

**Facultad de Ciencias Técnicas**

**Departamento de Informática**

**TRABAJO DE DIPLOMA**

*(Tesis presentada en opción al título de Ingeniero en Informática)*

**" Plataforma Web para la Gestión Eficiente de Información en la Cátedra Honorífica de Ciencias Técnicas "**

|  |  |
| --- | --- |
| **Autor:** | **Carlos Javier Pacheco Domínguez** |
| **Tutor(es):** | **Ing. Lucian Rodriguez Plasencia** |
|  |  |

**Pinar del Río - 2024**

# DECLARACIÓN DE AUTORÍA

Declaro que los resultados que aparecen en este informe son producto de la investigación realizada por este autor, por lo tanto, yo como mi tutor nos hacemos responsables de ellos. De igual forma autorizo a la Universidad de Pinar del Río “Hermanos Saíz Montes de Oca” a hacer uso del contenido de esta tesis con la finalidad que se estime conveniente.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Carlos Javier Pacheco Domínguez Ing. Lucian Rodriguez Plasencia



“Carlos Javier Pacheco Domínguez”

Cjpdtwet@gmail.com

Carlos Javier Pacheco Domínguez autoriza la divulgación del presente trabajo de diploma bajo licencia Creative Commons de tipo Reconocimiento No Comercial Sin Obra Derivada, se permite su copia y distribución por cualquier medio siempre que mantenga el reconocimiento de sus autores, no haga uso comercial de las obras y no realice ninguna modificación de ellas. La licencia completa puede consultarse en: <http://creativecommons.org/licenses/by-nc-nd/2.5/ar/legalcode>

Carlos Javier Pacheco Domínguez” autoriza al adscrito a la Universidad de Pinar del Río a distribuir el presente trabajo de diploma en formato digital bajo la licencia Creative Commons descrita anteriormente y a conservarlo por tiempo indefinido, según los requerimientos de la institución, en el repositorio de materiales didácticos disponible en:

Carlos Javier Pacheco Domínguez” autoriza al adscrito a la Universidad de Pinar del Río a distribuir el presente trabajo de diploma en formato digital bajo la licencia Creative Commons descrita anteriormente y a conservarlo por tiempo indefinido, según los requerimientos de la institución, en el repositorio de tesinas disponible en: [http://revistas.mes.edu.cu](http://revistas.mes.edu.cu/)

**Pensamiento o frase (opcional).**

*"No es lo que te sucede, sino cómo reaccionas ante ello lo que importa".*

*Epicteto*

# DEDICATORIA.

Con inmensa gratitud y amor, dedico esta tesis a mis queridos padres, Arianne Dóminguez Bernal y Juan Carlos Pacheco Barrera, quienes con su inquebrantable amor y sacrificio constante han sido el pilar de todos mis logros. Su ejemplo de perseverancia y dedicación ha sido mi guía en cada paso de este camino.

A mis abuelos, Zoila Bernal, Pilar Barrera y Reinaldo Domínguez, cuyas sabidurías y enseñanzas han dejado una huella imborrable en mi vida.

A mi hermana, Daniela Matos Alicea, cuya compañía y apoyo incondicional desde la distancia siempre me han brindado la fortaleza para seguir adelante.

A mis primos, Frank, David, Yandiel y Alberto, quienes siempre han estado a mi lado, apoyándome de todas las maneras posibles.

A mis tíos, Arian, Damaris, Gualberto y Rehuel, cuyo apoyo y aliento han sido fundamentales en este viaje.

A mis amigos, Eduardo, Luis Abel, Roger, Hairon, Luis Raúl y Orlando, por su apoyo y amistad incondicional que siempre me han alentado a seguir adelante.

Este trabajo es un reflejo de todo el amor, apoyo y sacrificio que he recibido de cada uno de ustedes. De corazón, les agradezco por ser mi inspiración y mi fuerza.

# AGRADECIMIENTOS.

Quiero expresar mi más profundo agradecimiento al Profesor e Ingeniero Eladio Carrillo, cuya guía y conocimiento han sido fundamentales en la realización de este trabajo. Su dedicación y apoyo constante han sido una fuente de inspiración y motivación.

A mi asesora, Isabel, le extiendo mi sincero agradecimiento por su invaluable ayuda y orientación a lo largo de este proyecto. Su paciencia y sabiduría han sido esenciales para superar los desafíos encontrados en el camino.

A mi tutor, Lucian, le agradezco por su ayuda durante este proceso.

También quiero reconocer a los profesores de la carrera que hicieron que me esforzara tanto.

Gracias a todos ustedes por su contribución a mi formación y éxito académico.

# RESUMEN.

El presente trabajo tiene como objetivo desarrollar una plataforma web especializada que optimice la gestión de información en el entorno de una cátedra honorífica, respondiendo a la necesidad de integrar las Tecnologías de la Información para mejorar la eficiencia, la interacción y la participación dentro de este ámbito académico. A través de una investigación detallada, se analizaron las necesidades y desafíos de la cátedra, incluyendo la desorganización de datos, la limitada comunicación entre sus miembros y las dificultades en la coordinación de actividades académicas y proyectos honoríficos. La propuesta contempla el diseño e implementación de una plataforma que permita centralizar y gestionar la información de manera eficiente, brindando una interfaz intuitiva, accesible y funcional. Además, la plataforma facilitará la coordinación de actividades y mejorará la interacción interna, promoviendo una mayor participación activa en las iniciativas académicas. Para el desarrollo, se emplearon tecnologías como React, Node.js y MongoDB, que garantizan una estructura escalable, segura y de fácil mantenimiento. Estas tecnologías permiten una experiencia de usuario dinámica y eficiente, optimizando la navegación y el acceso a la información dentro de la cátedra. Este sistema proporcionará a los usuarios una solución integral para la gestión de la información, mejorando la operatividad y colaboración dentro de la cátedra honorífica.

**PALABRAS CLAVES**

Cátedra, Proyecto, Información

# ABSTRACT

The aim of this work is to develop a specialized web platform that optimizes information management within an honorary chair, addressing the need to integrate Information Technologies to improve efficiency, interaction, and participation in this academic environment. Through detailed research, the needs and challenges of the chair were analyzed, including data disorganization, limited communication among its members, and difficulties in coordinating academic activities and honorary projects. The proposal includes the design and implementation of a platform that efficiently centralizes and manages information, providing an intuitive, accessible, and functional interface. Additionally, the platform will facilitate the coordination of activities and improve internal interaction, promoting greater active participation in academic initiatives. Technologies such as React, Node.js, and MongoDB were used for the development, ensuring a scalable, secure, and easy-to-maintain structure. These technologies provide a dynamic and efficient user experience, optimizing navigation and access to information within the chair. This system will offer users a comprehensive solution for information management, enhancing operability and collaboration within the honorary chair.

KEYWORDS  
Chair, Project, Information

Contenido

[DECLARACIÓN DE AUTORÍA II](#_Toc178934704)

[DEDICATORIA. IV](#_Toc178934705)

[AGRADECIMIENTOS. V](#_Toc178934706)

[RESUMEN. VI](#_Toc178934707)

[ABSTRACT VII](#_Toc178934708)

[INTRODUCCION. 1](#_Toc178934709)

[CAPITULO I: FUNDAMENTACION TEORICA. 15](#_Toc178934710)

[I.1Fundamentos Teóricos 15](#_Toc178934711)

[I.2 Diagnostico del estado actual del objeto de investigación. 16](#_Toc178934712)

[I.3 Sistemas informáticos afines. 17](#_Toc178934713)

[I.3.1 Sistema informático para la gestión de la información asociada a las Cátedras Honorificas en la Universidad de Ciencias Informáticas 17](#_Toc178934714)

[I.3.2 Plataforma digital para la satisfacción de las necesidades informacionales sobre la actividad científica estudiantil en Ciencias Médicas 17](#_Toc178934715)

[I.3.3 Conclusiones de los sistemas afines 18](#_Toc178934716)

[I.4 Caracterización de herramientas, metodologías y softwares que serán utilizados en el desarrollo del sistema propuesto. 19](#_Toc178934717)

[I.4.1 Metodología de desarrollo de software. 19](#_Toc178934718)

[I.4.2 Lenguaje de modelado 22](#_Toc178934719)

[I.4.3 Herramientas case, lenguaje de programación, bade de datos y mas tecnologías 24](#_Toc178934720)

[CAPITULO II: DISEÑO DEL SISTEMA 33](#_Toc178934721)

[II.1 Modelo de Dominio 33](#_Toc178934722)

[II.1.1 Definición de las entidades y concepto principales 33](#_Toc178934723)

[II.1.2 Representación del Modelo de Dominio 34](#_Toc178934724)

[II.1.3 Reglas del negocio a considerar 34](#_Toc178934725)

[II.2 Modelo de sistema 35](#_Toc178934726)

[II.2.1 Descripción de actores del sistema 35](#_Toc178934727)

[II.2.2 Requisitos funcionales 35](#_Toc178934728)

[II.2.3 Requisitos no funcionales 39](#_Toc178934729)

[II.2.4 Diagrama de Caso de Uso de Sistema 41](#_Toc178934730)

[II.2.5 Estudio de la factibilidad de las variables de Esfuerzo, Tiempo de Desarrollo y Costo de Proyecto 42](#_Toc178934731)

[II.2.5.1 Cálculo de los puntos de Caso de Uso 42](#_Toc178934732)

[II.2.5.2 Cálculo del factor de peso de los actores 43](#_Toc178934733)

[II.2.5.3 Cálculo del factor de peso de caso de uso sin ajustar 43](#_Toc178934734)

[II.2.5.4 Cálculo de los puntos de Caso de Uso Ajustado 43](#_Toc178934735)

[II.2.5.5 Cálculo de factor de complejidad técnica 43](#_Toc178934736)

[II.2.5.6 Cálculo del factor ambiente 44](#_Toc178934737)

[*II*.2.5.7 Cálculo del Esfuerzo de Implementación (E) 44](#_Toc178934738)

[II.2.5.8 Cálculo del Esfuerzo de Total (ET) 44](#_Toc178934739)

[CAPITULO III: DESCRIPCION DE LA SOLUCION PROPUESTA. 45](#_Toc178934740)

[III.1 Realización de Caso de Uso de Diseño 45](#_Toc178934741)

[III.1.1 Diagrama de Secuencias 45](#_Toc178934742)

[III.1.2 Diagramas de clases de caso de uso 47](#_Toc178934743)

[III.2 Modelo de Datos 49](#_Toc178934744)

[III.3 Definición del modelo de Arquitectura 50](#_Toc178934745)

[III.3.1 Patrón Cliente-Servidor 51](#_Toc178934746)

[III.4 Diagrama de Despliegue 52](#_Toc178934747)

[III.5 Principios de Diseño de Implementación en la Aplicación 55](#_Toc178934748)

[III.5.1 Interfaz de Usuario 56](#_Toc178934749)

[III.5.2 Formato de Salida de los Reportes 58](#_Toc178934750)

[III.6 Diagrama de Componentes 59](#_Toc178934751)

[III.7 Sistema de Seguridad 62](#_Toc178934752)

[III.7.1 Tratamiento de Errores 63](#_Toc178934753)

[III.8 Prueba 64](#_Toc178934754)

[CONCLUSIONES. 66](#_Toc178934755)

[RECOMENDACIONES. 68](#_Toc178934756)

[REFERENCIAS 70](#_Toc178934757)

[BIBLIOGRAFÍA 71](#_Toc178934758)

[ANEXOS. 72](#_Toc178934759)

[Anexo 1. Descripción de bajo nivel de todos los casos de uso del sistema 72](#_Toc178934760)

[Anexo 2. Estimación de Esfuerzo, Tiempo de Desarrollo y Costo 79](#_Toc178934761)

# INDICE DE TABLAS.

[Tabla 1 Descripción de CUS Gestión de Totales 42](#_Toc178933507)

[Tabla 2 Descripción de Prueba Añadir departamento 64](#_Toc178933508)

[Tabla 3 Descripción de Prueba eliminar tarea 65](#_Toc178933509)

[Tabla 4 Descripción de Prueba editar tarea 66](#_Toc178933510)

[Tabla 5 Descripción de Caso de Uso Gestión de Departamentos. 72](#_Toc178933511)

[Tabla 6 Descripción de Caso de Uso Gestión de Geografía. 73](#_Toc178933512)

[Tabla 7 Descripción de Caso de Uso Gestión de Publicaciones 74](#_Toc178933513)

[Tabla 8 Descripción de Caso de Uso Gestión de Personal 75](#_Toc178933514)

[Tabla 9 Descripción de Caso de Uso Gestión de Tareas 75](#_Toc178933515)

[Tabla 10 Descripción de Caso de Uso Gestión de Eventos Generales. 76](#_Toc178933516)

[Tabla 11 Descripción de Caso de Uso Gestión de Proyectos 77](#_Toc178933517)

[Tabla 12 Descripción de Caso de Uso Gestión de Premios 78](#_Toc178933518)

[Tabla 13 Descripción de Caso de Uso Gestión de Carreras 78](#_Toc178933519)

[Tabla 14 Descripción de Caso de Uso Gestión de Desglose 79](#_Toc178933520)

[Tabla 15 Factores técnicos 81](#_Toc178933521)

[Tabla 16 Factor Ambiente 82](#_Toc178933522)

# INDICE DE FIGURAS.

[Ilustración 1 Modelo de Dominio 34](#_Toc178933545)

[Ilustración 2 Diagrama de CUS 41](#_Toc178933546)

[Ilustración 3 Diagrama de Secuencia de login de un usuario o del administrador 46](#_Toc178933547)

[Ilustración 4 Diagrama de Secuencia de obtener desglose de Proyectos 46](#_Toc178933548)

[Ilustración 5 Diagrama de Secuencia de crear departamento 47](#_Toc178933549)

[Ilustración 6 Diagrama de clase gestionar Usuario 48](#_Toc178933550)

[Ilustración 7Diagrama de clase Gestionar Departamento 48](#_Toc178933551)

[Ilustración 8 Diagrama de clase Obtener Estadísticas 49](#_Toc178933552)

[Ilustración 9 Diagrama de Modelo de Datos Físico 50](#_Toc178933553)

[Ilustración 10 Representación de la Arquitectura 52](#_Toc178933554)

[Ilustración 11 Diagrama de Despliegue 54](#_Toc178933555)

[Ilustración 12 Interfaz Oscura 57](#_Toc178933556)

[Ilustración 13 Interfaz Clara 58](#_Toc178933557)

[Ilustración 14 Diagrama de componentes Añadir departamento 60](#_Toc178933558)

[Ilustración 15 Diagrama de componentes Obtener Geografía 61](#_Toc178933559)

[Ilustración 16 Diagrama de componentes Obtener Avatar del usuario 61](#_Toc178933560)

[Ilustración 17 Ejemplo de validación de campos obligatorios 63](#_Toc178933561)

[Ilustración 18 Ejemplo de validación del email 64](#_Toc178933562)

# INTRODUCCION.

Las Tecnologías de la Información y Comunicación (TIC) han emergido como protagonistas indiscutibles en la sociedad contemporánea, redefiniendo la forma en que interactuamos, aprendemos, trabajamos y nos comunicamos. Estas tecnologías, que abarcan desde hardware sofisticado hasta plataformas de software avanzadas, han tejido una red que conecta a individuos, empresas e instituciones en una red global interconectada.

En este contexto, se vuelve imperativo explorar la trascendencia y el impacto de las TIC en la sociedad actual, considerando su papel en la transformación de la comunicación, el acceso a la información y la generación de innovación. La presente investigación busca ahondar en la definición, aplicación, relevancia y repercusiones de estas tecnologías, con especial énfasis en un ámbito específico: la gestión de la información en el entorno académico.

**Definición y Aplicación de las TIC:**

En su esencia, las Tecnologías de la Información abarcan tanto hardware como software, proporcionando herramientas poderosas para almacenar, procesar y transmitir datos. La comunicación, impulsada por estas tecnologías, ha experimentado una metamorfosis, permitiendo interacciones instantáneas a través de diversas plataformas. En la esfera educativa, las TIC han revolucionado la forma en que se imparte el conocimiento, facilitando el acceso a recursos en línea y habilitando modalidades de aprendizaje a distancia.

\*Al profundizar en el ámbito académico, surge la necesidad de explorar cómo las TIC pueden ser canalizadas de manera específica para la gestión de información en el contexto de una cátedra honorífica.

**Relevancia e Impacto de las TIC**

La importancia de las TIC se evidencia en su contribución al desarrollo económico, la inclusión social y la transformación de la cultura. La globalización ha sido acelerada por estas tecnologías, generando oportunidades y desafíos. En el ámbito laboral, las TIC han transformado la naturaleza del trabajo, exigiendo habilidades digitales y fomentando la movilidad laboral.

En consonancia con esta evolución, la presente tesis busca explorar cómo las TIC pueden ser empleadas de manera estratégica en el diseño y desarrollo de un sitio web dedicado a la gestión de información en el contexto de una cátedra honorífica. Este sitio web no solo abordará la eficiencia en la gestión de datos, sino que también aspira a potenciar la experiencia académica y la participación en esta catedra de manera más integral.

El enfoque central de la presente tesis se centra en el desarrollo y aplicación de un sitio web especializado para la gestión eficiente de información en el contexto de una cátedra honorífica. Este proyecto abordará la creación de una plataforma digital que no solo optimice la administración de datos relacionados con la cátedra, sino que también busque mejorar la experiencia académica y la participación de los miembros involucrados.

La tesis explorará la relevancia de las Tecnologías de la Información y Comunicación (TIC) en el ámbito académico, destacando cómo estas herramientas pueden ser estratégicamente implementadas para fortalecer la gestión de información en una cátedra honorífica. A través de un enfoque integral, se buscará identificar y abordar las necesidades específicas de la cátedra, proporcionando una solución digital adaptada que contribuya a la eficiencia operativa y a la mejora de la interacción entre los miembros de la comunidad académica.

**Problemas encontrados**

En el contexto de la cátedra honorífica, se ha identificado una serie de dificultades y desafíos que afectan la gestión eficiente de la información. Actualmente, la falta de una plataforma digital especializada dificulta la organización y accesibilidad de datos relevantes para la cátedra. Este vacío en la infraestructura tecnológica conlleva a diversas problemáticas, entre las cuales se incluyen:

1.Desorganización de Datos: La ausencia de un sistema centralizado provoca la dispersión de información en diferentes formatos y ubicaciones, dificultando la localización y gestión efectiva de datos esenciales para el funcionamiento de la cátedra.

2. Limitada Interacción y Comunicación: La carencia de una plataforma digital específica limita las oportunidades para la interacción entre los miembros de la cátedra, así como la comunicación fluida de anuncios, eventos y recursos relevantes.

3. Dificultades en la Coordinación de Actividades: La falta de una herramienta dedicada para la planificación y coordinación de actividades académicas y eventos honoríficos puede resultar en la superposición de tareas y la falta de sincronización en la gestión de eventos especiales.

4. Barreras en la Participación Activa: La falta de un entorno digital interactivo podría obstaculizar la participación activa de los miembros de la cátedra, limitando su compromiso con las actividades y proyectos honoríficos.

5. Riesgo de Pérdida de Información: La dependencia de métodos tradicionales y no digitalizados aumenta el riesgo de pérdida de información crítica, ya sea debido a errores humanos, extravío de documentos o cambios en el personal encargado.

Estas dificultades resaltan la necesidad urgente de implementar una solución digital diseñada específicamente para abordar las complejidades de la gestión de información en una cátedra honorífica, con el objetivo de superar los obstáculos actuales y fortalecer la eficiencia y colaboración dentro de este entorno académico único.

**Problema científico**

En el ámbito de las cátedras honoríficas, la carencia de una infraestructura tecnológica adecuada para la gestión de información plantea un problema significativo que impacta la eficiencia y la interacción dentro de la comunidad académica. El problema científico central radica en la siguiente interrogante:

¿Cómo diseñar e implementar una plataforma web especializada que aborde de manera integral las necesidades específicas de una cátedra honorífica, optimizando la gestión de información, fomentando la interacción entre sus miembros y mejorando la participación activa en actividades académicas y proyectos honoríficos?

Este problema científico destaca la necesidad de desarrollar una solución digital innovadora y adaptada que no solo resuelva las dificultades identificadas en la gestión de información, sino que también promueva una mayor colaboración y participación en el contexto particular de una cátedra honorífica. La resolución de este problema contribuirá al avance del conocimiento en el campo de las Tecnologías de la Información aplicadas a entornos académicos especiales, ofreciendo una perspectiva valiosa sobre el diseño y la implementación de soluciones tecnológicas en este contexto específico.

**Objetivo general**

El objetivo principal de esta investigación es diseñar, desarrollar e implementar una plataforma web especializada para la gestión eficiente de información en el entorno de una cátedra honorífica. Esta plataforma aspira a superar las limitaciones actuales, proporcionando una solución integral que optimice la organización de datos, fomente la interacción entre los miembros de la cátedra y mejore la participación activa en actividades académicas y proyectos honoríficos**.**

Al alcanzar este objetivo general, se busca no solo abordar las dificultades identificadas en la gestión de información, sino también generar un impacto positivo en la dinámica y la colaboración dentro de la cátedra. La plataforma propuesta no solo actuará como un sistema eficiente de gestión de datos, sino que también se convertirá en un espacio interactivo que fortalezca la comunidad académica, promoviendo la excelencia en la ejecución de proyectos y actividades propias de la cátedra honorífica.

**Objeto de estudio.**

El objeto de estudio de esta investigación se centra en la conceptualización, diseño, desarrollo e implementación de una plataforma web especializada para la gestión de información en el contexto de una cátedra honorífica. Este objeto abarca los elementos y procesos involucrados en la creación de un entorno digital adaptado a las necesidades específicas de una cátedra honorífica, con el propósito de mejorar la eficiencia en la gestión de datos y potenciar la interacción y participación activa de sus miembros.

Componentes clave del objeto de estudio incluyen:

1.Requerimientos Específicos de la Cátedra Honorífica: Analizar y comprender las necesidades particulares de la cátedra en términos de gestión de información, coordinación de actividades académicas y promoción de la participación activa.

2.Desarrollo Tecnológico: Explorar las tecnologías pertinentes para el diseño y desarrollo de una plataforma web, considerando la accesibilidad, la seguridad y la usabilidad.

3.Diseño de la Interfaz y Experiencia del Usuario (UX): Diseñar una interfaz intuitiva y amigable que facilite la interacción de los usuarios con la plataforma, promoviendo una experiencia positiva y eficiente.

4.Integración de Funcionalidades Específicas:Identificar y desarrollar funciones específicas que aborden las problemáticas identificadas, como la gestión centralizada de datos, la coordinación de actividades y la facilitación de la comunicación interna.

5.Evaluación y Validación: Evaluar la eficacia de la plataforma a través de pruebas y validación, garantizando que cumpla con los objetivos propuestos y satisfaga las necesidades de la cátedra honorífica.

El objeto de estudio se enmarca en la intersección entre las Tecnologías de la Información y el ámbito académico, con el propósito de ofrecer una solución innovadora y adaptada que contribuya a mejorar la gestión de información y fortalecer la dinámica interna de una cátedra honorífica.

**Campo de acción**

El campo de acción de esta investigación se sitúa en la intersección de las Tecnologías de la Información (TI) y el ámbito académico, con un enfoque específico en el diseño y desarrollo de una plataforma web para la gestión de información en el contexto de una cátedra honorífica. Este campo abarca diversas dimensiones y actividades que contribuirán a la consecución de los objetivos propuestos.

Aspectos clave del campo de acción incluyen:

1.Análisis de Requerimientos Académicos: Explorar y comprender en detalle las necesidades y demandas específicas de la cátedra honorífica en términos de gestión de información, coordinación de actividades y participación de los miembros.

2.Investigación Tecnológica: Identificar las tecnologías más adecuadas para el diseño y desarrollo de la plataforma, considerando factores como la seguridad, la escalabilidad y la compatibilidad con los dispositivos utilizados por los miembros de la cátedra.

3.Diseño de la Arquitectura del Sistema: Desarrollar una arquitectura sólida y eficiente que respalde las funcionalidades necesarias, asegurando la integración coherente de componentes y la adaptabilidad a futuras actualizaciones.

4.Desarrollo de la Plataforma Web: Implementar la solución propuesta, llevando a cabo el diseño de la interfaz, la codificación de funcionalidades y la integración de sistemas, con un enfoque centrado en la experiencia del usuario.

5.Pruebas y Evaluación: Realizar pruebas exhaustivas para validar la efectividad y robustez de la plataforma, garantizando su funcionamiento adecuado y su capacidad para abordar las problemáticas identificadas.

6.Implementación y Adopción: Facilitar la transición hacia la nueva plataforma, brindando soporte técnico y capacitación para garantizar una adopción exitosa por parte de los miembros de la cátedra honorífica.

Este campo de acción busca crear un entorno digital que no solo resuelva las dificultades identificadas, sino que también promueva la colaboración y la participación activa en el marco de la cátedra honorífica, contribuyendo así a la mejora continua de las prácticas académicas y administrativas en este contexto específico.

**Tareas de investigación.**

Las tareas de investigación asociadas a este proyecto se diseñarían para abordar las diversas dimensiones del desarrollo de la plataforma web especializada para la gestión de información en una cátedra honorífica. Aquí se presentan algunas tareas clave que podrían llevarse a cabo:

1.Revisión Bibliográfica:

- Explorar investigaciones previas sobre el uso de tecnologías de la información en entornos académicos y la gestión de información en cátedras honoríficas.

- Analizar plataformas web similares utilizadas en otros contextos y su impacto en la eficiencia y colaboración.

2. Entrevistas y Encuestas:

- Realizar entrevistas con los miembros de la cátedra honorífica para comprender sus necesidades y expectativas en términos de gestión de información y colaboración.

- Diseñar y distribuir encuestas para recopilar datos cuantitativos sobre la frecuencia y las preferencias en el uso de herramientas digitales.

3.Análisis de Requerimientos:

- Identificar y documentar los requisitos específicos de la cátedra honorífica en cuanto a la gestión de datos, coordinación de actividades y comunicación interna.

- Establecer criterios de usabilidad y accesibilidad para la plataforma.

4.Investigación Tecnológica:

- Evaluar diversas tecnologías y frameworks web que puedan ser apropiados para el desarrollo de la plataforma, considerando factores como la seguridad, la escalabilidad y la compatibilidad.

- Analizar las mejores prácticas en el diseño de interfaces de usuario (UI) y la experiencia del usuario (UX).

5.Diseño de la Arquitectura del Sistema:

- Desarrollar un plan detallado de la arquitectura de la plataforma, definiendo la estructura de la base de datos, los módulos y la lógica de negocio.

- Diseñar un esquema de seguridad robusto para proteger la información sensible.

6.Desarrollo de la Plataforma:

- Implementar la interfaz de usuario según las especificaciones del diseño.

- Codificar las funciones específicas de gestión de información, coordinación de actividades y comunicación interna.

- Integrar tecnologías de seguridad y realizar pruebas de seguridad.

7.Pruebas y Evaluación:

- Realizar pruebas de funcionalidad para asegurar que todas las características operan correctamente.

- Evaluar la usabilidad de la plataforma mediante pruebas con usuarios reales de la cátedra honorífica.

- Identificar y corregir posibles fallos o áreas de mejora.

8.Implementación y Adopción:

- Desplegar la plataforma en un entorno de producción.

- Facilitar sesiones de capacitación para los miembros de la cátedra sobre el uso efectivo de la plataforma.

- Establecer un mecanismo de soporte técnico y recopilar retroalimentación continua.

Estas tareas de investigación se estructuran de manera secuencial para abordar los diversos aspectos del proyecto, desde la comprensión de las necesidades hasta la implementación exitosa y la adopción por parte de la comunidad académica.

**Resultados esperados.**

Los resultados esperados de esta investigación y desarrollo de la plataforma web para la gestión de información en una cátedra honorífica se enfocarán en lograr mejoras significativas en la eficiencia, la colaboración y la participación activa de sus miembros. Aquí se presentan los resultados esperados:

1.Plataforma Funcional:

- Desarrollo e implementación exitosa de la plataforma web especializada que cumpla con los requisitos y expectativas de la cátedra honorífica.

- Interfaz de usuario intuitiva y amigable que facilite la navegación y el acceso a las funciones clave.

2.Optimización de la Gestión de Información:

- Centralización efectiva de datos relevantes, facilitando su almacenamiento, actualización y acceso de manera eficiente.

- Mejora significativa en la organización y recuperación de información crítica para la toma de decisiones.

3.Coordinación Eficiente de Actividades:

- Herramientas integradas que permitan la planificación y coordinación eficaz de actividades académicas y proyectos honoríficos.

- Reducción de superposiciones y conflictos en la programación de eventos y tareas.

4.Fomento de la Colaboración y Comunicación:

- Mejora en la interacción y comunicación entre los miembros de la cátedra a través de funciones como mensajes internos, foros y notificaciones.

- Mayor participación en discusiones académicas y actividades colaborativas.

5.Incremento en la Participación Activa:

- Estímulo de la participación activa de los miembros en proyectos, eventos y decisiones importantes dentro de la cátedra honorífica.

- Mayor involucramiento en actividades académicas y honoríficas a través de la plataforma.

6.Seguridad de la Información:

- Implementación de medidas de seguridad efectivas para proteger la integridad y confidencialidad de la información almacenada en la plataforma.

7.Satisfacción del Usuario:

- Retroalimentación positiva de los usuarios de la plataforma en términos de usabilidad, accesibilidad y eficacia.

- Evaluación favorable en encuestas y evaluaciones de satisfacción.

8.Eficiencia Operativa:

- Reducción de tiempos y esfuerzos dedicados a tareas administrativas y de coordinación, permitiendo un uso más eficiente del tiempo.

- Mayor agilidad en la toma de decisiones y ejecución de proyectos.

Estos resultados esperados reflejan el impacto positivo que se busca lograr en la dinámica y la eficiencia operativa de la cátedra honorífica a través de la implementación de la plataforma web especializada. La consecución de estos objetivos contribuirá al éxito y la relevancia de la investigación.

**Métodos de investigación**

La investigación propuesta para el desarrollo de la plataforma web destinada a la gestión de información en una cátedra honorífica se basará en enfoques metodológicos específicos para garantizar una comprensión profunda de las necesidades, expectativas y desafíos inherentes a este entorno académico particular.

1. Entrevistas y Encuestas:

- Se llevarán a cabo entrevistas con profesores, estudiantes y personal administrativo de la cátedra honorífica para obtener percepciones detalladas sobre sus requisitos y preferencias.

- Se diseñarán encuestas estructuradas para recopilar datos cuantitativos sobre el uso de tecnologías existentes y la disposición para adoptar una nueva plataforma.

2. Análisis de Requerimientos:

- La colaboración cercana con los miembros de la cátedra permitirá identificar y documentar exhaustivamente los requisitos específicos, incluyendo la gestión de información, funcionalidades deseadas y desafíos actuales.

- Sesiones de trabajo colaborativo se organizarán para definir criterios de usabilidad y accesibilidad.

3. Investigación Tecnológica:

- Se llevará a cabo una evaluación detallada de tecnologías web para seleccionar aquellas adecuadas para el desarrollo de la plataforma, considerando aspectos como seguridad, escalabilidad y facilidad de integración.

- Se analizarán casos de estudio de plataformas similares para incorporar las mejores prácticas en diseño y desarrollo.

4. Diseño de la Interfaz y Experiencia del Usuario (UX):

- Prototipos de la interfaz de usuario se desarrollarán y someterán a pruebas de usabilidad con miembros de la cátedra para garantizar la intuición y facilidad de uso.

- La retroalimentación sobre el diseño visual y la navegación se recopilará para realizar ajustes antes de la implementación.

5. Pruebas y Evaluación:

- Se realizarán pruebas exhaustivas de las funciones específicas de gestión de información, coordinación de actividades y comunicación interna.

- Las pruebas de seguridad serán fundamentales para proteger la información sensible.

- Evaluaciones de usabilidad con usuarios reales ayudarán a identificar áreas de mejora.

6. Implementación y Adopción:

- La plataforma se implementará en un entorno de producción y se ofrecerán sesiones de capacitación personalizadas para los miembros de la cátedra.

- Un sistema de soporte técnico efectivo se establecerá, y se recopilará retroalimentación continua para realizar ajustes según sea necesario.

Estos métodos se alinean estratégicamente para abordar las complejidades de la gestión de información en una cátedra honorífica, asegurando un enfoque integral y efectivo en el desarrollo de la plataforma propuesta.

**Resumen del contenido de cada capitulo**

Capítulo I: Fundamentación Teórica

En este primer capítulo, se establecerán las bases conceptuales que sustentan la necesidad y relevancia del proyecto. Se realizará un diagnóstico del estado actual de la gestión de información en cátedras honoríficas, destacando las dificultades y desafíos existentes. Se llevará a cabo un análisis de sistemas informáticos afines y se caracterizarán las herramientas, metodologías y software que servirán de base para el desarrollo de la plataforma. Este capítulo sentará las bases teóricas y prácticas que respaldarán el diseño y desarrollo de la solución propuesta.

Capítulo II: Diseño del Sistema

El segundo capítulo se enfocará en el diseño detallado de la plataforma web para la gestión eficiente de información en la cátedra honorífica. Se abordarán aspectos arquitectónicos, funcionales y de interfaz de usuario. Se definirá la estructura de la base de datos, los módulos esenciales del sistema y la lógica de negocio. Además, se destacará la importancia de la seguridad de la información y se propondrán soluciones para proteger la integridad y confidencialidad de los datos. Este capítulo proporcionará un plano sólido y coherente para la implementación de la solución.

Capítulo III: Descripción de la Solución Propuesta

En este capítulo, se detallará la solución propuesta para la gestión eficiente de información en la cátedra honorífica. Se presentarán las características específicas de la plataforma, haciendo hincapié en cómo aborda las dificultades identificadas en el diagnóstico inicial. Se describirán las funcionalidades clave, como la centralización de datos, la coordinación de actividades y las herramientas de comunicación interna. Además, se explicará la interfaz de usuario diseñada para mejorar la experiencia de los miembros de la cátedra. Este capítulo proporcionará una visión integral de la solución, estableciendo la base para las conclusiones y recomendaciones finales de la tesis.

# CAPITULO I: FUNDAMENTACION TEORICA.

En este primer capítulo de la tesis, se establecen los cimientos teóricos y se examina el estado actual de la cátedra honorífica, identificando las oportunidades de mejora a través de una plataforma web. Se contrastan sistemas informáticos similares para extraer lecciones valiosas y se caracterizan las herramientas y metodologías seleccionadas para el desarrollo, justificando su elección en función de su idoneidad para alcanzar los objetivos educativos y técnicos del proyecto.

## I.1Fundamentos Teóricos

Los fundamentos teóricos para el desarrollo de una web destinada a la cátedra honorífica se basan en la necesidad de centralizar y facilitar el acceso a la información académica. Se consideran las teorías de gestión del conocimiento, que enfatizan la importancia de compartir y transferir información de manera eficiente en un entorno educativo. Además, se abordan los principios de arquitectura de la información, asegurando que la estructura del sitio web permita una navegación lógica y una búsqueda intuitiva. Desde el punto de vista técnico, se examinan las mejores prácticas en desarrollo web, como la responsividad y la accesibilidad, para garantizar que la plataforma sea utilizable en diversos dispositivos y por todos los usuarios. Estos fundamentos teóricos son cruciales para crear una web que no solo sirva como repositorio de información, sino que también promueva una comunidad académica más colaborativa e interactiva.

## I.2 Diagnostico del estado actual del objeto de investigación.

La estructura organizativa de la cátedra honorífica de ciencias técnicas es fundamental para su funcionamiento eficiente y la consecución de sus objetivos académicos. La cátedra está compuesta por varios departamentos, cada uno representando una disciplina técnica específica dentro del amplio espectro de las ciencias técnicas. Cada departamento tiene un representante que aporta su experiencia y conocimiento especializado, asegurando que todas las áreas de interés estén adecuadamente representadas.

En la cúspide de la estructura se encuentra el presidente de la cátedra, quien es responsable de la dirección estratégica y la toma de decisiones clave. El presidente trabaja en estrecha colaboración con los representantes de los departamentos para desarrollar programas educativos, promover la investigación y fomentar la colaboración interdisciplinaria.

Además, la cátedra puede incluir comités o grupos de trabajo dedicados a tareas específicas, como el desarrollo curricular, la innovación tecnológica y la colaboración con la industria. Estos grupos están formados por miembros seleccionados de diferentes departamentos, lo que facilita un enfoque integrado y una distribución equitativa de las responsabilidades.

Actualmente, la cátedra honorífica gestiona su información de manera manual, almacenando documentos y datos en carpetas dentro de una PC. Esta práctica conlleva varias limitaciones significativas: la recuperación de información es lenta y tediosa, la seguridad de los datos no está garantizada, y la colaboración entre profesores y estudiantes se ve obstaculizada. Además, la falta de un sistema de respaldo adecuado aumenta el riesgo de pérdida de información importante.

A través de métodos de indagación empírica como el análisis de documentos y entrevistas con el personal académico, se ha revelado que la gestión manual ha resultado en una distribución ineficiente del tiempo y recursos, afectando negativamente la productividad de la cátedra. Los estudiantes también han expresado frustración debido a la dificultad para acceder a materiales de estudio y comunicarse con los docentes.

La implementación de una plataforma web centralizada sería un paso significativo para mejorar la gestión de la información y la comunicación dentro de esta estructura organizativa. Permitiría una colaboración más fluida entre los departamentos, una distribución más eficiente de los recursos y un acceso más rápido a la información relevante para todos los miembros de la cátedra.

## I.3 Sistemas informáticos afines.

### I.3.1 Sistema informático para la gestión de la información asociada a las Cátedras Honorificas en la Universidad de Ciencias Informáticas

Este sistema permite la organización, centralización y disponibilidad de la información estadística de las CH en la UCI, además de brindar información referente de las CH y las actividades desarrolladas en años anteriores y las que se estén desarrollando en el momento. Por otra parte, permite a los usuarios la posibilidad de hacer consultas personalizadas a través de distintos filtros de búsqueda de la información de una cátedra. (UCI, 2022)

### I.3.2 Plataforma digital para la satisfacción de las necesidades informacionales sobre la actividad científica estudiantil en Ciencias Médicas

La plataforma digital mencionada tiene como objetivo principal satisfacer las necesidades informacionales de los estudiantes de la Universidad de Ciencias Médicas de Ciego de Ávila, Cuba, en relación con la Actividad Científica Estudiantil (ACE). Esta plataforma está diseñada para ofrecer un flujo continuo y actualizado de información sobre las actividades científicas que se llevan a cabo en la universidad. Permite a los estudiantes acceder a recursos relevantes para mejorar su formación en investigación, responder a las demandas de conocimiento durante su proceso de aprendizaje, y fomentar su participación en eventos científicos.

En resumen, la plataforma tiene el propósito de:

1. **Facilitar el acceso a la información** sobre la ACE en la universidad.
2. **Actualizar de manera dinámica** el contenido relacionado con los proyectos de investigación estudiantil y eventos científicos.
3. **Satisfacer las necesidades de los estudiantes** en cuanto a la formación en investigación científica.
4. **Fomentar el desarrollo de competencias investigativas** desde el pregrado, proporcionando información clave para el proceso de aprendizaje científico.

En definitiva, es una herramienta que busca mejorar la calidad y cantidad de la producción científica estudiantil, brindando a los estudiantes los recursos adecuados para desarrollarse como futuros investigadores en el ámbito de las Ciencias Médicas. (acimed, 2020)

### I.3.3 Conclusiones de los sistemas afines

No se puede usar directamente los sistemas mencionados para la cátedra honorífica de ciencias técnicas de la universidad principalmente porque:

1. **Contexto específico**: Ambos sistemas están diseñados para satisfacer necesidades muy particulares de sus respectivas áreas, como las cátedras honoríficas en la Universidad de Ciencias Informáticas y la actividad científica estudiantil en Ciencias Médicas. La cátedra de ciencias técnicas tiene un enfoque diferente y debe ajustarse a los requisitos específicos de esa disciplina.
2. **Enfoque en necesidades informacionales distintas**: Mientras que el sistema de la UCI se centra en la gestión estadística de las cátedras y su actividad histórica, la plataforma para Ciencias Médicas busca mejorar la formación y participación en investigación científica.
3. **Modelo de negocio o estructura organizacional**: Cada cátedra o plataforma está desarrollada bajo un modelo de gestión propio, diseñado para satisfacer las demandas y objetivos de su entorno institucional y académico. Adaptar esas estructuras a un contexto de ciencias técnicas podría resultar poco eficiente o inadecuado, ya que el enfoque y las actividades diferirían de lo que se busca en la cátedra honorífica

En resumen, aunque los sistemas tienen elementos útiles, se necesitara adaptar o desarrollar un sistema que responda específicamente a las características y demandas de la cátedra de ciencias técnicas.

## I.4 Caracterización de herramientas, metodologías y softwares que serán utilizados en el desarrollo del sistema propuesto.

### I.4.1 Metodología de desarrollo de software.

Utilizar una metodología de desarrollo de software es crucial por varias razones:

**Organización y Estructura:** Una metodología proporciona un marco claro y estructurado, lo que facilita la planificación y ejecución del proyecto. Permite a los equipos trabajar de manera coherente y ordenada, evitando el caos y la improvisación.

**Mejora de la Calidad:** Las metodologías incluyen buenas prácticas y estándares que ayudan a garantizar la calidad del producto final. Esto incluye pruebas rigurosas, revisiones de código y retroalimentación continua.

**Gestión de Riesgos:** Identificar y mitigar riesgos es una parte esencial de cualquier proyecto. Las metodologías de desarrollo de software ayudan a anticipar problemas y a implementar soluciones antes de que se conviertan en obstáculos significativos.

**Comunicación y Colaboración:** Proporcionan una base común para la comunicación entre los miembros del equipo y con los stakeholders. Esto asegura que todos estén alineados con los objetivos del proyecto y que las expectativas sean claras.

**Flexibilidad y Adaptación:** Metodologías como Agile permiten la adaptación y el cambio, lo que es crucial en entornos dinámicos donde las necesidades y los requisitos pueden evolucionar rápidamente.

**Entrega a Tiempo y Presupuesto:** Ayudan a gestionar el tiempo y los recursos de manera eficiente, aumentando las probabilidades de cumplir con los plazos y el presupuesto establecidos.

**Documentación y Trazabilidad:** Facilitan la creación de documentación detallada y la trazabilidad de todas las decisiones y cambios realizados durante el desarrollo, lo cual es valioso para el mantenimiento futuro y la resolución de problemas.

En resumen, una metodología de desarrollo de software no solo mejora la calidad y eficiencia del proyecto, sino que también facilita la gestión de riesgos, la comunicación y la colaboración, asegurando una entrega exitosa y a tiempo.

#### I.4.1.1 RUP

El Proceso Unificado de Rational (RUP, por sus siglas en inglés) es una metodología de desarrollo de software que se basa en principios de desarrollo iterativo e incremental. Fue desarrollado por Rational Software, que luego fue adquirida por IBM. RUP proporciona un marco de trabajo adaptable para el desarrollo de software, permitiendo a los equipos ajustar el proceso según las necesidades específicas del proyecto.

Fases de RUP

RUP se divide en cuatro fases principales:

**Inicio:** Definición del alcance y los requisitos del proyecto. Aquí se identifican los riesgos y se establece una visión del sistema.

**Elaboración:** Refina y amplía los requisitos iniciales. Se desarrollan los modelos arquitectónicos y se mitigan los riesgos principales.

**Construcción:** Desarrollo del producto, incluyendo la codificación y las pruebas. Aquí se construyen y refinan los componentes del sistema.

**Transición:** Se enfoca en la entrega del producto al usuario final. Incluye pruebas beta, despliegue y soporte inicial.

¿Por qué usar RUP como metodología de software?

Desarrollo Iterativo e Incremental: Permite la entrega de partes funcionales del software en ciclos cortos, lo que facilita la gestión de cambios y la mejora continua del producto.

**Gestión de Riesgos:** RUP identifica y mitiga riesgos tempranamente en el ciclo de vida del proyecto, reduciendo la probabilidad de fallos críticos.

**Enfoque en la Arquitectura:** Proporciona una estructura clara y bien definida para el sistema, asegurando que se construya una base sólida y escalable desde el principio.

**Adaptabilidad:** Es flexible y puede ajustarse según las necesidades específicas del proyecto y del equipo, permitiendo una personalización que maximiza la eficiencia y eficacia.

**Documentación Exhaustiva:** Promueve la creación de documentación detallada en cada fase del proyecto, lo que facilita el mantenimiento y la evolución del software a largo plazo.

**Mejor Comunicación y Colaboración:** RUP fomenta la comunicación clara y estructurada entre los miembros del equipo y los stakeholders, asegurando que todos estén alineados con los objetivos del proyecto.

En resumen, RUP es una metodología robusta y flexible que puede mejorar significativamente la gestión y calidad de los proyectos de desarrollo de software.

### I.4.2 Lenguaje de modelado

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es un estándar internacional para especificar, visualizar, construir y documentar los artefactos de un sistema de software. UML proporciona un conjunto de diagramas y técnicas para la representación gráfica de la arquitectura, el diseño y la implementación del software, cubriendo tanto los aspectos estructurales como de comportamiento del sistema.

Diagramas UML

UML incluye varios tipos de diagramas, entre los más utilizados se encuentran:

**Diagrama de Clases:** Muestra la estructura estática del sistema, incluyendo clases, atributos, operaciones y las relaciones entre ellos.

**Diagrama de Casos de Uso:** Representa las interacciones entre los usuarios y el sistema, describiendo las funciones principales que ofrece el sistema.

**Diagrama de Secuencia:** Ilustra cómo los objetos interactúan en un escenario particular de uso, ordenando las interacciones en el tiempo.

**Diagrama de Actividades:** Muestra el flujo de actividades dentro de un sistema, similar a un diagrama de flujo.

**Diagrama de Estados:** Representa los estados y transiciones de un objeto a lo largo de su ciclo de vida.

**Diagrama de Componentes:** Describe cómo el sistema se descompone en componentes y las interacciones entre ellos.

Por Qué Usar UML

Claridad y Comprensión: UML proporciona una manera visual de representar sistemas complejos, lo que facilita la comprensión y comunicación entre los miembros del equipo y los stakeholders.

**Documentación:** Permite crear una documentación detallada y precisa del sistema, útil tanto durante el desarrollo como para el mantenimiento posterior.

**Estandarización:** Al ser un lenguaje estándar, UML asegura que todos los involucrados en el proyecto puedan entender y trabajar con los modelos sin necesidad de herramientas propietarias.

**Análisis y Diseño:** Ayuda en el análisis de requisitos y en el diseño de la arquitectura del sistema, asegurando que todas las partes del sistema están bien definidas e integradas.

**Flexibilidad:** Puede ser utilizado en diferentes etapas del desarrollo del software, desde la captura de requisitos hasta el diseño detallado y la implementación.

**Mejora de la Calidad:** Al proporcionar una visión clara y estructurada del sistema, UML ayuda a identificar y resolver problemas de diseño antes de que se conviertan en costosos errores en etapas posteriores.

**Comunicación Efectiva:** Facilita la comunicación entre desarrolladores, diseñadores, analistas y otros stakeholders, asegurando que todos tienen una comprensión común del sistema.

En resumen, UML es una herramienta poderosa y versátil que mejora la calidad, claridad y eficiencia del proceso de desarrollo de software, haciendo más sencilla la gestión de sistemas complejos.

### I.4.3 Herramientas case, lenguaje de programación, bade de datos y mas tecnologías

Herramientas y Tecnologías

#### ****I.4.3.1 JavaScript****

JavaScript es uno de los lenguajes de programación más populares y es utilizado tanto en el front-end como en el back-end de aplicaciones web. Su versatilidad lo convierte en una herramienta fundamental para crear interfaces interactivas y sistemas de lógica en el servidor. JavaScript permite a los desarrolladores escribir código para diferentes plataformas sin tener que cambiar de lenguaje.

* + **Ventajas**: Su capacidad para ejecutarse tanto en el navegador como en el servidor lo hace ideal para desarrollar aplicaciones completas. Además, cuenta con una gran comunidad y ecosistema de librerías y frameworks.
  + **Uso en el proyecto**: Se utilizará en todo el desarrollo, tanto en el cliente como en el servidor, para gestionar la lógica de la aplicación, la manipulación del DOM y la comunicación con el servidor.

#### ****I****.4.3.2 React

**React**: React es una biblioteca de JavaScript desarrollada por Facebook para crear interfaces de usuario. A través de su enfoque basado en componentes, React permite desarrollar aplicaciones de una manera más modular y eficiente. Los componentes de React pueden ser reutilizados, lo que ahorra tiempo y reduce el margen de errores.

* + **Ventajas**: Ofrece una experiencia de usuario optimizada gracias a su Virtual DOM, que actualiza solo los elementos necesarios. Además, su ecosistema permite integrar fácilmente bibliotecas externas.
  + **Uso en el proyecto**: Se empleará para desarrollar el front-end, creando una interfaz dinámica, rápida y modular que mejore la experiencia del usuario.

#### ****I****.4.3.3 Node.js

**Node.js**: Node.js es un entorno de ejecución de JavaScript en el servidor que utiliza un modelo basado en eventos no bloqueantes, lo que lo hace altamente eficiente para manejar múltiples solicitudes concurrentes. Esto lo convierte en una opción ideal para aplicaciones en tiempo real como chats o sistemas que requieren conexiones constantes.

* + **Ventajas**: Su capacidad para manejar grandes cantidades de solicitudes simultáneas sin bloquear el hilo principal permite aplicaciones de alto rendimiento.
  + **Uso en el proyecto**: Será utilizado en el back-end para manejar la lógica del servidor, gestionar la base de datos y procesar las solicitudes del cliente de manera eficiente.

#### ****I****.4.3.4 Express.js

**Express.js**: Express es un framework minimalista para Node.js que facilita la creación de servidores y APIs. Proporciona un conjunto de herramientas sencillo pero poderoso para manejar rutas, solicitudes HTTP, middlewares, y más. A pesar de su simplicidad, Express es extremadamente flexible y personalizable.

* + **Ventajas**: Su sencillez y flexibilidad lo convierten en el framework más popular para la construcción de aplicaciones web y APIs con Node.js. Facilita la gestión de rutas y solicitudes, y permite integrarse con middleware fácilmente.
  + **Uso en el proyecto**: Se utilizará para gestionar las rutas y los servicios RESTful en el back-end, manejando las solicitudes HTTP y enviando respuestas al cliente.

#### ****I****.4.3.5 Mongoose

**Mongoose**: Mongoose es una biblioteca para Node.js que proporciona una solución elegante para trabajar con MongoDB mediante esquemas y validaciones. Simplifica la interacción con la base de datos MongoDB, permitiendo a los desarrolladores trabajar con datos en un formato más estructurado y definido.

* + **Ventajas**: Ofrece una estructura robusta para manejar datos, proporcionando validaciones, middleware, y la capacidad de definir esquemas para los documentos de MongoDB.
  + **Uso en el proyecto**: Se empleará para modelar y validar los datos que se almacenarán en MongoDB, facilitando la integración con la base de datos y asegurando que los datos sean coherentes y correctos.

#### ****I****.4.3.6 MongoDB

**MongoDB**: MongoDB es una base de datos NoSQL que almacena datos en formato de documentos JSON, lo que permite una mayor flexibilidad que las bases de datos relacionales tradicionales. Es muy utilizada en aplicaciones que manejan grandes volúmenes de datos y que requieren alta disponibilidad y escalabilidad.

* + **Ventajas**: Su estructura de documentos es flexible, lo que permite cambios en el esquema de datos sin afectar a toda la base. Es ideal para aplicaciones que necesitan escalar y manejar grandes cantidades de datos sin problemas de rendimiento.
  + **Uso en el proyecto**: Será utilizada como base de datos para almacenar los datos del sistema, aprovechando su flexibilidad y capacidad para gestionar datos en formatos no estructurados.

#### ****I.4.3.7 JWT****

**JSON Web Token (JWT)**: JWT es un estándar para la autenticación de usuarios que permite enviar información de forma segura entre el cliente y el servidor en formato JSON. Los tokens JWT son autovalidables, lo que elimina la necesidad de mantener sesiones en el servidor, mejorando la escalabilidad del sistema.

* + **Ventajas**: Proporciona una autenticación segura y escalable. Los tokens son fáciles de generar y pueden contener toda la información necesaria para identificar a un usuario sin tener que consultar constantemente una base de datos.
  + **Uso en el proyecto**: Se empleará para la autenticación de usuarios, asegurando que las solicitudes están autorizadas mediante el uso de tokens JWT.

#### ****I****.4.3.8 Bcrypt

**Bcrypt**: Bcrypt es una biblioteca que proporciona un método seguro para el hashing de contraseñas, protegiendo las credenciales de los usuarios. Utiliza técnicas de salting y hashing que dificultan los ataques de fuerza bruta.

* + **Ventajas**: Proporciona una alta seguridad para el almacenamiento de contraseñas. Su proceso de hashing incluye un "salt", lo que hace que las contraseñas no puedan ser fácilmente descifradas incluso si la base de datos es comprometida.
  + **Uso en el proyecto**: Se usará para cifrar las contraseñas antes de almacenarlas en la base de datos, asegurando que las credenciales de los usuarios estén protegidas.

#### ****I****.4.3.9 Connect Multiparty

**Connect Multiparty**: Connect Multiparty es un middleware que facilita la gestión de peticiones multipart/form-data, necesarias para la carga de archivos. Simplifica la recepción y el manejo de archivos en el servidor sin complicaciones.

* + **Ventajas**: Facilita el manejo de archivos en peticiones HTTP. Su capacidad para procesar archivos de manera eficiente lo convierte en una herramienta esencial para sistemas que permiten la carga de imágenes o documentos.
  + **Uso en el proyecto**: Será utilizado para gestionar las subidas de archivos en el sistema, como imágenes o documentos, permitiendo una experiencia fluida para el usuario.

#### ****I****.4.3.10 Moment.js/ Day.js

**Moment.js / Day.js**: Moment.js es una popular biblioteca de JavaScript para la manipulación de fechas y tiempos. Ofrece una API intuitiva para formatear, analizar y comparar fechas. Sin embargo, su peso ha hecho que alternativas más ligeras como Day.js ganen popularidad.

* + **Ventajas**: Proporciona una forma sencilla y flexible de trabajar con fechas en JavaScript. Day.js es una versión más ligera, con un tamaño reducido y un rendimiento superior.
  + **Uso en el proyecto**: Se emplearán para manejar y formatear fechas, especialmente en la creación de registros, eventos y tiempos en la aplicación.

#### ****I****.4.3.11 React Router DOM

**React Router DOM**: React Router DOM es una biblioteca que facilita la navegación entre diferentes páginas o componentes en aplicaciones React de manera dinámica. Permite a los desarrolladores crear rutas y gestionar la transición entre ellas sin recargar la página.

* + **Ventajas**: Permite una navegación fluida dentro de aplicaciones de una sola página (SPA). Proporciona soporte para rutas dinámicas y navegación basada en el historial del navegador.
  + **Uso en el proyecto**: Se utilizará para manejar la navegación entre las diferentes vistas de la aplicación, permitiendo a los usuarios moverse de una sección a otra sin interrupciones.

#### ****I****.4.3.12 Redux

**Redux**: Redux es una herramienta para la gestión del estado global de las aplicaciones. Ayuda a centralizar el estado de la aplicación en un único lugar, lo que facilita la sincronización entre los componentes de la UI y mejora la predictibilidad del comportamiento de la aplicación.

* + **Ventajas**: Facilita la gestión de estados complejos y asegura que todos los componentes tengan acceso al estado global de manera eficiente y predecible.
  + **Uso en el proyecto**: Se usará para gestionar el estado global del sistema, asegurando que la aplicación funcione de manera coherente y predecible en todos sus componentes.

#### I.4.3.13 React Dropzone

**React Dropzone**: React Dropzone es un componente que permite la carga de archivos mediante la funcionalidad drag-and-drop, lo que mejora la experiencia del usuario en aplicaciones web. Simplifica la implementación de áreas de carga de archivos intuitivas.

* + **Ventajas**: Mejora la experiencia del usuario al permitir una carga de archivos más intuitiva y amigable mediante la técnica de arrastrar y soltar.
  + **Uso en el proyecto**: Se utilizará para la carga de archivos en el front-end, facilitando el proceso de subida de documentos de manera sencilla y visualmente atractiva.

#### ****I.4.3.14 JWT Decode****

**JWT Decode**: JWT Decode es una biblioteca que permite decodificar tokens JWT sin necesidad de validarlos. Facilita la extracción de información de los tokens, como el nombre del usuario o sus permisos.

* + **Ventajas**: Permite obtener información del token JWT sin necesidad de hacer una consulta al servidor. Es útil para extraer datos como el rol del usuario o su identificación.
  + **Uso en el proyecto**: Será utilizada para decodificar los tokens JWT y obtener información sobre el usuario logueado de manera eficiente en el front-end.

#### ****I****.4.3.15 Nivo

**@nivo/core y @nivo/geo**: @nivo/core es una librería de visualización de datos que proporciona gráficos altamente personalizables y eficientes. @nivo/geo se especializa en la visualización geográfica de datos mediante mapas interactivos.

* + **Ventajas**: Ofrecen gráficos y visualizaciones ricas que permiten al usuario interactuar con los datos de manera intuitiva.
  + **Uso en el proyecto**: Serán utilizados para crear visualizaciones gráficas y mapas interactivos en la aplicación, permitiendo a los usuarios realizar análisis visuales de los datos.

#### ****I****.4.3.16 Material UI

**@mui/x-date-pickers, @mui/x-data-grid, @mui/material, @mui/icons-material**: Material UI (MUI) es un conjunto de componentes de interfaz de usuario basados en el sistema de diseño de Google Material Design. Proporciona componentes predefinidos y personalizables como pickers de fechas, grids de datos, y botones con iconos.

* + **Ventajas**: Facilita la creación de interfaces modernas, atractivas y coherentes con los estándares de diseño actuales.
  + **Uso en el proyecto**: Se utilizarán para crear una interfaz de usuario intuitiva, moderna y estéticamente atractiva, brindando una experiencia de usuario fluida y agradable.

#### ****I****.4.3.17 VSCODE

**Visual Studio Code (VSCode)**: VSCode es un editor de código ligero y altamente personalizable que se ha convertido en una de las herramientas favoritas de los desarrolladores. Su soporte para múltiples lenguajes y su amplia variedad de extensiones hacen que sea muy flexible.

* + **Ventajas**: Ofrece una gran cantidad de herramientas integradas, como depuración, control de versiones y terminal, todo en un solo lugar.
  + **Uso en el proyecto**: Será utilizado para escribir y depurar el código del sistema, aprovechando su integración con herramientas de desarrollo como Git y Node.js.

#### ****I****.4.3.18 ****Enterprise Architect herramienta CASE****

**Enterprise Architect**: Enterprise Architect es una herramienta de modelado y diseño de software que permite a los desarrolladores crear diagramas UML y gestionar grandes proyectos de software. Es muy utilizado en el diseño de arquitecturas de software complejas.

* + **Ventajas**: Proporciona un entorno completo para el modelado y la documentación de proyectos de software, facilitando la creación de diagramas UML y otros esquemas visuales.
  + **Uso en el proyecto**: Se empleará para diseñar y documentar la arquitectura del sistema, asegurando que todos los componentes del software estén bien estructurados y definidos.

#### ****I.4.3.19 Mongo Modeler herramienta CASE****

**Mongo Modeler**: Mongo Modeler es una herramienta gráfica para diseñar y visualizar esquemas de bases de datos MongoDB. Permite crear esquemas visuales de las colecciones y relaciones en MongoDB, lo que facilita la planificación de la base de datos.

* + **Ventajas**: Facilita la creación y visualización de esquemas de bases de datos, ayudando a diseñar de manera más efectiva la estructura de datos.
  + **Uso en el proyecto**: Se utilizará para diseñar y planificar los esquemas de la base de datos MongoDB, permitiendo una estructura clara y organizada.

# CAPITULO II: DISEÑO DEL SISTEMA

### II.1 Modelo de Dominio

Un **modelo de dominio** en la resolución de problemas e ingeniería del software, es un modelo conceptual de todos los temas relacionados con un problema específico. En él se describen las distintas entidades, sus atributos, papeles y relaciones, además de las restricciones que rigen el dominio del problema. Se crea con el fin de representar el vocabulario y los conceptos clave del dominio del problema. Captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema. (wikipedia, 2023)

### II.1.1 Definición de las entidades y concepto principales

Para una mejor comprensión del problema a resolver, es muy importante definir las entidades involucradas en el Modelo del Dominio. A continuación se explican los conceptos principales de dichas entidades:

**Activista:** activista de una carrera que es el encargado de realizar los informes

**Departamento:** uno de los departamentos de la cátedra

**Carrera:** carrera técnica que pertenece a cierto departamento

**Proyecto:** informe del proyecto que tiene todos los datos del mismo

Miembro: Miembro o colaborador que participa en el proyecto

**Premio:**  informe del premio que tiene todos los datos del mismo y se le otorga a un proyecto

**Publicación :**  informe de la publicación de algún miembro de la carrera

**Tarea:** informe de la tarea que se programa en la cátedra

**Evento:** informe del evento que se programa en la cátedra

### II.1.2 Representación del Modelo de Dominio



Ilustración 1 Modelo de Dominio

### II.1.3 Reglas del negocio a considerar

Las reglas de negocio guían la toma de decisiones diarias dentro de las empresas trazando las relaciones entre los objetos, como los nombres de cliente y sus pedidos correspondientes. Esta conversión de las actividades de negocio de una organización en lógica empresarial concreta permite a los ingenieros de software y analistas de negocio aplicar estas reglas en herramientas de flujos de trabajo u otras aplicaciones para habilitar la automatización de procesos. Sin ellas, los procesos de actualización pueden volverse más pesados y lentos, y los documentos están más expuestos a error humano e inconsistencias. Una empresa que implementa reglas de negocio puede ahorrar tiempo y dinero gracias a la optimización del trabajo y un menor abandono. (IBM, 2022)

A continuación se relacionan las reglas identificadas para este negocio:

* Una carrera tiene que tener un solo departamento
* Un activista solo puede pertenecer a un departamento.
* Un proyecto solo puede pertenecer a una carrera
* El proyecto puede tener muchos miembros y colaboradores
* Un miembro puede estar en varios proyectos
* Un proyecto puede tener varios premios
* Un premio puede ser otorgado a varios proyectos
* Una publicación solo puede ser de una carrera
* La catedra puede tener varias tareas
* La catedra puede tener varios eventos

## II.2 Modelo de sistema

### II.2.1 Descripción de actores del sistema

**Administrador del Sitio:** es el encargado de darle permiso a los demás usuarios del sistema para poder interactuar con todos los recursos de la aplicación.

**Usuario:** usuario con permiso del administrador para poder gestionar la información en el sistema

### II.2.2 Requisitos funcionales

Un requisito funcional es una declaración de cómo debe comportarse un sistema. Define lo que el sistema debe hacer para satisfacer las necesidades o expectativas del usuario. Los requisitos funcionales se pueden considerar como características que el usuario detecta. Son diferentes de los requisitos no funcionales, que definen cómo debe funcionar internamente el sistema (p. ej., rendimiento, seguridad, etc.).

Los requisitos funcionales se componen de dos partes: función y comportamiento. La función es lo que hace el sistema (por ejemplo, "calcular el impuesto sobre las ventas"). El comportamiento es cómo lo hace el sistema (p. ej., "El sistema calculará el impuesto sobre las ventas multiplicando el precio de compra por la tasa impositiva"). (Visure Solutions, Inc., 2020)

Los requisitos funcionales a desarrollar son:

R1 Gestión de Usuarios

• R1.1 Añadir Usuario

• R1.2 Actualizar Usuario

• R1.3 Buscar Usuario

• R1.4 Eliminar Usuario

• R1.5 Activar Usuario

• R1.6 Desactivar Usuario

R2 Gestión de Departamentos

• R2.1 Añadir Departamento

• R2.2 Actualizar Departamento

• R2.3 Buscar Departamento

• R2.4 Eliminar Departamento

R3 Gestión de Proyectos

• R3.1 Añadir Proyecto

• R3.2 Actualizar Proyecto

• R3.3 Buscar Proyecto

• R3.4 Eliminar Proyecto

R4 Gestión de Carreras

• R4.1 Añadir Carrera

• R4.2 Actualizar Carrera

• R4.3 Buscar Carrera

• R4.4 Eliminar Carrera

R5 Gestión de Geografía

• R5.1 Buscar Geografía

R6 Gestión de Desglose

• R6.1 Buscar Desglose

R7 Gestión de Totales

• R7.1 Buscar Totales

R8 Gestión de Avatares

• R8.1 Buscar Avatar

• R8.2 Actualizar Avatar

R9 Gestión de Eventos Generales

• R9.1 Añadir Evento General

• R9.2 Actualizar Evento General

• R9.3 Buscar Evento General

• R9.4 Eliminar Evento General

R10 Gestión de Publicaciones

• R10.1 Añadir Publicación

• R10.2 Actualizar Publicación

• R10.3 Buscar Publicación

• R10.4 Eliminar Publicación

R11 Gestión de Premios

• R11.1 Añadir Premio

• R11.2 Actualizar Premio

• R11.3 Buscar Premio

• R11.4 Eliminar Premio

• R11.5 Otorgar Premio

• R11.6 Revocar Premio

R12 Gestión de Tareas

• R12.1 Añadir Tarea

• R12.2 Actualizar Tarea

• R12.3 Buscar Tarea

• R12.4 Eliminar Tarea

R13 Gestión de Personal

• R13.1 Añadir Personal

• R13.2 Actualizar Personal

• R13.3 Buscar Personal

• R13.4 Eliminar Personal

### II.2.3 Requisitos no funcionales

Los requisitos no funcionales (NFR) son las restricciones impuestas a un sistema que definen sus atributos de calidad. Por lo general, se denotan con adjetivos como seguridad, rendimiento y escalabilidad. Los requisitos no funcionales son importantes porque ayudan a garantizar que el sistema satisfaga las necesidades del usuario. (Visure Solutions, Inc., 2020)

Los requisitos no funcionales para el software son:

**Apariencia o interfaz gráfica:**

- La interfaz del sistema será muy sencilla de usar, apenas se necesitara entrenamiento para utilizar el software.

**Rendimiento:**

- Tiempo de respuestas rápidas, el backend utiliza como base de datos mongodb que es una base de datos ligera y con una alta velocidad de respuesta así como indexado.

**Escalabilidad:**

- El software tiene la capacidad de crecer en volumen de datos sin afectar su rendimiento.

**Mantenibilidad:**

- El software será muy fácilmente modificable ya que se usan patrones de diseño como controller,proxy,watcher entre otros.

**Seguridad:**

- El software será seguro ya que se usa encriptación para el guardado de la contraseña de cada usuario en la base de datos y la obtención de su información, también división de privilegios

**Software:**

- El software que se necesita para el correcto funcionamiento de la aplicación debe ser alguno de los siguientes: GNU/Linux, window 8 o superior, MacOs.

**Hardware:**

- Los ordenadores en donde se vaya a interactuar con la aplicación deben contar con un procesador Dual Core o mayor, con 4 GB de RAM DDR3 o superior y un HDD de 100 GB de almacenamiento o superior.

### II.2.4 Diagrama de Caso de Uso de Sistema



Ilustración 2 Diagrama de CUS

#### II.2.4.1 Descripción de bajo nivel de los casos de uso del sistema

|  |  |
| --- | --- |
| Caso de Uso | Gestión de Totales |
| Actor(es) | 1-Administrador  2-usuario activo |
| Propósito | Gestionar la información de departamentos, carreras, personal y proyectos existentes |
| Descripción | 1- El actor accede a la plataforma o toca el botón de menú dashboard |
| Referencias | R7. |
| Precondiciones | El actor ha ingresado al sistema o se encuentra en la ventana dashboard |
| Poscondiciones | La información que contiene la base de datos relacionada a los departamentos ,carreras, personal y proyectos se procesa |
| Acción del Actor | Respuesta del Sistema |
|  |  |
| Otras acciones | - |
| Requerimientos especiales | - |

Tabla 1 Descripción de CUS Gestión de Totales

Nota: Otros en el Anexo 1

### II.2.5 Estudio de la factibilidad de las variables de Esfuerzo, Tiempo de Desarrollo y Costo de Proyecto

Antes de implementar el sistema se realizó una estimación del esfuerzo, tiempo de

desarrollo y costo, aplicando el método Punto de Casos de Uso, el cual fue ideado por

su enfoque dentro del paradigma orientado a objetos y/o ciclos de vida iterativo-

incrementales. Este proceso se lleva a cabo mediante una serie de pasos. (Vea en el

Anexo 2. Estimación del Esfuerzo, Tiempo de desarrollo y Costo).

### II.2.5.1 Cálculo de los puntos de Caso de Uso

Son el resultado de la suma del Factor de Peso de los Actores y el Factor de Peso de

los Casos de Uso por lo que se procedió al cálculo de los mismos

### II.2.5.2 Cálculo del factor de peso de los actores

Se determina teniendo en cuenta la cantidad de actores y su complejidad, un actor

puede ser simple, medio o complejo y sus valores de complejidad son respectivamente

1, 2 y 3. Un actor tiene como valor de complejidad 1 cuando este es un sistema

mediante una interfaz de programación (API, Web Service), 2 cuando es un sistema

mediante un protocolo o una interfaz basada en texto y 3 cuando el actor es una

persona que interactúa con el sistema mediante una interfaz gráfica. Este proyecto

posee 1 actor de complejidad baja 1 y 2 de complejidad alta por tanto su FPA tiene

un valor de 7.

### II.2.5.3 Cálculo del factor de peso de caso de uso sin ajustar

Se determina teniendo en cuenta la cantidad de casos de uso y su complejidad o peso.

La complejidad de un caso de uso se determina a partir de la cantidad de transacciones

que posee, siendo una transacción una secuencia atómica de actividades, las cuales

se realizan completamente o no se realiza ninguna. Un caso de uso será de tipo

Simple cuando posee menos de 4 transacciones con un peso asociado de 5, Medio

cuando posee de 4 a 7 transacciones con un peso asociado de 10 o Complejo cuando

posee más de 7 transacciones con un peso asociado de 15 por tanto su FPCU tiene

un valor de 160.

### II.2.5.4 Cálculo de los puntos de Caso de Uso Ajustado

Después de calculados los PCU (sin ajustar) estos fueron ajustados teniendo en

cuenta un grupo de factores técnicos y ambientales. El valor de los PCUA se obtiene

como resultado de la multiplicación del valor de los Puntos de Casos de Uso (obtenido

en el paso 1), el Factor de Complejidad Técnica y el Factor de Ambiente.

### II.2.5.5 Cálculo de factor de complejidad técnica

Se estimó mediante la cuantificación del peso de un grupo de factores que determinan

la complejidad técnica del software asignándole a cada factor un valor de 0 a 5 de

acuerdo con la relevancia que este tenga.

Una vez concluido esto, mediante el uso de la fórmula para la obtención del FCT, se

obtuvo su valor, siendo el mismo 0.95.

### II.2.5.6 Cálculo del factor ambiente

Se estimó mediante la cuantificación del peso de un grupo de factores vinculados a las

habilidades, entrenamientos y experiencias del grupo de desarrollo, asignándole a

cada factor un valor de 0 a 5 de acuerdo con la relevancia que este tenga. Concluido

esto, aplicando la fórmula del FA se obtuvo su valor, siendo el mismo 0.75.

Una vez obtenido el valor del FCT (0.95), el del FA (0.755) y conocido el valor del PCU,

al efectuar la multiplicación resulta como valor del PCUA 122

### *II*.2.5.7 Cálculo del Esfuerzo de Implementación (E)

Para calcularlo se convirtieron los Puntos de Casos de Uso Ajustados a esfuerzo de

desarrollo, multiplicando el valor del PCUA obtenido anteriormente por el Factor de

Conversión (FC) el cual se estima de 20 H/H obteniéndose como resultado un esfuerzo

de 2440 H/H.

### II.2.5.8 Cálculo del Esfuerzo de Total (ET)

Conocidos los porcientos aproximados del tiempo que requieren las etapas de

desarrollo de software (Análisis, Diseño, Implementación, Pruebas y Otras

Actividades) se obtuvo el esfuerzo total, siendo el mismo 6100 H/H.

Cálculo del Tiempo de Desarrollo (TDES)

Fue obtenido dividiendo el esfuerzo total obtenido en el paso anterior, por la cantidad

de hombres a participar en la construcción de la aplicación, siendo el mismo de 6100

horas.

Cálculo del Costo Total

El mismo fue obtenido multiplicando el esfuerzo total (obtenido en el paso 4) con el

Costo por hombres horas.

# CAPITULO III: DESCRIPCION DE LA SOLUCION PROPUESTA.

En este capítulo se abordan aspectos relacionados con el análisis y diseño de la solución propuesta, basado en la realización de los casos de uso del diseño (diagrama de interacción y diagrama de clases del diseño), en el modelo de datos, el modelo de arquitectura empleado y patrón de diseño aplicado, así como el diagrama de despliegue generado para modelar la propuesta de implementación del software en la entidad.

## III.1 Realización de Caso de Uso de Diseño

### III.1.1 Diagrama de Secuencias

Un diagrama de secuencias es un tipo de diagrama UML (Unified Modeling Language) que muestra cómo interactúan los objetos en un sistema a lo largo del tiempo. Este diagrama se utiliza principalmente para modelar la lógica de un escenario específico, describiendo el orden de los mensajes intercambiados entre los objetos para llevar a cabo una función o proceso. ( Lucid Software Inc, 2024)



Ilustración 3 Diagrama de Secuencia de login de un usuario o del administrador



Ilustración 4 Diagrama de Secuencia de obtener desglose de Proyectos



Ilustración 5 Diagrama de Secuencia de crear departamento

### III.1.2 Diagramas de clases de caso de uso

Un diagrama de clases es un tipo de diagrama en el Lenguaje Unificado de Modelado (UML) que ilustra la estructura estática de un sistema. Muestra las clases del sistema, sus atributos, operaciones (o métodos) y las relaciones entre los objetos. Aquí tienes los componentes clave:

**Clases:** Representan entidades o conceptos en el sistema. Por ejemplo, en un sistema de biblioteca, podrías tener clases como "Libro", "Usuario" y "Bibliotecario".

**Atributos:** Propiedades o características de una clase. Por ejemplo, la clase "Libro" podría tener atributos como "título", "autor" y "ISBN".

**Operaciones:** Funciones o métodos que las clases pueden realizar. Por ejemplo, la clase "Usuario" podría tener operaciones como "registrarse()" o "pedirPrestamo()".

**Relaciones:** Incluyen asociaciones (cómo las clases están conectadas entre sí), herencia (cuando una clase es una subclase de otra) y dependencias (cuando una clase utiliza a otra). (wikipedia, 2019)



Ilustración 6 Diagrama de clase gestionar Usuario



Ilustración 7Diagrama de clase Gestionar Departamento



Ilustración 8 Diagrama de clase Obtener Estadísticas

## III.2 Modelo de Datos

**Un modelo de datos físicos se define como una representación del modelo que se usará para la construcción de la base de datos.** Entre los modelos de datos, el modelo de datos físico es el último peldaño para poder crear una base de datos. Están: el modelo conceptual, el modelo de datos lógico y, por último, el modelo de datos físicos.

Al ser el último en el modelado de datos, el modelo físico posee amplia información sobre la futura base de datos, es decir, brinda un contexto más específico. De este modo, se muestra la estructura de las diferentes tablas y los componentes de las mismas, en donde se incluye el **nombre de las columnas, el tipo de datos de cada una de ellas y sus restricciones, las claves primarias y claves externas, las distintas relaciones entre tablas, los índices y los procedimientos dentro de la base de datos.** (KeepCoding®, 2024)

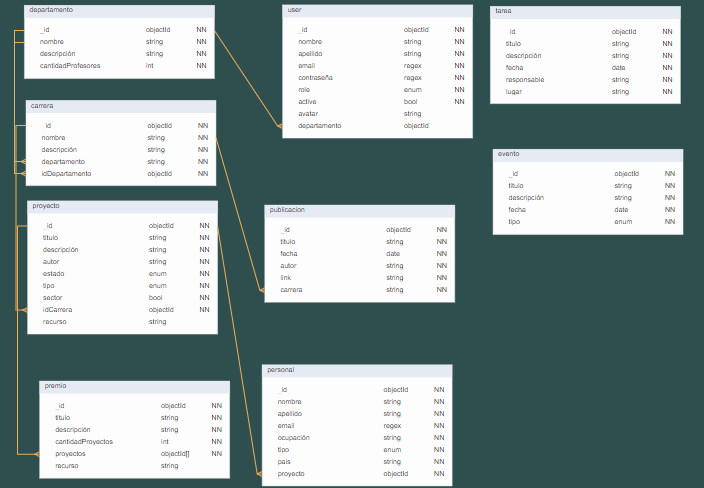


Ilustración Diagrama de Modelo de Datos Físico

## III.3 Definición del modelo de Arquitectura

Un modelo de arquitectura se refiere a una representación estructural de un sistema de software. Este modelo describe cómo los componentes del sistema interactúan entre sí y cómo se organizan para cumplir con los requisitos del sistema.

Algunos ejemplos comunes de modelos de arquitectura en software incluyen:

**Arquitectura en capas:** Divide el sistema en capas, cada una con una responsabilidad específica, como la capa de presentación, la capa de lógica de negocio y la capa de datos.

**Arquitectura de microservicios:** Descompone el sistema en pequeños servicios independientes que se comunican entre sí a través de APIs.

**Arquitectura orientada a servicios (SOA):** Similar a los microservicios, pero con un enfoque en servicios más grandes y reutilizables.

**Arquitectura de cliente-servidor:** Divide el sistema en clientes que solicitan servicios y servidores que proporcionan esos servicios que es el que usaremos en este proyecto.

### III.3.1 Patrón Cliente-Servidor

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas o cargas de trabajo se reparten entre los proveedores de recursos o servicios, llamados servidores, y los solicitantes de servicios, llamados clientes. Este modelo es muy común en aplicaciones de red y sistemas distribuidos.

Ventajas de la arquitectura cliente-servidor:

**Escalabilidad:** Es fácil agregar más servidores para manejar un mayor volumen de solicitudes de los clientes, lo que permite que el sistema crezca según sea necesario1.

**Centralización de recursos:** Los recursos comunes a todos los usuarios se administran en el servidor, lo que facilita la gestión y el mantenimiento.

**Seguridad mejorada:** Los servidores pueden ser monitoreados y administrados de manera centralizada, lo que permite implementar medidas de seguridad más robustas.

**Reducción de costos:** Al utilizar un único servidor para proporcionar servicios a varios usuarios, los costos son menores en comparación con tener múltiples sistemas independientes.

**Facilidad de mantenimiento:** Las actualizaciones y el mantenimiento se pueden realizar en el servidor sin afectar directamente a los clientes, lo que reduce el tiempo de inactividad y las interrupciones. (© Arsys, 2021)

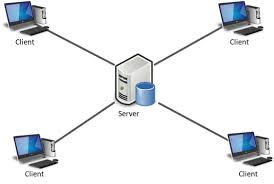


Ilustración 10 Representación de la Arquitectura

## III.4 Diagrama de Despliegue

Un diagrama de despliegue es una representación visual que muestra cómo se distribuyen los componentes de un sistema en diferentes nodos físicos, como servidores, computadoras y otros dispositivos de hardware1

. Este tipo de diagrama es parte del Lenguaje Unificado de Modelado (UML) y se utiliza para ilustrar la arquitectura de ejecución de un sistema, incluyendo tanto el hardware como el software2

. Elementos de un Diagrama de Despliegue

Nodos: Representan los entornos de ejecución físicos, como servidores, computadoras y otros dispositivos3

. Componentes: Muestran los artefactos de software que se despliegan en los nodos4

. Conectores: Indican las relaciones y comunicaciones entre los componentes y los nodos5

. Middleware: Representa el software que facilita la comunicación entre diferentes componentes y nodos6

Ejemplo de Uso:

Imagina que estás desarrollando una aplicación web. Un diagrama de despliegue te ayudaría a visualizar cómo se distribuyen los servidores web, bases de datos y otros componentes en la infraestructura física. Esto es crucial para entender cómo interactúan los diferentes elementos y para identificar posibles problemas de configuración o rendimiento. (Creately, 2019)



Ilustración 11 Diagrama de Despliegue

## III.5 Principios de Diseño de Implementación en la Aplicación

En el proceso de creación de software, es crucial considerar los principios de diseño que definirán las características finales del producto. Un diseño coherente y consistente en un software es esencial para lograr una buena aceptación por parte de los usuarios. Una aplicación informática debe contar con un diseño gráfico atractivo, buena navegabilidad, usabilidad y una distribución adecuada de los contenidos. Para alcanzar este objetivo, en el diseño de la interfaz de usuario del sistema informático, se han considerado los siguientes principios:

**Visibilidad del estado del sistema:** Mantener al usuario siempre informado.

**Correspondencia con el mundo real:** Utilizar un lenguaje familiar para el usuario.

**Control y libertad del usuario:** Permitir al usuario controlar la interacción.

**Consistencia y estándares:** Mantener una coherencia en todo el sistema.

**Prevención de errores:** Ofrecer opciones de confirmación antes de acciones importantes.

**Reconocimiento en lugar de memorización:** Hacer visibles las instrucciones de uso.

**Flexibilidad y eficiencia:** Ofrecer múltiples formas de realizar acciones e información.

**Diseño estético y minimalista:** Utilizar solo los elementos esenciales.

**Ayuda a los usuarios:** Mensajes de error en lenguaje claro y comprensible.

### III.5.1 Interfaz de Usuario

Tema Claro

Colores: El tema claro utiliza tonos grises claros y blancos (#ffffff, #f6f6f6, #f0f0f0) como fondo principal, proporcionando una apariencia limpia y brillante. Los botones y elementos destacados utilizan la paleta secundaria amarilla (#ffd166, #ffda85, #ffedc2) para asegurar un contraste visual efectivo y resaltar acciones importantes.

Contraste: Los textos y otros elementos interactivos utilizan colores secundarios amarillos (#d3d4de, #a6a9be, #7a7f9d) para asegurar que sean fácilmente visibles sobre el fondo claro.

Iconos y Tooltips: Los botones en el menú lateral y la barra de navegación están acompañados de iconos intuitivos para facilitar la navegación. Además, los tooltips proporcionan información adicional cuando el usuario pasa el cursor sobre estos iconos, mejorando la experiencia de uso.

Mensajes de Error: Los formularios y el tratamiento de errores incluyen mensajes de error en rojo, asegurando que los usuarios reciban una alerta clara y visible cuando ocurre un problema.

Tema Oscuro

Colores: En el tema oscuro, los fondos utilizan tonos de la paleta azul (#21295c, #191F45, #141937), creando una apariencia elegante y moderna. Los textos y elementos interactivos utilizan tonos grises y negros (#666666, #525252, #3d3d3d) para proporcionar un contraste claro y asegurar la legibilidad.

Contraste: Los botones y elementos destacados también utilizan la paleta secundaria amarilla (#ffd166, #ffda85, #ffedc2) para resaltar y mantener la coherencia en ambos temas.

Iconos y Tooltips: Similar al tema claro, los botones en el menú lateral y la barra de navegación tienen iconos intuitivos, y los tooltips ayudan a los usuarios proporcionando descripciones útiles al pasar el cursor.

Mensajes de Error: Los mensajes de error en los formularios y durante el tratamiento de errores se muestran en rojo, asegurando que sean visibles sobre el fondo oscuro.

Funcionalidad

Navegabilidad y Usabilidad: La interfaz está diseñada para ser fácil de navegar y usar, con una disposición clara de los contenidos y elementos interactivos bien definidos. Los iconos y tooltips mejoran la intuición y comprensión de la interfaz, haciendo que sea accesible para todos los usuarios.

Estética y Consistencia: Ambas paletas de colores y el uso de iconos y tooltips están seleccionadas para mantener una apariencia coherente y agradable, siguiendo principios de diseño minimalista y estético. Los mensajes de error en rojo garantizan que los problemas sean fácilmente identificados y abordados.



Ilustración 12 Interfaz Oscura

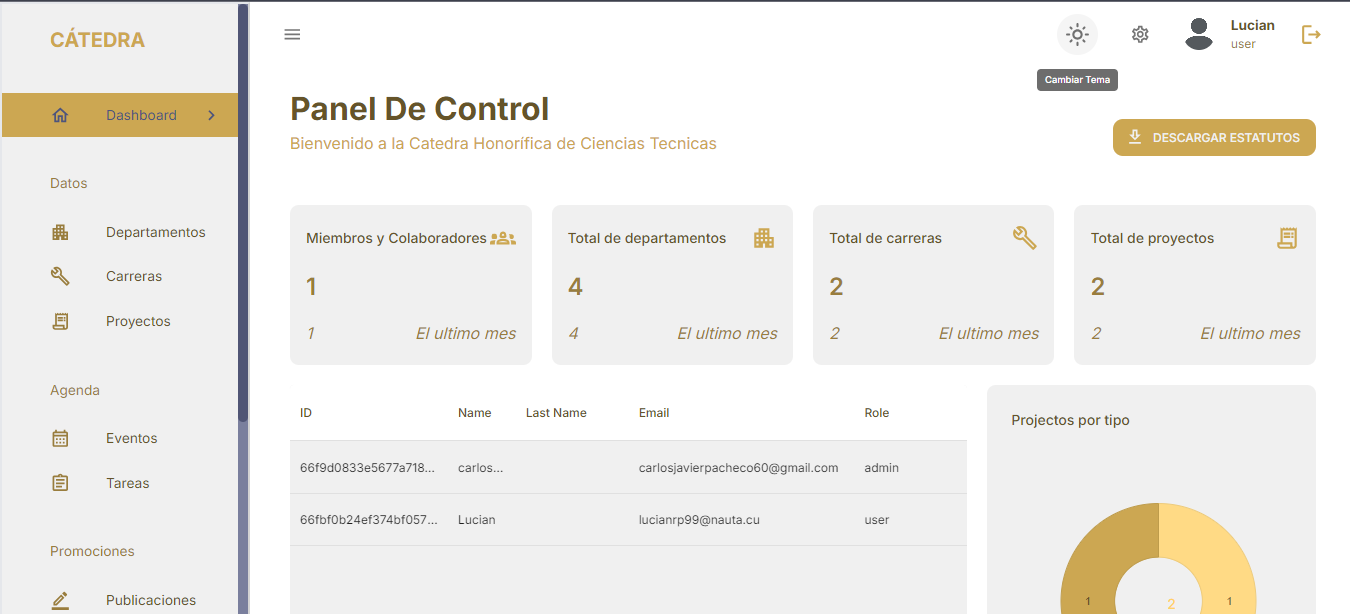


Ilustración 13 Interfaz Clara

### III.5.2 Formato de Salida de los Reportes

El formato de salida de los reportes se diseñó con un enfoque en la accesibilidad y la funcionalidad, permitiendo a los usuarios interactuar de manera intuitiva y eficiente con los datos presentados.

Características Principales

Filtros por Columna: Todas las tablas en los reportes incluyen opciones de filtrado por columna, lo que permite a los usuarios ordenar y seleccionar datos específicos según sus necesidades. Los filtros son fáciles de usar y permiten una rápida reorganización de la información para obtener insights precisos y personalizados.

Exportación a Excel: La tabla de proyectos presenta la opción de exportar los datos en formato Excel. Esta funcionalidad permite a los usuarios guardar los reportes localmente, facilitando el análisis adicional, la manipulación de datos y el archivo de la información para su uso futuro.

Impresión de Tablas: Además de la exportación, los usuarios pueden imprimir directamente las tablas desde la interfaz de usuario. La opción de impresión está optimizada para presentar los datos de manera clara y legible en papel, asegurando que la información impresa mantenga la coherencia y la estructura del reporte digital.

Usabilidad y Navegación

Interfaz Intuitiva: La interfaz de usuario está diseñada para ser altamente intuitiva, con iconos y tooltips que facilitan el uso de las funciones de filtrado, exportación e impresión. Los elementos interactivos están claramente marcados y son fácilmente accesibles, garantizando una experiencia de usuario sin complicaciones.

Mensajes de Confirmación: Antes de realizar acciones críticas, como exportar o imprimir, se muestran mensajes de confirmación para evitar errores y asegurar que los usuarios estén seguros de sus decisiones.

Consistencia y Estética

Diseño Consistente: El diseño de los reportes mantiene una apariencia coherente con el resto de la interfaz del sistema, utilizando las mismas paletas de colores y estilos gráficos. Esto asegura que los reportes sean no solo funcionales, sino también visualmente agradables y en línea con la estética general de la aplicación.

Accesibilidad: Se ha tenido en cuenta la accesibilidad en todo el proceso de diseño, asegurando que los usuarios con diferentes niveles de habilidad técnica puedan utilizar todas las funciones de los reportes sin dificultad.

## III.6 Diagrama de Componentes

Un diagrama de componentes es un tipo de diagrama del Lenguaje Unificado de Modelado (UML) que se utiliza para representar la estructura física de un sistema de software. Este diagrama muestra cómo un sistema está dividido en componentes y las relaciones y dependencias entre estos componentes . Elementos de un Diagrama de Componentes:

**Componentes:** Representan unidades funcionales del sistema, como módulos, clases, o archivos.

**Interfaces:** Definen las conexiones entre los componentes y cómo interactúan entre sí.

**Puertos:** Indican los puntos de entrada y salida de los componentes.

**Conectores:** Muestran las relaciones y comunicaciones entre los componentes.

Propósito

El propósito principal de un diagrama de componentes es proporcionar una visión clara y estructurada de cómo se organiza un sistema de software, facilitando la comunicación y comprensión entre los desarrolladores y otros interesados.



Ilustración 14 Diagrama de componentes Añadir departamento



Ilustración 15 Diagrama de componentes Obtener Geografía



Ilustración 16 Diagrama de componentes Obtener Avatar del usuario

## III.7 Sistema de Seguridad

La seguridad de los datos y su integridad son conceptos relacionados pero distintos. Mientras que la integridad se enfoca en la precisión y validez de la información, la seguridad se trata de protegerla contra accesos no autorizados, alteraciones o destrucciones. La seguridad de la información implica implementar estrategias que protejan los procesos donde la información es un activo crucial. Estas estrategias incluyen el establecimiento de políticas, controles de seguridad, tecnologías y procedimientos diseñados para detectar y mitigar amenazas que puedan explotar vulnerabilidades y poner en riesgo la información. En otras palabras, buscan proteger tanto la información como los sistemas que la almacenan y administran.

En el sistema, hay dos tipos de usuarios: el Administrador y el Usuario. El Administrador es responsable de otorgar permisos a las cuentas de los usuarios para que puedan interactuar con la plataforma. Un sistema de seguridad robusto asegura la protección de los datos, sin importar cómo los usuarios obtienen acceso a estos.

El primer paso del proceso de seguridad es la Autenticación, que obliga al usuario a presentar algún tipo de identificación para verificar su identidad. Una vez autenticado, el siguiente paso es la Autorización, que determina si el usuario tiene los privilegios necesarios para acceder a un recurso específico.

Se utilizó la metodología de encriptación AES(Advanced Encryption Standard), el cual se usa con el fin de cifrar datos y de protegerlos contra cualquier acceso ilícito.

En la aplicación, se utiliza la librería bcrypt.js para encriptar las contraseñas de los usuarios, garantizando su protección contra accesos no autorizados. Además, se implementó JSON Web Token (JWT) para enviar la información de los usuarios encriptada, asegurando que los datos transmitidos sean seguros y privados.

Este enfoque integral de seguridad no solo protege la información, sino que también asegura que los usuarios tengan una experiencia segura y confiable al interactuar con la plataforma.

### III.7.1 Tratamiento de Errores

El tratamiento de errores en una aplicación web es fundamental para asegurar una experiencia de usuario eficiente y sin frustraciones. Un manejo adecuado de errores ayuda a identificar y solucionar problemas rápidamente, lo que mejora la estabilidad y la seguridad del sistema. Además, proporciona feedback claro y útil a los usuarios, permitiéndoles entender y corregir sus acciones. Esto no solo aumenta la confianza en la aplicación, sino que también reduce el riesgo de fallos críticos y potenciales vulnerabilidades. En resumen, un buen manejo de errores es esencial para mantener una aplicación web funcional, segura y user-friendly.

A continuación, se muestra un fragmento de código donde se valida que todos los campos sean obligatorios.

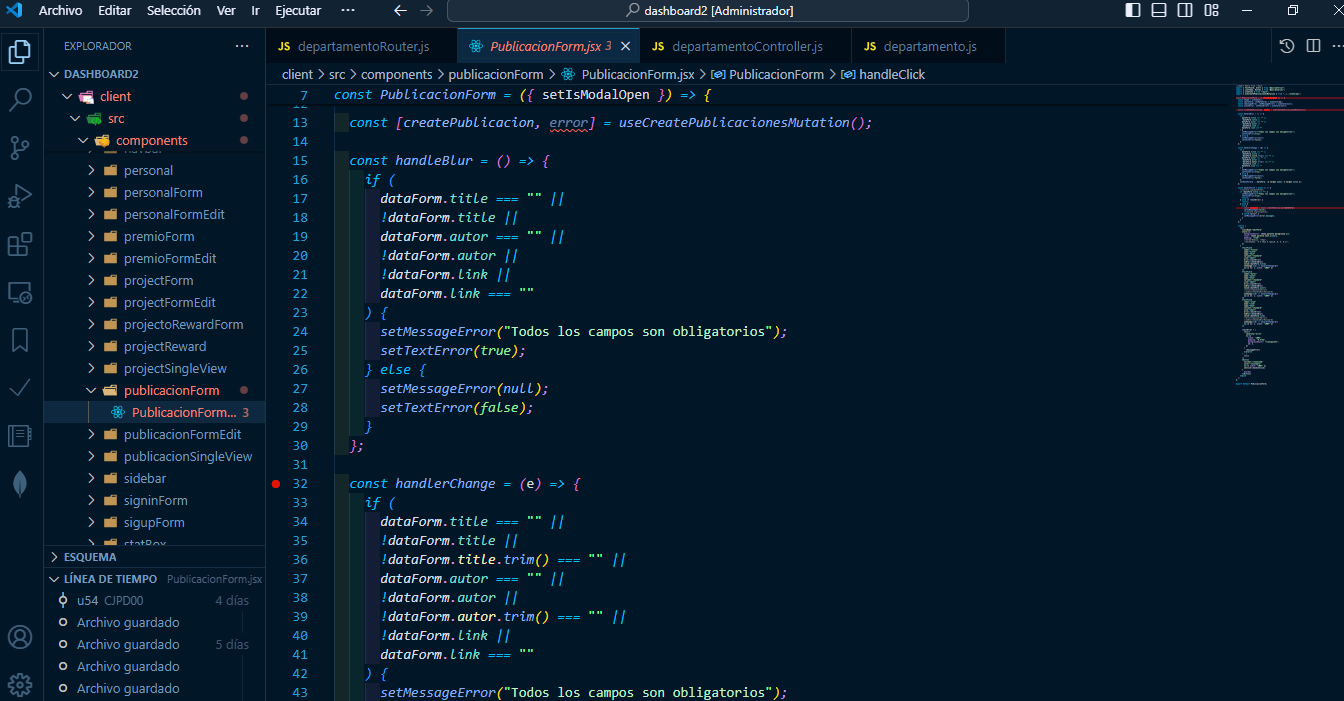


Ilustración 17 Ejemplo de validación de campos obligatorios

Seguidamente otro ejemplo, de que el email sea un email valido:

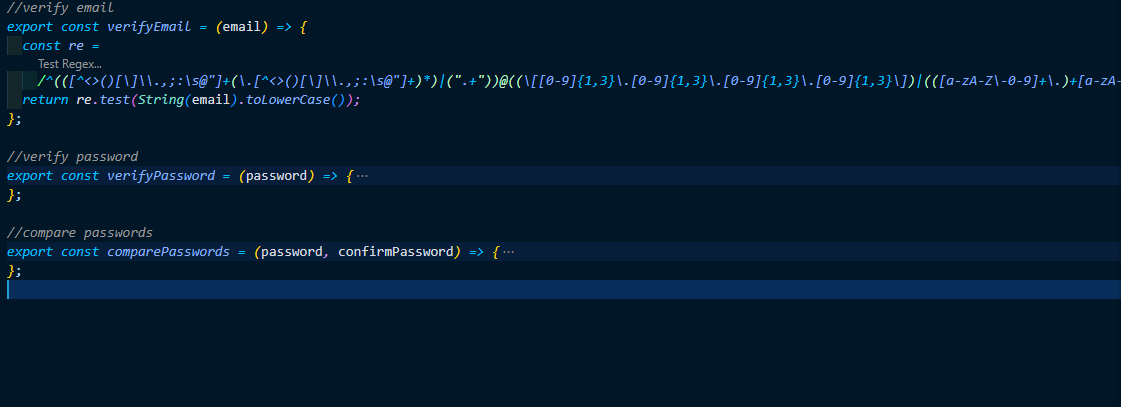


Ilustración 18 Ejemplo de validación del email

## III.8 Prueba

Luego de realizada la implementación y tratados los errores de la aplicación se

procedió a realizarle pruebas a la aplicación para comprobar el correcto

funcionamiento de esta, a través de casos de pruebas.

|  |
| --- |
| Caso de uso: Gestionar departamento (Insertar) |
| Caso de Prueba: Añadir departamento |
| Descripción |
| * El usuario entra a la pantalla de Añadir departamento, llena los campos requeridos y presiona el botón guardar. |
| Condiciones de ejecución |
| * El usuario debe estar autenticado. |
| Entrada |
| * Se escribe “infórmatica” en el input nombre. * Se selecciona o se escribe “10” en el input de tipo número. * Se escribe “aaaaaaaaaa” en el input de tipo textArea. * Se da clic en el botón “guardar” para registrarlo. |
| Resultado esperado |
| * Mostrar la carta del departamento en la interfaz |
| Evaluación de la prueba |
| Prueba superada con éxito. |

Tabla 2 Descripción de Prueba Añadir departamento

|  |
| --- |
| Caso de uso: Gestionar Tarea (Eliminar) |
| Caso de Prueba: Eliminar Tarea |
| Descripción |
| * El usuario entra a la pantalla de tareas selecciona una fila de la tabla tareas y da clic en la opción de eliminar |
| Condiciones de ejecución |
| * El usuario debe estar autenticado. * La tarea debe existir |
| Entrada |
| * Seleccionar tarea * Clic en eliminar |
| Resultado esperado |
| * Mostrar la tabla de tareas actualizada |
| Evaluación de la prueba |
| Prueba superada con éxito. |

Tabla 3 Descripción de Prueba eliminar tarea

|  |
| --- |
| Caso de uso: Gestionar Tarea (Editar) |
| Caso de Prueba: Editar Tarea |
| Descripción |
| * El usuario entra a la pantalla de tareas selecciona una fila de la tabla tareas y da clic en la opción de editar |
| Condiciones de ejecución |
| * El usuario debe estar autenticado. * La tarea debe existir |
| Entrada |
| * Se escribe “limpiar” en el input de tipo text Título * Se escribe “asfasfsaf” en el input de tipo textArea Descripción * Se escribe “juan” en el input de tipo text Responsable * Se escribe “parque” en el input de tipo text Lugar * Se selecciona la fecha en el dataPicker Fecha * Presiona el botón guardar |
| Resultado esperado |
| * Mostrar la tabla de tareas actualizada |
| Evaluación de la prueba |
| Prueba superada con éxito. |

Tabla 4 Descripción de Prueba editar tarea

# CONCLUSIONES.

Se identificaron avances significativos en el uso de tecnologías de la información en entornos académicos, destacando la importancia de las plataformas web para mejorar la gestión de información y facilitar la colaboración. Las plataformas similares estudiadas demostraron que su adopción incrementa la eficiencia operativa y promueve la participación activa de los miembros de las cátedras. Esto refuerza la necesidad de desarrollar soluciones adaptadas a contextos específicos como el de la cátedra honorífica.

Las entrevistas y encuestas realizadas revelaron necesidades específicas en cuanto a la gestión de datos y la coordinación de actividades. Los miembros de la cátedra valoran la centralización de la información y la integración de herramientas que faciliten tanto la comunicación como la colaboración en tiempo real. La mayoría mostró preferencia por una interfaz intuitiva y accesible que no requiera una curva de aprendizaje elevada.

Los requerimientos documentados subrayan la necesidad de una plataforma segura y escalable, con funciones claras para la gestión de datos y la coordinación de actividades. Los criterios de usabilidad y accesibilidad establecidos resultaron esenciales para garantizar que la plataforma fuera inclusiva, fácil de usar y adaptable a las distintas necesidades de los miembros de la cátedra.

Tras evaluar diversas tecnologías y frameworks web, se determinó que la combinación de tecnologías como React, Node.js y MongoDB proporciona una solución óptima en términos de seguridad, escalabilidad y compatibilidad. Las mejores prácticas de UI/UX evaluadas fueron clave para diseñar una experiencia de usuario fluida y agradable, alineada con las expectativas de los usuarios.

Se desarrolló una arquitectura robusta que integra una estructura de base de datos eficiente, módulos organizados para la gestión de la información y una lógica de negocio clara. El diseño de un esquema de seguridad adecuado garantiza la protección de datos sensibles, lo cual es fundamental en un entorno académico donde la privacidad y la integridad de la información son críticas.

La implementación de la interfaz de usuario siguió estrictamente las especificaciones del diseño, asegurando que todas las funciones requeridas (gestión de información, coordinación de actividades y comunicación) estuvieran operativas. Las tecnologías de seguridad integradas resultaron efectivas, garantizando la protección frente a accesos no autorizados y la preservación de la confidencialidad de la información.

El desarrollo de la plataforma web ha permitido una mejora sustancial en la gestión de información y la colaboración dentro de la cátedra honorífica. Los avances en la implementación de tecnologías y el diseño de una arquitectura segura y accesible contribuyeron a una solución integral que satisface las necesidades académicas actuales. La plataforma ha demostrado ser una herramienta eficaz para optimizar las operaciones y facilitar la comunicación entre los miembros.

# RECOMENDACIONES.

1. **Mejora continua basada en la retroalimentación:** Es esencial establecer un proceso continuo de recopilación de retroalimentación de los usuarios de la cátedra honorífica. Las opiniones y sugerencias obtenidas pueden ayudar a identificar áreas de mejora en la interfaz, la funcionalidad y la experiencia de usuario. Implementar un sistema de seguimiento de incidencias y solicitudes de nuevas características sería clave para mantener la plataforma actualizada y ajustada a las necesidades emergentes.
2. **Escalabilidad del sistema:** A medida que la cátedra crece y se incorporan nuevos miembros, la plataforma debe estar preparada para manejar un mayor volumen de usuarios y datos. Se recomienda monitorear el rendimiento y realizar pruebas periódicas de carga para asegurar que el sistema pueda escalar sin afectar la experiencia de usuario. Considerar la posibilidad de implementar soluciones de infraestructura en la nube, como servidores escalables y bases de datos distribuidas, que permitan gestionar el crecimiento sin problemas.
3. **Actualización tecnológica constante:** La tecnología avanza rápidamente, por lo que es crucial realizar evaluaciones periódicas de las herramientas y frameworks utilizados (React, Node.js, MongoDB, etc.). La adopción de nuevas versiones o tecnologías más eficientes puede mejorar el rendimiento, la seguridad y la funcionalidad de la plataforma. Se recomienda planificar un ciclo de actualización tecnológica para mantener la plataforma competitiva y en sintonía con los avances del mercado.
4. **Seguridad proactiva:** La seguridad debe seguir siendo una prioridad, especialmente con el crecimiento de la base de usuarios y el almacenamiento de más información sensible. Se recomienda implementar auditorías de seguridad regulares, realizar análisis de vulnerabilidades, y aplicar parches de seguridad tan pronto como estén disponibles. Además, se podría considerar la implementación de autenticación multifactor (MFA) y el cifrado de datos en reposo para reforzar aún más la protección de la información.
5. **Modularidad para futuras expansiones:** El diseño modular del sistema permite agregar nuevas funcionalidades sin afectar las existentes. Es recomendable seguir utilizando esta estrategia para futuras expansiones, como la integración de nuevas herramientas colaborativas, gestión avanzada de proyectos o funcionalidades para análisis de datos. Esta estructura permitirá que la plataforma se adapte a los cambios en las necesidades de la cátedra sin tener que realizar rediseños profundos.
6. **Integración con otras plataformas académicas:** Para mejorar la interoperabilidad y facilitar el uso de la plataforma, se recomienda explorar la posibilidad de integrar la solución con otras plataformas académicas o de gestión institucional (por ejemplo, sistemas de gestión de aprendizaje o herramientas de videoconferencia). Esto permitiría a los usuarios acceder a múltiples servicios desde un solo entorno y simplificaría la administración de diversas actividades académicas.
7. **Fomento de la adopción y el uso adecuado:** Se sugiere organizar capacitaciones periódicas para los nuevos miembros de la cátedra y desarrollar guías de uso actualizadas para asegurar una correcta adopción de la plataforma. Además, ofrecer talleres o seminarios sobre el uso avanzado de las funciones puede incentivar a los usuarios a sacar el máximo provecho de la herramienta y descubrir características que optimicen su trabajo colaborativo y gestión de información.
8. **Análisis de datos e informes personalizados:** Incorporar funciones de análisis de datos e informes personalizados puede ofrecer un valor añadido significativo. Estas herramientas permitirían a los administradores de la cátedra monitorear el uso de la plataforma, analizar la eficiencia en la gestión de proyectos y tomar decisiones informadas basadas en datos. Se recomienda desarrollar un módulo para generar informes automáticos que puedan ser personalizados según las necesidades específicas de la cátedra.

# REFERENCIAS

Lucid Software Inc. (6 de 1 de 2024). *Lucid Software Inc*. Obtenido de Lucid Software Inc web site: https://www.lucidchart.com/pages/es/diagrama-de-secuencia

© Arsys. (10 de 12 de 2021). *© Arsys*. Obtenido de © Arsys web site: https://www.arsys.es/blog/todo-sobre-la-arquitectura-cliente-servidor

acimed. (1 de 1 de 2020). *acimed*. Obtenido de acimed web site: https://acimed.sld.cu/index.php/acimed/article/view/1455/991

Creately. (10 de 4 de 2019). *Creately*. Obtenido de Creately web site: https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/

IBM. (3 de 5 de 2022). *IBM we site*. Obtenido de IBM we site: https://www.ibm.com/es-es/topics/business-rules

KeepCoding®. (18 de 3 de 2024). *KeepCoding®*. Obtenido de KeepCoding® web site: https://keepcoding.io/blog/modelo-de-datos-fisicos/

UCI. (1 de 1 de 2022). *repositorio uci*. Obtenido de repositorio uci web site: https://repositorio.uci.cu/jspui/bitstream/123456789/10572/1/TD\_9846.pdf

Visure Solutions, Inc. (2 de 5 de 2020). *Visure Solutions, Inc.* Obtenido de Visure Solutions, Inc. web site: https://visuresolutions.com/es/blog/requerimientos-funcionales/

Visure Solutions, Inc. (2 de 5 de 2020). *Visure Solutions, Inc.* Obtenido de Visure Solutions, Inc. web site: https://visuresolutions.com/es/blog/requerimientos-no-funcionales/

wikipedia. (3 de 4 de 2019). *wikipedia*. Obtenido de wikipedia web site: https://es.wikipedia.org/wiki/Diagrama\_de\_clases

wikipedia. (1 de 10 de 2023). *wikipedia* . Obtenido de wikipedia web site: https://es.wikipedia.org/wiki/Modelo\_de\_dominio

# BIBLIOGRAFÍA

Lucid Software Inc. (6 de 1 de 2024). *Lucid Software Inc*. Obtenido de Lucid Software Inc web site: https://www.lucidchart.com/pages/es/diagrama-de-secuencia

© Arsys. (10 de 12 de 2021). *© Arsys*. Obtenido de © Arsys web site: https://www.arsys.es/blog/todo-sobre-la-arquitectura-cliente-servidor

acimed. (1 de 1 de 2020). *acimed*. Obtenido de acimed web site: https://acimed.sld.cu/index.php/acimed/article/view/1455/991

Creately. (10 de 4 de 2019). *Creately*. Obtenido de Creately web site: https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/

IBM. (3 de 5 de 2022). *IBM we site*. Obtenido de IBM we site: https://www.ibm.com/es-es/topics/business-rules

KeepCoding®. (18 de 3 de 2024). *KeepCoding®*. Obtenido de KeepCoding® web site: https://keepcoding.io/blog/modelo-de-datos-fisicos/

UCI. (1 de 1 de 2022). *repositorio uci*. Obtenido de repositorio uci web site: https://repositorio.uci.cu/jspui/bitstream/123456789/10572/1/TD\_9846.pdf

Visure Solutions, Inc. (2 de 5 de 2020). *Visure Solutions, Inc.* Obtenido de Visure Solutions, Inc. web site: https://visuresolutions.com/es/blog/requerimientos-funcionales/

Visure Solutions, Inc. (2 de 5 de 2020). *Visure Solutions, Inc.* Obtenido de Visure Solutions, Inc. web site: https://visuresolutions.com/es/blog/requerimientos-no-funcionales/

wikipedia. (3 de 4 de 2019). *wikipedia*. Obtenido de wikipedia web site: https://es.wikipedia.org/wiki/Diagrama\_de\_clases

wikipedia. (1 de 10 de 2023). *wikipedia* . Obtenido de wikipedia web site: https://es.wikipedia.org/wiki/Modelo\_de\_dominio

# ANEXOS.

## Anexo 1. Descripción de bajo nivel de todos los casos de uso del sistema

|  |  |
| --- | --- |
| Caso de Uso | Gestión de Departamentos |
| Actor(es) | 1-Administrador  2-usuario activo |
| Propósito | Gestionar la información de los departamentos existentes |
| Descripción | 1- El actor pulsa el botón Agregar departamento  2- Introduce los datos en los componentes.  3- Pulsa el botón guardar |
| Referencias | R2. |
| Precondiciones | El actor ha ingresado al sistema y se encuentra en la ventana de departamentos |
| Poscondiciones | La información que contiene la base de datos relacionada a los departamentos queda actualizada |
| Acción del Actor | Respuesta del Sistema |
|  |  |
| Otras acciones | - |
| Requerimientos especiales | - |

Tabla 5 Descripción de Caso de Uso Gestión de Departamentos.

|  |  |
| --- | --- |
| Caso de Uso | Gestión de Geografía |
| Actor(es) | 1-Administrador  2-usuario activo |
| Propósito | Obtener la estadística de la ubicación del personal de los proyectos |
| Descripción | 1- El actor pulsa el botón de menú geografía |
| Referencias | R5. |
| Precondiciones | El actor ha ingresado al sistema y se encuentra en la ventana de geografía |
| Poscondiciones | La información que contiene la base de datos relacionada al personal se procesa |
| Acción del Actor | Respuesta del Sistema |
|  |  |
| Otras acciones | - |
| Requerimientos especiales | - |

Tabla 6 Descripción de Caso de Uso Gestión de Geografía.

|  |  |
| --- | --- |
| Caso de Uso | Gestión de Publicaciones |
| Actor(es) | 1-Administrador  2-usuario activo |
| Propósito | Gestionar la información de las publicaciones existentes |
| Descripción | 1- El actor pulsa el botón Agregar publicación  2- Introduce los datos en los componentes.  3- Pulsa el botón guardar |
| Referencias | R10. |
| Precondiciones | El actor ha ingresado al sistema y se encuentra en la ventana de publicaciones |
| Poscondiciones | La información que contiene la base de datos relacionada a las publicaciones queda actualizada |
| Acción del Actor | Respuesta del Sistema |
|  |  |
| Otras acciones | - |
| Requerimientos especiales | - |

Tabla 7 Descripción de Caso de Uso Gestión de Publicaciones

|  |  |
| --- | --- |
| Caso de Uso | Gestión de Personal |
| Actor(es) | 1-Administrador  2-usuario activo |
| Propósito | Gestionar la información del personal de un proyecto existente |
| Descripción | 1- El actor pulsa el botón Agregar miembro  2- Introduce los datos en los componentes.  3- Pulsa el botón guardar |
| Referencias | R13. |
| Precondiciones | El actor ha ingresado al sistema y se encuentra en la ventana del proyecto correspondiente |
| Poscondiciones | La información que contiene la base de datos relacionada al personal queda actualizada |
| Acción del Actor | Respuesta del Sistema |
|  |  |
| Otras acciones | - |
| Requerimientos especiales | - |

Tabla 8 Descripción de Caso de Uso Gestión de Personal

|  |  |
| --- | --- |
| Caso de Uso | Gestión de Tareas |
| Actor(es) | 1-Administrador  2-usuario activo |
| Propósito | Gestionar la información de las tareas existentes |
| Descripción | 1- El actor pulsa el botón Agregar tarea  2- Introduce los datos en los componentes.  3- Pulsa el botón guardar |
| Referencias | R12. |
| Precondiciones | El actor ha ingresado al sistema y se encuentra en la ventana de tareas |
| Poscondiciones | La información que contiene la base de datos relacionada a las tareas queda actualizada |
| Acción del Actor | Respuesta del Sistema |
|  |  |
| Otras acciones | - |
| Requerimientos especiales | - |

Tabla 9 Descripción de Caso de Uso Gestión de Tareas

|  |  |
| --- | --- |
| Caso de Uso | Gestión de Eventos Generales |
| Actor(es) | 1-Administrador  2-usuario activo |
| Propósito | Gestionar la información de los eventos existentes |
| Descripción | 1- El actor pulsa el botón Agregar evento  2- Introduce los datos en los componentes.  3- Pulsa el botón guardar |
| Referencias | R9. |
| Precondiciones | El actor ha ingresado al sistema y se encuentra en la ventana de eventos |
| Poscondiciones | La información que contiene la base de datos relacionada a los eventos queda actualizada |
| Acción del Actor | Respuesta del Sistema |
|  |  |
| Otras acciones | - |
| Requerimientos especiales | - |

Tabla 10 Descripción de Caso de Uso Gestión de Eventos Generales.

|  |  |
| --- | --- |
| Caso de Uso | Gestión de Proyectos |
| Actor(es) | 1-Administrador  2-usuario activo |
| Propósito | Gestionar la información de los proyectos existentes |
| Descripción | 1- El actor pulsa el botón Agregar proyecto  2- Introduce los datos en los componentes.  3- Pulsa el botón guardar |
| Referencias | R3. |
| Precondiciones | El actor ha ingresado al sistema y se encuentra en la ventana de proyectos |
| Poscondiciones | La información que contiene la base de datos relacionada a los proyectos queda actualizada |
| Acción del Actor | Respuesta del Sistema |
|  |  |
| Otras acciones | - |
| Requerimientos especiales | - |

Tabla 11 Descripción de Caso de Uso Gestión de Proyectos

|  |  |
| --- | --- |
| Caso de Uso | Gestión de Premios |
| Actor(es) | 1-Administrador  2-usuario activo |
| Propósito | Gestionar la información de los premios existentes |
| Descripción | 1- El actor pulsa el botón Agregar premio  2- Introduce los datos en los componentes.  3- Pulsa el botón guardar |
| Referencias | R11. |
| Precondiciones | El actor ha ingresado al sistema y se encuentra en la ventana de premios |
| Poscondiciones | La información que contiene la base de datos relacionada a los premios queda actualizada |
| Acción del Actor | Respuesta del Sistema |
|  |  |
| Otras acciones | - |
| Requerimientos especiales | - |

Tabla 12 Descripción de Caso de Uso Gestión de Premios

|  |  |
| --- | --- |
| Caso de Uso | Gestión de Carreras |
| Actor(es) | 1-Administrador  2-usuario activo |
| Propósito | Gestionar la información de las carreras existentes |
| Descripción | 1- El actor pulsa el botón Agregar carrera  2- Introduce los datos en los componentes.  3- Pulsa el botón guardar |
| Referencias | R4. |
| Precondiciones | El actor ha ingresado al sistema y se encuentra en la ventana de carreras |
| Poscondiciones | La información que contiene la base de datos relacionada a las carreras queda actualizada |
| Acción del Actor | Respuesta del Sistema |
|  |  |
| Otras acciones | - |
| Requerimientos especiales | - |

Tabla 13 Descripción de Caso de Uso Gestión de Carreras

|  |  |
| --- | --- |
| Caso de Uso | Gestión de Desglose |
| Actor(es) | 1-Administrador  2-usuario activo |
| Propósito | Gestionar la información de los proyectos existentes |
| Descripción | 1- El actor pulsa el botón de menú desglose o accede a la plataforma |
| Referencias | R6. |
| Precondiciones | El actor ha ingresado al sistema y se encuentra en la ventana de desglose o dashboard |
| Poscondiciones | La información que contiene la base de datos relacionada a los proyectos se procesa |
| Acción del Actor | Respuesta del Sistema |
|  |  |
| Otras acciones | - |
| Requerimientos especiales | - |

Tabla 14 Descripción de Caso de Uso Gestión de Desglose

## Anexo 2. Estimación de Esfuerzo, Tiempo de Desarrollo y Costo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Factor | Descripción | Peso | Valor | P\*V | Comentario |
| TCF01 |  | 2,00 | 2,00 | 4,00 | Es una aplicación  cliente-servido |
| TCF02 |  | 1,00 | 3.00 | 3.00 | Velocidad limitada por la  entrada de datos por el  operador |
| TCF03 |  | 1,00 | 1,00 | 1,00 | Escasas restricciones de  eficiencia |
| TCF04 |  | 1,00 | 0,00 | 0,00 | No se efectúan  operaciones complejas |
| TCF05 |  | 1,00 | 3.00 | 3.00 | Si se requiere que el  código sea reutilizable |
| TCF06 |  | 0.00 | 0.00 | 0.00 | El sistema debe ser fácil  de instalar para los  posteriores usuarios que  lo utilizarán |
| TCF07 |  | 0,50 | 5,00 | 2,50 | El sistema debe ser fácil  de usar para aquellos  usuarios que no poseen  un conocimiento amplio  en sistemas informáticos |
| TCF08 |  | 1,00 | 1,00 | 1,00 | Si se requiere que el  sistema sea portable |
| TCF09 |  | 1,00 | 2,00 | 2,00 | Fácil mantenimiento |
| TCF010 |  | 1,00 | 3,00 | 3,00 | Si existe concurrencia |
| TCF011 |  | 1,00 | 3,00 | 3,00 | Acceso restringido por  niveles de usuario |
| TCF012 |  | 1,00 | 0,00 | 0,00 | Usuarios con experiencia  en el uso de este tipo de  aplicaciones. Presenta  ayuda |
|  |  |  | Total: | 22.50 |  |

Tabla 15 Factores técnicos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Métrica | Descripción | Peso | Valor | P\*V | Comentario |
| ECF01 | Familiaridad con el modelo de  proyecto utilizado | 1,50 | 4,00 | 6,00 | Está bastante  familiarizado |
| ECF02 | Experiencia en la aplicación | 0,50 | 3,00 | 1,50 | Se ha trabajado  medianamente en  aplicaciones de este tipo |
| ECF03 | Experiencia en orientación a  objetos | 1,00 | 4,00 | 4,00 | Hay buena experiencia  en la orientación a  objeto |
| ECF04 | Capacidad del analista líder | 0,50 | 4,00 | 2,00 | El desarrollador tiene  una buena preparación |
| ECF05 | Motivación | 1,00 | 5,00 | 5,00 | El desarrollado está  motivado |
| ECF06 | Estabilidad de los requerimientos | 2,00 | 4,00 | 8,00 | Se esperan cambios |
| ECF07 | Personal a tiempo compartido | -1,00 | 3,00 | -3,00 | A tiempo parcial |
| ECF08 | Dificultad del lenguaje de  programación | -1,00 | 2,00 | -2,00 | El lenguaje de  programación utilizado  no es tan complejo  (Javascript) |
|  |  |  | Total: | 21,50 |  |

Tabla 16 Factor Ambiente