cumplan con las expectativas planteadas.

3.1. Pruebas de rendimiento

Durante el desarrollo del portal web de CuTB, se realizaron pruebas para evaluar el rendimiento del sistema en términos de tamaño de las rutas y carga inicial de los recursos JavaScript. Estas métricas son esenciales para garantizar una experiencia de usuario eficiente y una carga rápida de las páginas.

En la imagen se detalla el tamaño y el peso del JavaScript inicial de las diferentes rutas del portal:

- **Rutas principales de usuario:** las rutas públicas, como /, /about, /certifyMe, /news, /trainMe, y /trainMe/courses, presentan un tamaño promedio de 324 B y una carga inicial de JavaScript de 227 kB. Estos valores son óptimos para el tipo de contenido estático que representan.
- **Rutas administrativas:** las rutas bajo el prefijo /admin, que incluyen secciones como /admin/users, /admin/providers y /admin/courses, tienen tamaños de entre 502 B y 2.48 kB, con una carga inicial de JavaScript de 231 kB en promedio. Esto refleja un diseño más pesado, pero aceptable, para funcionalidades con mayor interacción dinámica.
- **Ruta de autenticación:** la ruta /auth/login, que gestiona el acceso al portal, es la más pesada, con un tamaño de 5.01 kB y una carga inicial de JavaScript de 183 kB. Esto podría estar relacionado con la inclusión de bibliotecas externas o componentes adicionales necesarios para el proceso de autenticación.

- **Otras rutas relevantes:** la ruta /workWithUs, que permite la gestión de colaboraciones, tiene un tamaño de 4.03 kB y una carga inicial de 179 kB, siendo una de las más pesadas entre las rutas de contenido.

Optimización y Middleware: El middleware utilizado en el portal tiene un peso de **27.5 kB**, representando un componente clave en la gestión de solicitudes. Adicionalmente, los archivos JavaScript compartidos por todas las rutas suman un total de **87.2 kB**, con dos archivos principales de **31.6 kB** y **53.7 kB** respectivamente, optimizados mediante técnicas de preprocesamiento y división de código.

Estas pruebas de rendimiento permitieron identificar las rutas más pesadas y evaluar la necesidad de optimizaciones adicionales para mejorar los tiempos de carga inicial, asegurando que el portal sea funcional y eficiente bajo condiciones reales de uso.

3.1.1. Herramienta de prueba

Para analizar las métricas de rendimiento del portal web de CuTB, se utilizó la funcionalidad de generación de reportes y optimización integrada en el framework **Next.js**. Este enfoque permitió evaluar con precisión el tamaño de las rutas, la carga inicial de JavaScript y la eficiencia general del sistema. Las razones para elegir esta funcionalidad son las siguientes:

- **Análisis detallado de rutas:** la herramienta de reporte de Next.js facilitó la inspección del tamaño y el peso de las diferentes rutas, tanto públicas como administrativas. Esto incluyó métricas como el tamaño de los archivos estáticos y el peso del JavaScript inicial, lo que ayudó a identificar áreas susceptibles de optimización.
- **Medición de la carga inicial:** gracias a las capacidades de Next.js, fue posible registrar los tamaños de las cargas iniciales de las rutas más utilizadas, detectando las rutas más pesadas, como /auth/login y /workWithUs. Estas métricas proporcionaron información valiosa para priorizar ajustes que mejoraran la experiencia de usuario.
- **Optimización automática y manual:** Next.js implementa técnicas de optimización automáticas, como la división de código (code splitting) y la carga diferida (lazy loading), que mejoran la eficiencia general del portal. Además, el reporte permitió identificar y reducir los puntos críticos, como la carga de archivos JavaScript compartidos, ajustando su peso total a 87.2 kB, una mejora significativa para el sistema.
- **Compatibilidad con el entorno:** dado que el portal de CuTB fue desarrollado con Node.js y Tailwind CSS, la integración con Next.js fue fluida, asegurando que las métricas generadas fueran precisas y representativas del entorno real de despliegue.

El uso de las herramientas de reporte de Next.js fue esencial para garantizar un desempeño eficiente del portal, identificando y resolviendo problemas clave en las rutas y recursos, y alcanzando un equilibrio óptimo entre funcionalidad y rendimiento para los usuarios finales.

3.1.2. Resultados de las pruebas

Las métricas generadas por las herramientas de análisis de Next.js permitieron evaluar la eficiencia del portal web de CuTB y la carga de sus diferentes rutas, tanto en escenarios de uso estándar como en situaciones más exigentes.

Escenarios de prueba:

- **Carga inicial:** se evaluaron las rutas principales, como /, /trainMe, /admin, y /workWithUs, midiendo el tamaño y el peso del JavaScript inicial. Los resultados mostraron que las rutas más optimizadas presentan un peso promedio de 227 kB, mientras que las más pesadas, como /auth/login, alcanzaron hasta 5.01 kB.
- **Evaluación estática y dinámica:** el análisis incluyó tanto rutas estáticas como dinámicas, revelando la eficacia del sistema en la generación de páginas pre-renderizadas y en tiempo de ejecución.

Resultados obtenidos: Se identificaron las siguientes áreas clave de optimización:

- **Tamaño compartido del JavaScript:** Se logró reducir el peso total de los scripts compartidos a 87.2 kB, lo que contribuyó a una carga más rápida y eficiente.
- **Balance entre rutas públicas y administrativas:** Las rutas administrativas, como /admin/providers y /admin/users, demostraron un desempeño estable, manteniendo tiempos de respuesta aceptables para un entorno multiusuario.

En la imagen a continuación se presenta una captura del reporte generado por Next.js, donde se detallan las métricas correspondientes a las rutas evaluadas. Ver Figura 3.1:

```

✓ Compiled successfully
✓ Linting and checking validity of types
✓ Collecting page data
✓ Generating static pages (25/25)
✓ Collecting build traces
✓ Finalizing page optimization

```

Route (app)	Size	First Load JS
o /	324 B	227 kB
o /_not-found	875 B	88.1 kB
o /about	324 B	227 kB
o /admin	1.86 kB	91.7 kB
o /admin/approvalCurseExam	502 B	227 kB
o /admin/courses	2.35 kB	231 kB
o /admin/exams	2.32 kB	231 kB
o /admin/membership_provider_request	2.16 kB	231 kB
o /admin/news	2.29 kB	231 kB
o /admin/providers	2.48 kB	231 kB
o /admin/scr	2.35 kB	231 kB
o /admin/users	2.31 kB	231 kB
o /auth/login	5.01 kB	183 kB
o /certifyMe	324 B	227 kB
o /news	913 B	227 kB
o /probar	140 B	87.3 kB
o /scr	1.29 kB	228 kB
o /trainMe	323 B	227 kB
o /trainMe/courses	324 B	227 kB
o /trainMe/providers	324 B	227 kB
o /unauthorized	185 B	99.2 kB
o /workWithUs	4.03 kB	179 kB
+ First Load JS shared by all	87.2 kB	
chunks/117-e3a8092bc23aaaa4.js	31.6 kB	
chunks/fd9d1056-7fbf5fc2b4cd2d64.js	53.7 kB	
other shared chunks (total)	1.95 kB	
f Middleware	27.5 kB	
o (Static) prerendered as static content		

Figura 3.1. Test

Estas pruebas y resultados permitieron identificar puntos críticos de mejora y realizar ajustes precisos para garantizar que el portal de CuTB cumpla con las expectativas de eficiencia y escalabilidad necesarias para un uso diario confiable.

3.2. Casos de Pruebas Funcionales

Los casos de prueba funcionales son fundamentales para garantizar que las funcionalidades clave del portal web de CuTB cumplan con los requisitos establecidos y se comporten de manera adecuada. Estos casos ayudan a validar que cada componente del sistema opere según lo esperado en diferentes escenarios.

Para el portal web de CuTB, se centraron las pruebas en la correcta gestión de las secciones principales del sistema: Entréname (cursos y proveedores), Certifícame (certificaciones), y Noticias. Los casos de prueba se diseñaron para evaluar aspectos como la creación, visualización y actualización de registros en estas secciones.

A continuación, se presenta una tabla que detalla los casos de prueba realizados, incluyendo el paso a paso de su ejecución, el resultado esperado y el estado final tras la ejecución de las pruebas:

Tabla 3.1. Casos de prueba

Caso de Prueba	Descripción	Resultado Esperado	Estado
Crear un curso	- Navegar a la sección Entréname. - Seleccionar "Agregar curso". - Ingresar datos válidos para el curso (nombre, proveedor, duración). - Guardar el curso y verificar que se haya creado correctamente.	Se permite crear un curso con un proveedor existente.	Satisfactorio
Publicar una noticia	- Navegar a la sección Noticias. - Seleccionar "Nueva noticia". - Ingresar título, contenido y fecha. - Guardar y verificar que la noticia sea visible en la lista.	Se permite publicar noticias con datos válidos.	Satisfactorio
Continúa en la siguiente página			

Tabla 3.1. (Continuación de la página anterior)

Caso de Prueba	Descripción	Resultado Esperado	Estado
Registrar una certificación	• Navegar a la sección Certifícame. • Seleccionar Registrar certificación". • Ingresar datos válidos (nombre, descripción, requisitos). • Guardar y verificar que la certificación sea accesible desde el portal.	Se permite registrar certificaciones con requisitos específicos.	Satisfactorio
Actualizar un curso	• Navegar a la sección Entréname. • Seleccionar un curso existente. • Modificar los datos del curso (nombre o proveedor). • Guardar los cambios y verificar que se hayan aplicado correctamente.	Se permite actualizar los datos de un curso existente.	Satisfactorio
Eliminar una noticia	• Navegar a la sección Noticias. • Seleccionar una noticia publicada. • Elegir la opción "Eliminar". • Verificar que la noticia ya no sea visible en la lista de noticias.	Se permite eliminar noticias sin afectar la estabilidad del sistema.	Satisfactorio

3.3. Pruebas de aceptación

Se realizaron pruebas de aceptación basadas en los casos de prueba funcionales previamente definidos. Estas pruebas involucraron a los usuarios finales del portal, quienes representan a los miembros de CuTB que

utilizarán el sistema en sus actividades diarias.

En la primera iteración, se detectaron algunos problemas menores relacionados con la actualización de noticias y la gestión de certificaciones, que no cumplían completamente con los parámetros esperados. Sin embargo, estos inconvenientes fueron solucionados y, en la segunda iteración, el sistema logró un 100 % de aceptación.

El éxito de estas pruebas confirma que el portal cumple con los requisitos establecidos, asegurando que las funcionalidades y la experiencia de usuario estén alineadas con lo esperado.

A continuación, se describen los casos de pruebas de aceptación definidos:

Tabla 3.2. Prueba de aceptación # 1

Caso de prueba de aceptación	
Código: HU7_P1	Historia de usuario: 7
Nombre: Listar usuarios certificados	
Descripción: Prueba la funcionalidad de listar los usuarios certificados.	
Condiciones de ejecución: El usuario debe estar en la ventana de SCR.	
Pasos de ejecución: El usuario selecciona en la barra de navegación la opción SCR.	
Resultados esperados: El sistema carga una página con el listado de usuarios certificados.	

Tabla 3.3. Prueba de aceptación # 2

Caso de prueba de aceptación	
Código: HU8_P1	Historia de usuario: 8
Nombre: Filtrar búsqueda de usuarios certificados	
Descripción: Prueba la funcionalidad de filtrar búsqueda de usuarios certificados.	
Condiciones de ejecución: El usuario debe estar en la ventana de SCR.	
Pasos de ejecución: El usuario selecciona el botón de Filtrar, el sistema muestra las opciones disponibles, el usuario selecciona una opción.	
Resultados esperados: El sistema establece el campo por el que se realizarán las búsquedas.	

Tabla 3.4. Prueba de aceptación # 3

Caso de prueba de aceptación	
Código: HU9_P1	Historia de usuario: 9
Nombre: Buscar usuarios certificados	
Descripción: Prueba la funcionalidad de buscar usuarios certificados.	
Condiciones de ejecución: El usuario debe estar en la ventana de SCR.	
Pasos de ejecución: El usuario rellena el campo del formulario, el sistema muestra los datos en la tabla a razón de la búsqueda del usuario.	
Resultados esperados: El sistema muestra los datos en la tabla a razón de la búsqueda del usuario.	

Tabla 3.5. Prueba de aceptación # 4

Caso de prueba de aceptación	
Código: HU10_P1	Historia de usuario: 10
Nombre: Iniciar sesión	
Descripción: Prueba la funcionalidad de iniciar sesión.	
Condiciones de ejecución: El usuario debe estar en la ventana de inicio de sesión.	
Pasos de ejecución: El usuario introduce los datos necesarios y verifica que sean correctos, el usuario presiona el botón Iniciar.	
Resultados esperados: En caso de que los datos fueron correctos el sistema inicia sesión y navega de forma automática al panel de administración, en caso contrario muestra un mensaje de error.	

Tabla 3.6. Prueba de aceptación # 5

Caso de prueba de aceptación	
Código: HU11_P1	Historia de usuario: 11
Nombre: Cerrar sesión	
Descripción: Prueba la funcionalidad de cerrar sesión.	
Condiciones de ejecución: El usuario debe estar autenticado, debe estar en el panel de administración.	
Pasos de ejecución: El usuario selecciona el botón de Cerrar sesión.	

Continúa en la próxima página

Tabla 3.6. Continuación de la página anterior

Resultados esperados: El sistema cierra la sesión del usuario y navega de forma automática a la sección de Home.

El resto de las pruebas de aceptación se muestran en el apéndice C de los anexos.

3.4. Resultado de las pruebas

Para garantizar que los entregables cumplan con los requisitos definidos, se llevaron a cabo pruebas siguiendo la metodología Scrum. En total, se ejecutaron diversas pruebas distribuidas entre unitarias, de aceptación y la resolución de no conformidades. El gráfico a continuación muestra la cantidad de pruebas realizadas y categorizadas según su tipo para cada caso de prueba. Esta información evidencia el nivel de cobertura y la atención prestada a las no conformidades, asegurando la calidad del producto final entregado. Ver Figura 3.2:

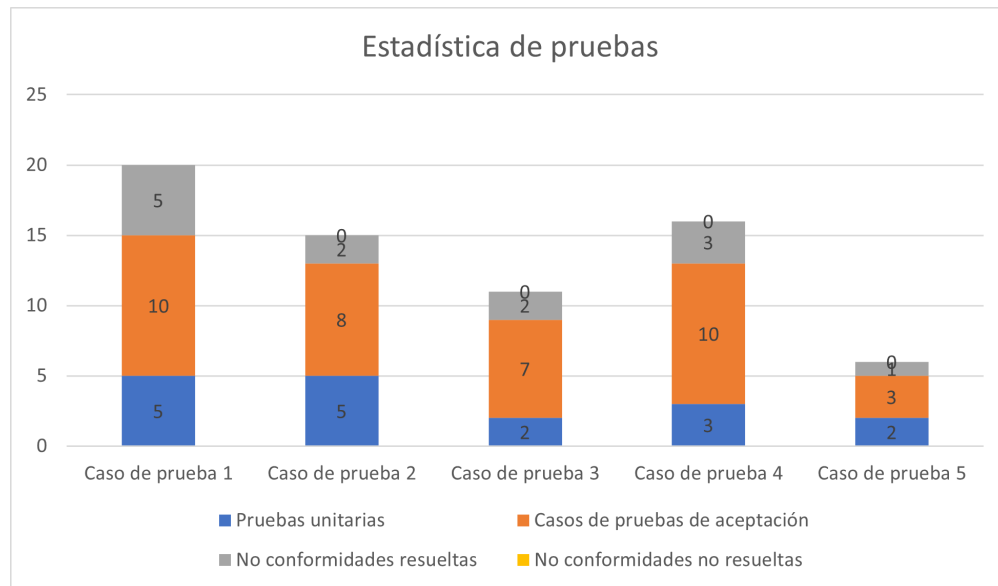


Figura 3.2. Resultado de pruebas

Conclusiones del capítulo

- Las pruebas de rendimiento confirmaron que el portal maneja eficientemente hasta 200 usuarios concurrentes, con tiempos de respuesta dentro de los límites aceptables.
- Los casos de prueba funcionales fueron satisfactorios, garantizando que las funcionalidades clave (gestión de cursos, noticias y certificaciones) operen correctamente.
- Las pruebas de aceptación fueron exitosas, alcanzando un 100 % de aceptación en la segunda iteración.

La tesis sobre el portal web de CuTB para certificaciones de software concluye con los siguientes resultados:

- **Cumplimiento del objetivo principal:** se desarrolló un portal web que cumple con los estándares del ISTQB, facilitando la gestión y difusión de información sobre certificaciones y cursos en Cuba.
- **Diseño y desarrollo eficiente:** la aplicación de la metodología ágil SCRUM y el uso de tecnologías modernas como Node.js y Next.js permitieron un desarrollo ágil y escalable, logrando un sistema funcional y adaptable a las necesidades del cliente.
- **Pruebas exitosas:** las pruebas de rendimiento y funcionalidad validaron la eficiencia y estabilidad del portal, asegurando tiempos de respuesta adecuados y una experiencia de usuario satisfactoria.
- **Impacto en la comunidad:** el portal web fortalece la visibilidad y accesibilidad de las certificaciones internacionales de software en Cuba, contribuyendo al desarrollo profesional de la comunidad de testers en el país.

En conclusión, el proyecto no solo satisface los requisitos técnicos establecidos, sino que también proporciona una herramienta clave para garantizar la sostenibilidad y el crecimiento de CuTB en su ámbito de acción.

Recomendaciones

Debido al continuo proceso de mejora y evolución que caracteriza a cualquier sistema de software, se sugieren las siguientes acciones:

- Incorporar una sección para que se puedan publicar materiales de estudio y que permita realizar test de autoevaluación de sus conocimientos.
- Utilizar una base de datos no relacional.

Referencias bibliográficas

- AGUIRRE, Santiago, 2021. *HTML5 Avanzado 1: Formularios Avanzados-Contenido Responsive-SEO*. RedUsers (vid. pág. 6).
- ARDIS, Mark; BUDGEN, David; HISLOP, Gregory W; OFFUTT, Jeff; SEBERN, Mark y VISSER, Willem, 2015. SE 2014: Curriculum guidelines for undergraduate degree programs in software engineering. *Computer*. Vol. 48, n.º 11, págs. 106-109 (vid. pág. 17).
- BARKER, Richard, 1994. *El modelo entidad-relación CASE* methodtm*. Ediciones Díaz de Santos (vid. pág. 19).
- CSS, 2024. *CSS: Hojas de estilo en cascada | MDN* [online] [visitado 2024-12-04]. Disponible desde: <https://developer.mozilla.org/en-US/docs/Web/CSS> (vid. pág. 6).
- DURÁN RODRÍGUEZ, Mayra Janeth. Modelo Integral de Atributos Para la Estimación de Historias de Usuario en Scrum (vid. pág. 11).
- FONTELA, Carlos, 2012. *UML: modelado de software para profesionales*. Alpha Editorial (vid. pág. 6).
- Introducción a Express/Node - Aprende desarrollo web | MDN*, 2024 [online] [visitado 2024-12-04]. Disponible desde: https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction (vid. pág. 7).
- IZERTIS. *Automatizar pruebas de una API usando Postman* [online] [visitado 2024-12-04]. Disponible desde: <https://www.izertis.com/es/-/blog/automatizar-pruebas-de-una-api-usando-postman> (vid. pág. 8).
- LOPEZ-PELLICER, Francisco J.; BÉJAR, Rubén; LATRE, Miguel A.; NOGUERAS-ISO, Javier y ZARAZAGA-SORIA, Francisco Javier, 2015. GitHub como herramienta docente. En: *GitHub como herramienta docente* [online]. Universitat Oberta La Salle, págs. 66-73 [visitado 2024-12-07]. ISBN 978-99920-70-10-9. Disponible desde: <https://upcommons.upc.edu/handle/2117/76761>. Accepted: 2015-09-14T09:28:14Z (vid. pág. 8).
- LUNA, Fernando, 2019. *JavaScript-Aprende a programar en el lenguaje de la web*. RedUsers (vid. pág. 6).
- Modelado de datos, 2024. *Modelado de datos* [online] [visitado 2024-12-04]. Disponible desde: <https://www.astera.com/es/type/blog/database-schema/>. Section: Blog (vid. pág. 19).

Apéndice

Entrevista

Preguntas de la entrevista realizada a la Dr. C.Yaimí Trujillo Casañola:

- ¿Existen procesos manuales que les gustaría automatizar? ¿Cuáles son los más prioritarios?
- ¿Cómo visualizan que este sistema puede contribuir al cumplimiento de los estándares del ISTQB?
- ¿Qué nivel de personalización consideran necesario para ajustarse a las operaciones específicas de la junta?
- ¿Existe alguna funcionalidad específica requerida por el ISTQB que debemos priorizar?
- ¿Cuántos tipos de usuarios diferentes (roles) manejará el sistema?
- ¿Qué funcionalidades consideran más urgentes para una primera versión del sistema?
- ¿Hay algo más que consideren importante y que no hayamos abordado en esta entrevista?

Historias de Usuario

Tabla B.1. Historia de usuario # 4

Historia de usuario	
Número: 4	Nombre: Descargar material de estudio
Usuario: Visitante	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 2
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Ofrecer la posibilidad de navegar al repositorio donde se encuentran los materiales de estudio de las diferentes certificaciones disponibles.	
Observaciones: Ninguna	

Tabla B.2. Historia de usuario # 5

Historia de usuario	
Número: 5	Nombre: Formulario de colaboración
Usuario: Visitante	
Prioridad en negocio: Bajo	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 2
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Proporcionar un formulario para que los interesados puedan postularse como proveedores o colaboradores.	
Observaciones: Ninguna	

Tabla B.3. Historia de usuario # 6

Historia de usuario	
Número: 6	Nombre: Mostrar noticias

Continúa en la próxima página

Tabla B.3. Continuación de la página anterior

Usuario: Visitante	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0.4	Iteración asignada: 2
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Publicar noticias relevantes sobre certificaciones, eventos y novedades de CuTB.	
Observaciones: Ninguna	

Tabla B.4. Historia de usuario # 7

Historia de usuario	
Número: 7	Nombre: Listar usuarios certificados
Usuario: Visitante	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.4	Iteración asignada: 2
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Mostrar un listado de los usuarios que ya han obtenido certificaciones ISTQB, con detalles sobre sus certificaciones.	
Observaciones: Ninguna	

Tabla B.5. Historia de usuario # 8

Historia de usuario	
Número: 8	Nombre: Filtrar búsqueda de usuarios certificados
Usuario: Visitante	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.4	Iteración asignada: 2
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir filtrar las de los usuarios certificados por nombre, certificación, nivel y fecha.	
Observaciones: Ninguna	

Tabla B.6. Historia de usuario # 9

Historia de usuario	
Número: 9	Nombre: Buscar usuarios certificados
Usuario: Visitante	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 2
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Implementar un buscador para localizar usuarios certificados de manera eficiente.	

Continúa en la próxima página

Tabla B.6. Continuación de la página anterior

Observaciones: Ninguna

Tabla B.7. Historia de usuario # 10

Historia de usuario	
Número: 10	Nombre: Iniciar sesión
Usuario: Administrador, Gestor de Actividades	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir que los administradores y gestores puedan acceder al sistema de manera segura mediante credenciales únicas.	
Observaciones: Ninguna	

Tabla B.8. Historia de usuario # 11

Historia de usuario	
Número: 11	Nombre: Cerrar sesión
Usuario: Administrador, Gestor de Actividades	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 0.1	Iteración asignada: 3
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Implementar una funcionalidad que permita a los administradores y gestores cerrar sesión de manera segura, asegurándose de que sus datos no permanezcan en caché.	
Observaciones: Ninguna	

Tabla B.9. Historia de usuario # 12

Historia de usuario	
Número: 12	Nombre: Regresar al panel de administración
Usuario: Administrador, Gestor de Actividades	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 0.1	Iteración asignada: 3
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Proporcionar un enlace o botón que permita regresar al panel de administración desde cualquier sección del sistema.	
Observaciones: Ninguna	

Tabla B.10. Historia de usuario # 13

Historia de usuario	
Número: 13	Nombre: Listar Usuarios
Usuario: Administradores	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Mostrar un listado de todos los usuarios registrados para fines administrativos.	
Observaciones: Ninguna	

Tabla B.11. Historia de usuario # 14

Historia de usuario	
Número: 14	Nombre: Añadir Usuario
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 4
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir la creación de nuevos usuarios en el sistema.	
Observaciones: Ninguna	

Tabla B.12. Historia de usuario # 15

Historia de usuario	
Número: 15	Nombre: Editar Usuario
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 4
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Modificar los datos de los usuarios existentes en el sistema.	
Observaciones: Ninguna	

Tabla B.13. Historia de usuario # 16

Historia de usuario	
Número: 16	Nombre: Eliminar Usuario
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: : Ibrahim Márquez Sánchez	

Continúa en la próxima página

Tabla B.13. Continuación de la página anterior

Descripción: Eliminar usuarios registrados de manera permanente.
Observaciones: Ninguna

Tabla B.14. Historia de usuario # 17

Historia de usuario	
Número: 17	Nombre: Listar Proveedores
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 5
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Mostrar un listado de todos los proveedores en el sistema.	
Observaciones: Ninguna	

Tabla B.15. Historia de usuario # 18

Historia de usuario	
Número: 18	Nombre: Añadir Proveedor
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 5
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir registrar nuevos proveedores en el sistema.	
Observaciones: Ninguna	

Tabla B.16. Historia de usuario # 19

Historia de usuario	
Número: 19	Nombre: Editar Proveedor
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 5
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir modificar la información de proveedores registrados.	
Observaciones: Ninguna	

Tabla B.17. Historia de usuario # 20

Historia de usuario	
Número: 20	Nombre: Eliminar Proveedor
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 5
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir la eliminación de proveedores registrados.	
Observaciones: Ninguna	

Tabla B.18. Historia de usuario # 21

Historia de usuario	
Número: 21	Nombre: Listar solicitudes de membresía o proveedor
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 5
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Mostrar un listado de todas las solicitudes de membresías o proveedores.	
Observaciones: Ninguna	

Tabla B.19. Historia de usuario # 22

Historia de usuario	
Número: 22	Nombre: Eliminar solicitud de membresía o proveedor
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 5
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir eliminar solicitudes de membresías o proveedores.	
Observaciones: Ninguna	

Tabla B.20. Historia de usuario # 23

Historia de usuario	
Número: 23	Nombre: Listar Noticias
Usuario: Gestor de Actividades	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	

Continúa en la próxima página

Tabla B.20. Continuación de la página anterior

Descripción: Mostrar todas las noticias disponibles en el portal.
Observaciones: Ninguna

Tabla B.21. Historia de usuario # 24

Historia de usuario	
Número: 24	Nombre: Añadir Noticias
Usuario: Gestor de Actividades	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir la creación de nuevas noticias para el portal.	
Observaciones: Ninguna	

Tabla B.22. Historia de usuario # 25

Historia de usuario	
Número: 25	Nombre: Editar Noticias
Usuario: Gestor de Actividades	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir la modificación de noticias existentes.	
Observaciones: Ninguna	

Tabla B.23. Historia de usuario # 26

Historia de usuario	
Número: 26	Nombre: Eliminar Noticias
Usuario: Gestor de Actividades	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Eliminar noticias del portal de forma permanente.	
Observaciones: Ninguna	

Tabla B.24. Historia de usuario # 27

Historia de usuario	
Número: 27	Nombre: Listar Cursos
Usuario: Gestor de Actividades	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Mostrar un listado de cursos disponibles.	
Observaciones: Ninguna	

Tabla B.25. Historia de usuario # 28

Historia de usuario	
Número: 28	Nombre: Añadir Cursos
Usuario: Gestor de Actividades	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Registrar nuevos cursos en el sistema.	
Observaciones: Ninguna	

Tabla B.26. Historia de usuario # 29

Historia de usuario	
Número: 29	Nombre: Editar Cursos
Usuario: Gestor de Actividades	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Modificar la información de cursos existentes.	
Observaciones: Ninguna	

Tabla B.27. Historia de usuario # 30

Historia de usuario	
Número: 30	Nombre: Eliminar Cursos
Usuario: Gestor de Actividades	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	

Continúa en la próxima página

Tabla B.27. Continuación de la página anterior

Descripción: Permitir la eliminación de cursos del sistema.
Observaciones: Ninguna

Tabla B.28. Historia de usuario # 31

Historia de usuario	
Número: 31	Nombre: Listar Exámenes
Usuario: Gestor de Actividades	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Mostrar un listado de las certificaciones disponibles.	
Observaciones: Ninguna	

Tabla B.29. Historia de usuario # 32

Historia de usuario	
Número: 32	Nombre: Añadir Exámenes
Usuario: Gestor de Actividades	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Registrar nuevas certificaciones en el sistema.	
Observaciones: Ninguna	

Tabla B.30. Historia de usuario # 33

Historia de usuario	
Número: 33	Nombre: Editar Exámenes
Usuario: Gestor de Actividades	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Modificar información de certificaciones existentes.	
Observaciones: Ninguna	

Tabla B.31. Historia de usuario # 34

Historia de usuario	
Número: 34	Nombre: Eliminar Exámenes
Usuario: Gestor de Actividades	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 6
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir la eliminación de certificaciones del sistema.	
Observaciones: Ninguna	

Tabla B.32. Historia de usuario # 35

Historia de usuario	
Número: 35	Nombre: Listar Usuarios Certificados
Usuario: Gestor de Actividades	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 7
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Mostrar un listado de todos los usuarios registrados.	
Observaciones: Ninguna	

Tabla B.33. Historia de usuario # 36

Historia de usuario	
Número: 36	Nombre: Añadir Usuario Certificado (SCR)
Usuario: Gestor de Actividades	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.3	Iteración asignada: 7
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir registrar nuevos usuarios en el sistema.	
Observaciones: Ninguna	

Tabla B.34. Historia de usuario # 37

Historia de usuario	
Número: 37	Nombre: Editar Usuario Certificado (SCR)
Usuario: Gestor de Actividades	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 7
Programador responsable: : Ibrahim Márquez Sánchez	

Continúa en la próxima página

Tabla B.34. Continuación de la página anterior

Descripción: Modificar información de usuarios registrados.
Observaciones: Ninguna

Tabla B.35. Historia de usuario # 38

Historia de usuario	
Número: 38	Nombre: Eliminar Usuario Certificado (SCR)
Usuario: Gestor de Actividades	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 7
Programador responsable: : Ibrahim Márquez Sánchez	
Descripción: Permitir eliminar usuarios del sistema.	
Observaciones: Ninguna	

Pruebas de Aceptación

Tabla C.1. Prueba de aceptación # 6

Caso de prueba de aceptación	
Código: HU13_P1	Historia de usuario: 13
Nombre: Listar usuarios	
Descripción: Prueba la funcionalidad de listar los usuarios.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: El usuario selecciona en el panel de administración la opción Gestionar usuarios.	
Resultados esperados: El sistema carga una página con el listado de los usuarios.	

Tabla C.2. Prueba de aceptación # 7

Caso de prueba de aceptación	
Código: HU14_P1	Historia de usuario: 14
Nombre: Registrar usuario	
Descripción: Prueba la funcionalidad de añadir un usuario.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de usuarios.	
Pasos de ejecución: El usuario selecciona el botón de Añadir Usuario, el sistema carga un modal con un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario presiona el botón Añadir.	
Resultados esperados: El sistema añade un usuario si los datos fueron correctos, en caso contrario muestra un mensaje de error.	

Tabla C.3. Prueba de aceptación # 8

Caso de prueba de aceptación	
Código: HU15_P1	Historia de usuario: 15
Nombre: Modificar usuario	
Descripción: Prueba la funcionalidad de editar usuario.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de usuarios.	
Pasos de ejecución: El usuario selecciona el botón de modificar, el sistema carga un modal con un formulario con los datos cargados, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario presiona el botón Guardar cambios.	
Resultados esperados: El sistema modifica un usuario si los datos fueron correctos, en caso contrario muestra un mensaje de error.	

Tabla C.4. Prueba de aceptación # 9

Caso de prueba de aceptación	
Código: HU16_P1	Historia de usuario: 16
Nombre: Eliminar usuario	
Descripción: Prueba la funcionalidad de eliminar un usuario.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de usuarios.	
Pasos de ejecución: El usuario selecciona el botón de eliminar, el sistema muestra un cuadro de confirmación, el usuario presiona el botón Confirmar.	
Resultados esperados: El sistema elimina un usuario del sistema.	

Tabla C.5. Prueba de aceptación # 10

Caso de prueba de aceptación	
Código: HU17_P1	Historia de usuario: 17
Nombre: Listar proveedores	
Descripción: Prueba la funcionalidad de listar los proveedores.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: El usuario selecciona en el panel de administración la opción Gestionar proveedores.	
Resultados esperados: El sistema carga una página con el listado de proveedores.	

Tabla C.6. Prueba de aceptación # 11

Caso de prueba de aceptación	
Código: HU18_P1	Historia de usuario: 18
Nombre: Registrar proveedor	
Descripción: Prueba la funcionalidad de añadir un proveedor.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de proveedores.	
Pasos de ejecución: El usuario selecciona el botón de Añadir Proveedor, el sistema carga un modal con un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario presiona el botón Añadir.	
Resultados esperados: El sistema añade un proveedor si los datos fueron correctos, en caso contrario muestra un mensaje de error.	

Tabla C.7. Prueba de aceptación # 12

Caso de prueba de aceptación	
Código: HU19_P1	Historia de usuario: 19
Nombre: Modificar proveedor	
Descripción: Prueba la funcionalidad de editar un proveedor.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de proveedores.	
Pasos de ejecución: El usuario selecciona el botón de editar, el sistema carga un modal con un formulario con los datos cargados, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario presiona el botón Aceptar.	
Resultados esperados: El sistema modifica un proveedor si los datos fueron correctos, en caso contrario muestra un mensaje de error.	

Tabla C.8. Prueba de aceptación # 13

Caso de prueba de aceptación	
Código: HU20_P1	Historia de usuario: 20
Nombre: Eliminar proveedor	
Descripción: Prueba la funcionalidad de eliminar un proveedor.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de proveedores.	

Continúa en la próxima página

Tabla C.8. Continuación de la página anterior

Pasos de ejecución: El usuario selecciona el botón de eliminar, el sistema muestra un cuadro de confirmación, el usuario presiona el botón Confirmar.
Resultados esperados: El sistema elimina un proveedor del sistema.

Tabla C.9. Prueba de aceptación # 14

Caso de prueba de aceptación	
Código: HU21_P1	Historia de usuario: 21
Nombre: Listar solicitudes de membresía o proveedor	
Descripción: Prueba la funcionalidad de listar las solicitudes de membresía o proveedor.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: El usuario selecciona en el panel de administración la opción Gestionar solicitudes de membresía o proveedor.	
Resultados esperados: El sistema carga una página con el listado de solicitudes de membresía o proveedor.	

Tabla C.10. Prueba de aceptación # 15

Caso de prueba de aceptación	
Código: HU22_P1	Historia de usuario: 22
Nombre: Eliminar solicitud de membresía o proveedor	
Descripción: Prueba la funcionalidad de eliminar una solicitud de membresía o proveedor.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de solicitudes de membresía o proveedor.	
Pasos de ejecución: El usuario selecciona el botón de eliminar, el sistema muestra un cuadro de confirmación, el usuario presiona el botón Confirmar.	
Resultados esperados: El sistema elimina una solicitud de membresía o proveedor del sistema.	

Tabla C.11. Prueba de aceptación # 16

Caso de prueba de aceptación

Continúa en la próxima página

Tabla C.11. Continuación de la página anterior

Código: HU23_P1	Historia de usuario: 23
Nombre: Listar noticias	
Descripción: Prueba la funcionalidad de listar las noticias.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: El usuario selecciona en el panel de administración la opción Gestionar noticias.	
Resultados esperados: El sistema carga una página con el listado de noticias.	

Tabla C.12. Prueba de aceptación # 17

Caso de prueba de aceptación	
Código: HU24_P1	Historia de usuario: 24
Nombre: Registrar noticia	
Descripción: Prueba la funcionalidad de añadir una noticia.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de noticias.	
Pasos de ejecución: El usuario selecciona el botón de Añadir Noticia, el sistema carga un modal con un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario presiona el botón Guardar.	
Resultados esperados: El sistema añade una noticia si los datos fueron correctos, en caso contrario muestra un mensaje de error.	

Tabla C.13. Prueba de aceptación # 18

Caso de prueba de aceptación	
Código: HU25_P1	Historia de usuario: 25
Nombre: Modificar noticia	
Descripción: Prueba la funcionalidad de editar una noticia.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de noticias.	
Pasos de ejecución: El usuario selecciona el botón de editar, el sistema carga un modal con un formulario con los datos cargados, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario presiona el botón Guardar.	

Continúa en la próxima página

Tabla C.13. Continuación de la página anterior

Resultados esperados: El sistema modifica una noticia si los datos fueron correctos, en caso contrario muestra un mensaje de error.
--

Tabla C.14. Prueba de aceptación # 19

Caso de prueba de aceptación	
Código: HU26_P1	Historia de usuario: 26
Nombre: Eliminar noticia	
Descripción: Prueba la funcionalidad de eliminar una noticia.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de noticias.	
Pasos de ejecución: El usuario selecciona el botón de eliminar, el sistema muestra un cuadro de confirmación, el usuario presiona el botón Confirmar.	
Resultados esperados: El sistema elimina una noticia del sistema.	

Tabla C.15. Prueba de aceptación # 20

Caso de prueba de aceptación	
Código: HU27_P1	Historia de usuario: 27
Nombre: Listar cursos	
Descripción: Prueba la funcionalidad de listar los cursos.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: El usuario selecciona en el panel de administración la opción Gestionar Cursos.	
Resultados esperados: El sistema carga una página con el listado de cursos.	

Tabla C.16. Prueba de aceptación # 21

Caso de prueba de aceptación	
Código: HU28_P1	Historia de usuario: 28
Nombre: Registrar curso	
Descripción: Prueba la funcionalidad de añadir un curso.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de cursos.	

Continúa en la próxima página

Tabla C.16. Continuación de la página anterior

Pasos de ejecución: El usuario selecciona el botón de Añadir Curso, el sistema carga un modal con un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario presiona el botón Aceptar.
Resultados esperados: El sistema añade un curso si los datos fueron correctos, en caso contrario muestra un mensaje de error.

Tabla C.17. Prueba de aceptación # 22

Caso de prueba de aceptación	
Código: HU29_P1	Historia de usuario: 29
Nombre: Modificar Curso	
Descripción: Prueba la funcionalidad de editar un curso.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de cursos.	
Pasos de ejecución: El usuario selecciona el botón de modificar, el sistema carga un modal con un formulario con los datos cargados, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario presiona el botón Guardar.	
Resultados esperados: El sistema modifica un curso si los datos fueron correctos, en caso contrario muestra un mensaje de error.	

Tabla C.18. Prueba de aceptación # 23

Caso de prueba de aceptación	
Código: HU30_P1	Historia de usuario: 30
Nombre: Eliminar curso	
Descripción: Prueba la funcionalidad de eliminar un curso.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de cursos.	
Pasos de ejecución: El usuario selecciona el botón de eliminar, el sistema muestra un cuadro de confirmación, el usuario presiona el botón Confirmar.	
Resultados esperados: El sistema elimina un curso del sistema.	

Tabla C.19. Prueba de aceptación # 24

Caso de prueba de aceptación	
Código: HU31_P1	Historia de usuario: 31
Nombre: Listar exámenes	
Descripción: Prueba la funcionalidad de listar los exámenes.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: El usuario selecciona en el panel de administración la opción Gestionar Exámenes.	
Resultados esperados: El sistema carga una página con el listado de exámenes.	

Tabla C.20. Prueba de aceptación # 25

Caso de prueba de aceptación	
Código: HU32_P1	Historia de usuario: 32
Nombre: Añadir examen	
Descripción: Prueba la funcionalidad de añadir un examen.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de exámenes.	
Pasos de ejecución: El usuario selecciona el botón de Añadir, el sistema carga un modal con un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario presiona el botón Aceptar.	
Resultados esperados: El sistema registra un examen si los datos fueron correctos, en caso contrario muestra un mensaje de error.	

Tabla C.21. Prueba de aceptación # 26

Caso de prueba de aceptación	
Código: HU33_P1	Historia de usuario: 33
Nombre: Editar examen	
Descripción: Prueba la funcionalidad de editar un examen.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de exámenes.	
Pasos de ejecución: El usuario selecciona el botón de editar, el sistema carga un modal con un formulario con los datos cargados, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario presiona el botón Guardar.	

Continúa en la próxima página

Tabla C.21. Continuación de la página anterior

Resultados esperados: El sistema modifica un examen si los datos fueron correctos, en caso contrario muestra un mensaje de error.
--

Tabla C.22. Prueba de aceptación # 27

Caso de prueba de aceptación	
Código: HU34_P1	Historia de usuario: 34
Nombre: Eliminar examen	
Descripción: Prueba la funcionalidad de eliminar un examen.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de exámenes.	
Pasos de ejecución: El usuario selecciona el botón de eliminar, el sistema muestra un cuadro de confirmación, el usuario presiona el botón Confirmar.	
Resultados esperados: El sistema elimina un examen del sistema.	

Tabla C.23. Prueba de aceptación # 28

Caso de prueba de aceptación	
Código: HU35_P1	Historia de usuario: 35
Nombre: Listar usuarios certificados	
Descripción: Prueba la funcionalidad de listar los usuarios certificados.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: El usuario selecciona en el panel de administración la opción Gestionar usuarios certificados.	
Resultados esperados: El sistema carga una página con el listado de usuarios certificados.	

Tabla C.24. Prueba de aceptación # 29

Caso de prueba de aceptación	
Código: HU36_P1	Historia de usuario: 36
Nombre: Añadir usuario certificado	
Descripción: Prueba la funcionalidad de añadir un usuario certificado.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de usuarios certificados.	

Continúa en la próxima página

Tabla C.24. Continuación de la página anterior

Pasos de ejecución: El usuario selecciona el botón de Añadir Registro, el sistema carga un modal con un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario presiona el botón Añadir.
Resultados esperados: El sistema añade un usuario certificado si los datos fueron correctos, en caso contrario muestra un mensaje de error.

Tabla C.25. Prueba de aceptación # 30

Caso de prueba de aceptación	
Código: HU37_P1	Historia de usuario: 37
Nombre: Modificar usuario certificado	
Descripción: Prueba la funcionalidad de modificar un usuario certificado.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de usuarios certificados.	
Pasos de ejecución: El usuario selecciona el botón de modificar, el sistema carga un modal con un formulario con los datos cargados, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario presiona el botón Guardar Cambios.	
Resultados esperados: El sistema modifica un usuario certificado si los datos fueron correctos, en caso contrario muestra un mensaje de error.	

Tabla C.26. Prueba de aceptación # 31

Caso de prueba de aceptación	
Código: HU38_P1	Historia de usuario: 38
Nombre: Eliminar usuario certificado	
Descripción: Prueba la funcionalidad de eliminar un usuario certificado.	
Condiciones de ejecución: El usuario debe estar autenticado, debe tener cargado el listado de usuarios certificados.	
Pasos de ejecución: El usuario selecciona el botón de eliminar, el sistema muestra un cuadro de confirmación, el usuario presiona el botón Confirmar.	
Resultados esperados: El sistema elimina un usuario certificado del sistema.	