



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE
HIDALGO

INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA



ÁREA ACADÉMICA DE COMPUTACIÓN Y
ELECTRÓNICA

T E S I S

USO DE METODOLOGÍA WEB PARA EL DISEÑO DE UN SISTEMA DE
CONTROL

PARA TENER EL TÍTULO DE:

LICENCIADO EN CIENCIAS COMPUTACIONALES

P R E S E N T A

P.L.C.C. SANDRA OLVERA SOTO

DIRECTOR DE TESIS

M.C.C. GONZALO ALBERTO TORRES SAMPERIO

CODIRECTOR DE TESIS

M.I.D. ALBERTO SUÁREZ NAVARRETE

MINERAL DE LA REFORMA, HIDALGO, MÉXICO 2024





Universidad Autónoma del Estado de Hidalgo
 Instituto de Ciencias Básicas e Ingeniería
 School of Engineering and Basic Sciences

Mineral de la Reforma, Hgo., a 02 de mayo de 2024

Número de control: ICBI-D/531/2024
 Asunto: Autorización de impresión.

MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH

Con fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; Título Quinto, Capítulo II, Capítulo V, Artículo 51 Fracción IX del Estatuto General de nuestra Institución, por este medio le comunico que el Jurado asignado a la Pasante de la Licenciatura en Ciencias Computacionales **Sandra Olivera Soto**, quien presenta el trabajo de titulación "Uso de metodología Web para el diseño de un Sistema de Control", después de revisar el trabajo en reunión de Sinodales ha decidido autorizar la impresión del mismo, hechas las correcciones que fueron acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

Presidenta: Dra. Theira Irasema Samperio Monroy

Secretaria: MID. Ma. de Jesús Gutiérrez Sánchez

Vocal: M.C.C. Gonzalo Alberto Torres Samperio

Suplente: MID. Alberto Suarez Navarrete

Sin otro particular por el momento, reciba un cordial saludo.

Atentamente
 "Amor, Orden y Progreso"

Dr. Otilio Arturo Acevedo Sandoval
 Director del ICBI



QAAS/YDC



Ciudad del Conocimiento, Carretera Pachuca-Tulancingo Km. 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México, C.P. 42184
 Teléfono: 771 71 720 00 Ext. 2231 Fax 2109
 dirección_icbi@uaeh.edu.mx

uaeh.edu.mx



Agradecimientos

Agradezco sinceramente a todas las personas que contribuyeron de alguna manera en la realización de este trabajo de tesis. Gracias a Dios por permitirme llegar a este proceso, el cual, quiero expresar mi profunda gratitud a mis profesores por su orientación, apoyo y sabias sugerencias a lo largo de este proceso. Su guía fue fundamental para alcanzar los objetivos propuestos.

Agradezco de manera especial a mis padres, cuya confianza y apoyo incondicional fueron mi mayor fortaleza durante este camino. A mis queridos hermanos, quienes estuvieron a mi lado en los momentos de desafío, brindándome aliento y motivación para seguir adelante y alcanzar mis metas. Su presencia y respaldo fueron fundamentales en cada paso que di hacia la culminación de este proyecto. Gracias por ser mi sostén y mi inspiración.



Índice general

Resumen.....	8
Abstrac	9
Introducción.....	10
Problemática.....	11
Propuesta de solución	12
Justificación	13
Objetivos.....	14
Objetivo general	14
Objetivos específicos	14
Alcances	15
Limitaciones.....	15
Capítulo I Fundamentos teóricos y Antecedentes de la Investigación.....	16
1.1 Marco teórico.....	16
1.1.1 Historia de los sistemas web.....	16
1.1.2 Sistemas de Información.....	20
1.1.3 La expectativa de la tecnología.....	22
1.2 Marco conceptual	23
1.2.1 Sistemas web	23
1.2.2 Sistemas de control	24
1.2.3 Bases de Datos	24
1.2.4 Estado del arte.....	28
1.2.5 Internet como instrumento de trabajo.....	30
Capítulo II: Marco referencial	32
2.1 Metodologías.....	32
2.1.1 Metodología de desarrollo de aplicaciones <i>web</i>	32
2.1.2 Metodología estructurada	36
2.1.3 Metodología orientada al objeto	40
2.1.4 Metodología ágil.....	43
2.2 Comparación de metodologías.....	45
2.3 Elección de metodología.....	46



Capítulo III: Análisis	49
3.1 Análisis del Sistema	49
3.1.1 Requisitos de funcionalidad.....	49
3.1.2 Requisitos no funcionales.....	50
3.1.3 Perspectiva del sistema.....	51
3.1.4 Funcionalidad del sistema	51
3.1.5 Especificación de requerimientos	52
3.1.6 Proceso de ingeniería de requerimientos	53
Capítulo IV: Diseño.....	55
4.1 Base de datos	55
4.1.1 Modelo Entidad-Relación (E-R).....	56
4.1.2 Modelo Relacional	56
4.2 Diagramas UML	60
4.1 Diagrama de Caso de Uso.....	61
4.2.2 Diagrama Navegacional.....	63
4.2.3 Diagrama de Actividad.....	65
4.2.4 Diagrama de Clases	67
4.3 Prototipo de Interfaz del sistema de control de Innovación Educativa para el Alumnado	71
Conclusiones	74
Trabajos futuros.....	75
Glosario	76
Referencias	78



Índice de tablas

Tabla 1 Análisis de las versiones web 1.0, web 2.0 y web 3.0 Fuente: Elaboración propia.	19
Tabla 2 Análisis de las características de los SI Fuente: (Gilbson y Nolan 1974).....	21
Tabla 3 Análisis de tipos de bases de datos Fuente: (Equipo editorial, Etecé).....	26
Tabla 4 Análisis de comparación de metodologías Fuente: Elaboración propia.	45
Tabla 5 Análisis de comparación de metodologías de desarrollo para aplicaciones web Fuente: Elaboración propia.	46
Tabla 6 Análisis de la problemática en el sistema y necesidad Fuente: Elaboración propia.	49
Tabla 7 Análisis de requisito funcional registro de usuario Fuente: Elaboración propia....	52
Tabla 8 Análisis de requisito funcional autenticación de usuario Fuente: Elaboración propia.	52
Tabla 9 Análisis de requisito funcional Registro de actividades Fuente: Elaboración propia.	52
Tabla 10 Análisis de requisito funcional generación de reportes Fuente: Elaboración propia.	53
Tabla 11 Análisis de requisito funcional roles de seguridad Fuente: Elaboración propia...	53
Tabla 12 Roles de seguridad Elaboración propia.....	54
Tabla 13 Análisis para la tabla usuarios Fuente: Elaboración propia.	57
Tabla 14 Análisis de la tabla reportes Fuente: Elaboración propia.	59
Tabla 15 Análisis de la tabla de tipo reporte Fuente: Elaboración propia.	59
Tabla 16 Análisis de la tabla materia reportes Fuente: Elaboración propia.	60
Tabla 17 Análisis de la tabla de semestre Fuente: Elaboración propia.	60



Índice de Ilustraciones

Ilustración 1 Logo de Things Fuente: (THINGS s.f).....	30
Ilustración 2 Logo de UML Based Web Fuente: (Daniel, 2015).....	35
Ilustración 3 Presentación de Modelos UWE Fuente: (UML).....	35
Ilustración 4 Fases de SOHDM Fuente: (Ríos).	36
Ilustración 5 Fases de Modelo de Cascada	39
Ilustración 6 Modelo RAD	43
Ilustración 7 Metodología ágil Fuente: (Pressman 2005).	44
Ilustración 9 Modelo Entidad-Relación E-R Fuente: Elaboración propia.....	56
Ilustración 10 Modelo Relacional Fuente: Elaboración propia.	57
Ilustración 11 Diagrama de Caso de Uso Fuente: Elaboración propia.	63
Ilustración 12 Diagrama navegacional Fuente: Elaboración propia.....	65
Ilustración 13 Diagrama de Actividades Fuente: Elaboración propia.	67
Ilustración 14 Diagrama de clase Fuente: Elaboración propia.....	70
Ilustración 15 Prototipo de Interfaz Inicial Fuente Elaboración propia.	71
Ilustración 16 Prototipo de Interfaz de registro Fuente: Elaboración propia.....	72
Ilustración 17 Prototipo de Interfaz Inicio de sesión Fuente: Elaboración propia.	72
Ilustración 18 Prototipo de Interfaz de reportes Fuente: Elaboración propia.	73
Ilustración 19 Prototipo de Interfaz modificación de datos usuario Fuente: Elaboración propia.	73



Resumen

El mundo avanza hacia una era digital, que trasciende las expectativas, gracias al poder y la creatividad de los seres humanos. Estar actualizados en el mundo de la tecnología, conlleva un beneficio personal como profesional, el cual se obtiene competitividad al ofrecer soluciones innovadoras.

El proyecto nace de la necesidad de un sistema de control. Se busca obtener un modelo mediante la etapa de análisis y diseño, a través de un estudio con el apoyo de diversas metodologías, para lograr una mayor claridad y comprensión, que permita adaptarse de manera efectiva al sistema.

Se anticipa que el documento generado, desempeñe un papel fundamental en el proceso de desarrollo, al servir como una guía precisa, que refleje fielmente el análisis realizado. Este trabajo, no solo proporciona una visión clara de los requisitos y objetivos del proyecto, sino que también sirva, como una referencia útil para el equipo de desarrollo, a lo largo de todas las etapas del proceso. Su elaboración y su enfoque detallado deberían garantizar que el equipo tenga una comprensión completa de las necesidades del cliente, y que pueda traducirlas de manera efectiva, en soluciones de *software* funcionales y satisfactorias.

En última instancia, se confía en que este documento contribuirá significativamente a la eficiencia y éxito del proyecto de desarrollo.

Palabras Clave

Metodología, Sistema de Control, Tecnologías *Web*, Análisis, Diseño.



Abstrac

The world is moving towards a digital age, which transcends expectations, thanks to the power and creativity of human beings. Being updated in the world of technology brings personal and professional benefit, having the competitiveness to offer innovative solutions.

Being updated in the world of technology brings personal and professional benefit, having the competitiveness to offer innovative solutions.

The project arises from the need for a control system. The aim is to obtain a model through the analysis and design stage, through a study and analysis of various methodologies, to achieve greater clarity and understanding, which allows it to be effectively adapted to the system.

It is anticipated that the generated document will play a fundamental role in the development process, by serving as a precise guide that faithfully reflects the analysis carried out. This work not only provides a clear view of the project requirements and objectives, but also serves as a useful reference for the development team throughout all stages of the process. Its elaboration and detailed approach should ensure that the team has a complete understanding of the customer's needs, and can translate them effectively, into functional and satisfactory software solutions.

Ultimately, it is trusted that this document will contribute significantly to the efficiency and success of the development project.

Key words

Methodology, Control System, Web Technologies, Analysis, Design.



Introducción

En las últimas décadas, el desarrollo de sistemas ha experimentado una evolución notable, impulsada por avances tecnológicos, cambios en las metodologías de trabajo y una creciente demanda de soluciones informáticas, más eficientes y adaptables.

La gestión eficiente de la información y la generación de sistemas de control, son aspectos fundamentales en las actividades del día a día en la sociedad. Con la integración de tecnología *web* y la creciente digitalización de los procesos en diversos campos.

Este trabajo contiene análisis y diseño del sistema, partiendo desde los requerimientos que propone una idea conceptual, con base a una metodología *web*, para abordar las necesidades de un sistema de control, que se realizan procesos manuales y registros en diferentes formatos digitales.

En el primer capítulo, se presenta los fundamentos, las cuales se desarrollará el análisis y diseño del proyecto. Además, se exploran los antecedentes que han guiado esta investigación, para brindar así un marco de referencia fundamental, y comprender el estudio en su totalidad.

En el segundo capítulo, se exponen diversas metodologías para el análisis del sistema, las cuales son comparadas a través de cuadros comparativos, para permitir así, una evaluación detallada de cada una. El objetivo principal, es identificar y seleccionar la metodología más adecuada, según las necesidades y características particulares del sistema. Tras un análisis detallado, se optó por la metodología "*Based Web*", la cual se explica detalladamente junto con su funcionamiento.

En el tercer capítulo, se compone de la fase de análisis, que permite entender los requerimientos del sistema haciendo una estructura lógica para el diseño.

En el cuarto capítulo, ofrece una descripción del diseño de un proceso que abarca la planificación de la estructura del sistema y la organización de la información, estableciendo una base sólida, para la fase de desarrollo.



Problemática

A pesar del avance tecnológico y la creciente digitalización en la coordinación de la Licenciatura de Ciencias Computacionales, persiste el uso de métodos tradicionales, lo que limita la capacidad de responder de manera ágil y efectiva en los reportes de Actividades de Innovación Educativa.

La diversidad de formatos en los registros manuales y digitales genera riesgos de errores humanos, pérdida de reportes, ineficiencia en el uso de recursos y retrasos en la entrega. La falta de innovación en el sistema de control puede afectar la calidad y fiabilidad de la información generada, lo que a su vez puede comprometer la toma de decisiones informada en los usuarios finales.



Propuesta de solución

Ante la problemática, se propone una solución innovadora en el análisis y diseño del sistema de control. Esta solución busca mejorar la efectividad, mayor control y gestión de actividades, evitar errores, retrasos y optimizar el uso de los recursos disponibles.

El desarrollo de este sistema es esencial para satisfacer las necesidades y demandas específicas que se presentan. Para lograrlo, es imprescindible contar con una comprensión completa de los requisitos y condiciones que el sistema debe cumplir. Esto implica identificar y analizar detalladamente todas las características, funcionalidades y restricciones que deben incorporarse en el diseño y la implementación del sistema.

Solo al tener claridad sobre estos requerimientos, se podrá desarrollar un sistema que sea efectivo, eficiente y capaz de cumplir con las expectativas y necesidades de los usuarios finales.

En este contexto, esta tesis pretende abordar la problemática planteada, y proponer un sistema de control conceptual basado en una metodología específica, que se adapte a las necesidades particulares identificadas. Para lograr que se aplique a un enfoque multidisciplinario que combinará conceptos de ingeniería de *software*, diseño *web* y gestión de proyectos.

Porque a la creación de un *software*, siempre presenta desafíos y complejidades, lo que aumenta el riesgo de no cumplir con las expectativas de los clientes y usuarios finales, si el equipo de desarrollo no sigue una metodología desde el inicio. Por lo tanto, la selección adecuada de la metodología es fundamental, para el éxito del proyecto.



Justificación

La tecnología ha transformado profundamente la manera en que se concibe, diseña y opera los sistemas de control. Estos sistemas desempeñan un papel vital en la supervisión y regulación de procesos, máquinas y dispositivos para asegurar un funcionamiento eficiente, seguro y óptimo.

Con el análisis y diseño conceptual del sistema de control basado en la metodología *web*, se busca la optimización de los procesos y control de datos, ya que es importante y esencial para evitar errores, pérdidas de información.

Tener un mayor control y funcionamiento en la generación de información, dará mejor una comunicación clara y amigable con usuarios finales.

De igual manera, para atender a las necesidades anteriores, se pretende proporcionar un sistema funcional de calidad para el registro de información. Así beneficiar, el poder al incrementar eficiencia en los procesos existentes es fundamental para llevar a la excelencia mediante sistematización de servicios.

Esta tesis, no solo busca mejorar los procesos de gestión de información, sino también fomentar la innovación y el progreso mediante el aprovechamiento efectivo, de las tecnologías basadas en la *web*.



Objetivos

Objetivo general

Elaborar el análisis y diseño de un sistema de control, para la coordinación de la Licenciatura de Ciencias Computacionales, con el propósito de facilitar una representación conceptual clara y efectiva. Este proceso seguirá un enfoque ágil e iterativo, lo que permitirá una gestión eficaz y controlada del proyecto.

Objetivos específicos

- Analizar los requerimientos funcionales y no funcionales del sistema de control, para garantizar un buen desarrollo que cumpla con las necesidades y expectativas del usuario, mediante una investigación profunda.
- Implementar la metodología “*Based Web*”, mediante el desarrollo de sus fases para una construcción eficaz del sistema.
- Desarrollar un marco conceptual que integre los principios de control de procesos a las necesidades del sistema.



Alcances

Este trabajo se centra en dos fases esenciales: el análisis y el diseño conceptual del sistema de control. Ambas fases son interdependientes y se complementan para garantizar un desarrollo integral y efectivo del sistema de control de Actividades de Innovación Educativa.

Limitaciones

- Algunos docentes de la Licenciatura en Ciencias Computacionales desconocían de qué se les estaba hablando, al abordar las necesidades del sistema de las Actividades de Innovación Educativa, debido a la falta de información y comunicación entre ellos.
- Al indagar sobre la información citada, se encontró con la restricción de información por políticas de la organización, ya que el acceso estaba restringido a personas autorizadas con credencial actual de la organización.



Capítulo I

Fundamentos teóricos y Antecedentes de la Investigación

1.1 Marco teórico

Se pretende proporcionar una base sólida, para comprender, cómo las metodologías *web*, puede ser aplicada de manera efectiva en el diseño, desarrollo y funcionamiento de sistemas de control, y cómo puede contribuir a mejorar la eficiencia, la flexibilidad y la adaptabilidad de estos sistemas en diversos entornos y aplicaciones.

1.1.1 Historia de los sistemas web

En 1962 al aumento del comunismo, las fuerzas áreas de Estados Unidos solicitaron a un pequeño equipo de Investigadores que desarrollará una red de comunicaciones militares capaz de mantenerse operativa en caso de un ataque nuclear. La principal idea de garantizar la continuidad de las comunicaciones incluso si se dañaran o destruyera algunos de sus equipos.

Paul Baran es ampliamente reconocido como una de las figuras fundamentales en la concepción de internet; en 1964, concibió la idea de establecer una red con una estructura similar a la de una gran telaraña, se basaba en la vulnerabilidad de los sistemas centralizados, ideó un enfoque híbrido que, combinada con las topologías de estrella y maya, permitiendo que los datos se movieran de manera dinámica buscando la ruta más eficiente (Potter, 1999).

Arpanet (Advance Reserarch Projects Agency Network) red de agencia de proyectos de investigación avanzada fue establecida en 1969 como una red informática robusta destinada a transmitir datos militares y conectar a los principales grupos de investigación a lo largo de Estado Unidos. Utilizó en NCP (*Network Control Protocol* o Protocolo de Control de Red) y, posteriormente, la primera versión de protocolo de Internet, conocida como TCP/IP, desempeñando un papel



destacado en el desarrollo inicial de internet. Actualmente, ARPANET es considerada la precursora de internet (Potter, 1999).

La *World Wide Web*, nace en 1989 por Tim Berners-Lee, un investigador del CERN en Ginebra, diseñó un sistema de intercambio de información en la Red con posibilidades multimedia, crearon un protocolo de transferencia de hipertexto (HTTP), que estandarizó la comunicación entre cliente/servidor, URL un identificador, una especie de dirección *web* de un recurso de internet y HTML un lenguaje marcado de hipertexto. Su lanzamiento estuvo disponible para su lanzamiento en enero de 1992 (World Wide Web foundation, 2008).

Las primeras aplicaciones *web* iniciales consistían en páginas web estáticas simples que proporcionaban información elemental. Con la evolución de la *web*, aparecieron aplicaciones *web* dinámicas gracias a la introducción de lenguajes de programación como PHP y *JavaScript*. Esto posibilitó la creación de formularios interactivos y contenido que se actualizaba de manera instantánea.

En la *web* 1.0 sin duda alguna fue el periodo más largo de internet en la red y representa los orígenes de “algo” que al principio los creadores habían podido imaginar su repercusión.

1. Páginas estáticas
2. Extensiones propias de HTML
3. Libros de vistas
4. Botones “GIF”
5. Formularios HTML vía email.
6. Sin participación del usuario y visitante.
7. Páginas fijas difíciles de actualizar.



Con la aparición de la *web 2.0*, las aplicaciones *web* evolucionaron hacia plataformas interactivas y orientadas a la interacción social, los usuarios tenían la capacidad de compartir imágenes, mensajes y mantener comunicación con amigos y familiares en distintas partes del mundo como:

- Redes Sociales
- Servicios *web*
- Aplicaciones *web*
- *Blogs*

Esto conlleva un cambio radical a la hora de programación y tecnologías empleadas que se pasaba del HTML a técnicas como CSS, AJAX, JAVA y XML.

La *web 3.0* representa la evolución en la forma que interactuamos en línea y se manifiesta a través de diversas vías; estas incluyen la transformación de la *web* en una gigantesca base de datos, la expansión de la accesibilidad de contenido a través de aplicaciones no vinculadas a navegadores, el impulso de las tecnologías de inteligencias artificial, el desarrollo de la *web* semántica la *web* geoespacial, la *web 3D*, internet de las cosas (Telefónica S.A., s.f.).

La primera aparición registrada de “3.0 data en 2006 en un artículo de Jeffrey Zeldman, quién había criticado la *web 2.0*.

La *web* semántica, o *web 3.0* se enfoca en convertir la *web* en una “Web de datos”, incorporando metadatos semánticos y ontológicos a la World Wide Web. La *web 3.0* busca la conectividad universal, independientemente del lugar, dispositivo o momento en que los usuarios deseen acceder a la información (Torresburiel , 2022).



A través de este cuadro comparativo de la tabla (1), se muestra las características distintivas de cada versión de la web, destacando su impacto en la tecnología, la forma en que se relaciona con el mundo digital.

Tabla 1 Análisis de las versiones web 1.0, web 2.0 y web 3.0 Fuente: Elaboración propia.

Característica	Web 1.0	Web 2.0	Web 3.0
Tecnología	Estática. Páginas web simples con HTML estático.	Dinámica. Introducción a tecnologías como AJAX, CSS, JavaScript, para mayor interactividad.	Avanzada. Utilización de tecnologías IA, Web 3D.
Interactividad	Baja. Los usuarios son consumidores pasivos de contenido.	Media. Los usuarios pueden interactuar entre sí y con el contenido.	Alta. Los usuarios participan activamente en la creación y consumo del contenido.
Personalización	Mínima. El contenido es estático y no se adapta a los intereses individuales del usuario.	Moderada. Se introducen funciones de personalización como <i>feeds</i> personalizados y recomendaciones basadas al comportamiento del usuario.	Avanzada. Los sistemas de inteligencia artificial y aprendizaje automático se utilizan para adaptar el contenido y las experiencias web a cada usuario de manera única.
Ejemplos	Páginas estáticas de la década de 1990.	Redes sociales como <i>Facebook, Twitter</i> . Servicios web, <i>Blogs</i> .	Sistemas de recomendación de contenido, asistentes virtuales, aplicaciones de internet de las cosas.



1.1.2 Sistemas de Información

La evolución de los sistemas de información puede dividirse claramente en dos periodos históricos marcadas: El primero, que transcurre antes de la década de los años 60's, se caracteriza por la ausencia de acceso a las computadoras en las empresas, lo que resultaba en un trabajo manual y dificultades para convertir los datos en información confiable. El segundo periodo después de los años 70's, coincide con la incorporación de las computadoras al entorno empresarial. En esta etapa, las computadoras eran vistas como herramientas que simplificaban las actividades de la empresa, facilitando los procesos administrativos. Principalmente se utilizaban para llevar a cabo tareas contables y procesar documentos (Comas Rodríguez & Medina León, 2013).

Posteriormente, el avance en la informática y las telecomunicaciones posibilitó una mejora significativa en la eficiencia de las tareas, así como el ahorro de tiempo en su realización y la capacidad de almacenar una gran cantidad de información en espacios reducidos. Este aumento en la capacidad de la tecnología despertó un mayor interés en los sistemas de Información por parte de las organizaciones. Estos sistemas se convirtieron en una fuente de ventaja competitiva por derecho propio, ofreciendo una herramienta poderosa para diferenciarse de la competencia y obtener mejores resultados. De esta manera, los sistemas de información son considerados como elementos estratégicos dentro de las empresas (Trasobares, 2003).

Durante los últimos treinta años, a medida que la tecnología ha avanzado, los sistemas de información también han evolucionado. Inicialmente concebidos como sistemas aislados para registrar las operaciones diarias, en la actualidad han alcanzado un nivel donde, gracias al uso de la inteligencia artificial y herramientas sofisticadas, son capaces de diagnosticar el estado actual de las empresas de una forma autónoma y respaldar el proceso de toma de decisiones.



Gibson y Nolan (1974) ellos describieron en la evolución de los sistemas de información basándose en la evolución de las tecnologías de Información como se muestra en la tabla (2). Conforme a los equipos informáticos, software, hardware, bases de datos y telecomunicaciones fueron evolucionando, los sistemas de información comenzaron a adquirir una importancia creciente en las organizaciones (Trasobares, 2003).

Tabla 2 Análisis de las características de los SI Fuente: (Gibson y Nolan 1974).

Etapas de la evolución de los sistemas de información	Características
1. Iniciación	Introducción de las computadoras en las empresas. Aplicaciones informáticas orientadas a la mecanización y automatización de los procesos ordinarios. Escaso gasto en informática y escasa formación personal.
2. Contagio	La aplicación de las tecnologías de información origina resultados espectaculares. Difusión de las tecnologías de información en todas las áreas de la empresa. Aumenta la cualificación del personal. Existe una gran descoordinación y poca planificación en el desarrollo de los sistemas de información.
3. Control	La alta dirección de la organización se preocupa de los sistemas de información como consecuencia de alto coste en ellos. Centralización de los proyectos de inversión en tecnologías de información.
4. Integración	Se controla el incremento del gasto. Se produce la integración de los sistemas de información existentes en las distintas áreas de la empresa. Mejora y perfeccionan los sistemas de información.
5. Administración de la información	El sistema de información adquiere una dimensión estratégica en la empresa. Descentralización de ciertas aplicaciones informáticas.
6. Madurez	Desarrollo de los sistemas información en los niveles superiores de la organización apareciendo los sistemas estratégicos de información. Adquiere gran importancia la creatividad y la innovación.



1.1.3 La expectativa de la tecnología

La capacidad de transformación y mejora de las nuevas tecnologías siguen evolucionando para cumplir con las crecientes demandas de los usuarios y las empresas en términos de funcionalidad, rendimiento, seguridad y accesibilidad. Con el avance de las tecnologías para el desarrollo de páginas web y evolución constante de lenguajes y herramientas como HTML5, CSS3 y *Javascript*, están marcando el ritmo de la revolución en la forma que interactuamos con la web y sus aplicaciones con una experiencia digital más fluida, interactiva y eficiente.

Una de las principales expectativas tecnológicas es la aceleración de los procesos y el manejo de mayor cantidad de datos. Esto se debe a varios factores:

- **Avances en Hardware:** La continua mejora en la capacidad de procesamiento de los microprocesadores y almacenamiento.
- **Tecnologías de Nube:** Escalable y flexible, procesar grandes volúmenes de datos.
- **Automatización de Procesos:** Utilizando tecnologías como la robótica y la automatización de flujos de trabajo más rápido, liberando tiempo y recursos para otras actividades.



1.2 Marco conceptual

Esta área consiste en una serie de conceptos que se mencionan este documento para tener una claridad y profundidad sobre el tema mencionado, exploraremos los fundamentos y aplicaciones de los sistemas de control, destacando su importancia. Se combinan estas dos áreas, conocido como sistemas *web* y sistemas de control que se centra en el diseño y desarrollo de aplicaciones web permitiendo controlar y monitorear sistemas físicos de manera remota y eficiente. Estos sistemas integran tecnologías web con técnicas de control y automatización para ofrecer soluciones innovadoras en campos como la industria, la energía, la salud, la educación, entre otros.

Al fusionar los principios de los sistemas web y los sistemas de control, se abre un mundo de posibilidades en términos de accesibilidad, escalabilidad y flexibilidad.

1.2.1 Sistemas web

Sistemas web o también denominado aplicaciones web, son aquellos que están creados e instalados no sobre una plataforma o sistemas operativos, si no que se alojan a un servidor en internet o en una intranet (red local), sin necesidad de tener una aplicación en el escritorio.

El sistema web se puede utilizar en cualquier navegador web (Chrome, Firefox, internet Explorer etc...) sin importar el sistema operativo.

Las aplicaciones web trabajan con base de datos que permiten procesar, almacenar y mostrar información de forma dinámica para el usuario (Ramos Martín, 2015).

En este sentido *Barners* lo define como “Un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de internet”. Por ejemplo, la *World Wide Web* (WWW).

Una de las principales características va a ser su alto grado de interacción con el usuario, y el diseño de su interfaz claro y simple.



1.2.2 Sistemas de control

En la evolución de los sistemas de control ha sido un proceso constante impulsado por avances tecnológicos y cambios en las necesidades industriales y sociales que ha llevado a una mayor automatización, precisión y eficiencia.

Todos los sistemas tenían un funcionamiento manual. Los sistemas aparecen como una forma de captar información (Mario Alberto, 2007).

Los sistemas de control consisten en una serie de componentes conectados entre sí y diseñados para alcanzar un objetivo determinado. En otras palabras, este tipo de sistemas lleva a cabo acciones en dispositivos con el fin de supervisar, orientar o influir en el comportamiento de un sistema dentro de un entorno específico (SDI, 2022).

Desde tiempos antiguos, se registran las primeras instancias de adaptaciones basadas en sistemas de control. Un ejemplo proviene de la antigua Grecia, es el reloj de Ktesibios que según registros data del año 250 antes de cristo.

Actualmente los sistemas de control juegan un papel fundamental en la mejora de los procesos, dado que cuentan con capacidades para abordar y resolver información de manera automatizada, gracias a los avances tecnológicos se ha logrado tener la posibilidad de visualizar gran cantidad de información en un monitor electrónico, donde a veces no es necesario el factor humano. Estos sistemas son utilizados en diferentes sectores: Industriales, económicos, sociales, laborales, educativos, entre otros (Campos, 2021).

1.2.3 Bases de Datos

Es un conjunto organizado de información estructurada que se almacena electrónicamente en un sistema informático. Esta información puede estar relacionada entre sí y se utiliza para diversos fines, como almacenar registros de clientes, transacciones financieras, inventarios, registros médicos, entre otros. Las bases de datos están diseñadas para permitir la gestión eficiente, la recuperación rápida y la manipulación segura de grandes volúmenes de datos (Oracle, 2024).



Existen diversas formas de clasificar las bases de datos, considerando aspectos específicos (Equipo editorial, Etecé, 2023).

Tipos de base de datos

Bases de datos relacionales

Las bases de datos relacionales son las más comunes, siendo la tecnología preferida. Estas bases de datos están diseñadas para almacenar datos organizados en tablas que están relacionadas entre sí. La disposición de los datos se realiza en filas y columnas, empleando claves o llaves para vincular la información entre distintas tablas. Estas son funcionales para aplicaciones que demandan transacciones en tiempo real, como sistemas de gestión de inventario, ventas o finanzas (arsys, 2023).

Base de datos distribuidas

Las bases de datos distribuidas son aquellas en lo que los datos están almacenados y gestionados de manera distribuida en múltiples ubicaciones físicas o lógicas. En lugar de ser centralizada, los datos se distribuyen en diversos servidores, a menudo en redes distintas, que mantienen una porción de los datos.

Los clientes pueden acceder a los datos y recuperar la información sin necesidad de preocuparse de que ésta se encuentre distribuida entre varias máquinas (arsys, 2023).

Base de datos orientada a objetos

Las bases de datos orientadas a objetos son un modelo en el que la información se guarda utilizando objetos. Estos objetos contienen conjuntos diversos de datos, con la complejidad necesaria para representar la información requerida. Para quienes tienen experiencia en programación, estas bases de datos pueden entenderse



como sistemas que almacenan registros de manera similar a objetos en la Programación Orientada a Objetos (Digital guide IONOS, 2023).

Las bases de datos orientadas a objetos son particularmente convenientes cuando ya estamos utilizando lenguajes de programación orientados a objetos, como Java, ya que los objetos del código fuente se pueden incorporar sin dificultad a la base de datos.

Hay diferentes tipos de bases, las cuales son estáticas y las dinámicas. La principal diferencia entre ambas es que, mientras las bases de datos estáticas son de solo lectura y no permite modificar o añadir datos, las bases de datos dinámicas son mucho más flexibles y sí permiten editar, actualizar o borrar datos como se muestra en la tabla (3).

Tabla 3 Análisis de tipos de bases de datos Fuente: (Equipo editorial, Etecé)

Según su variabilidad	Según su contenido
Bases de datos estáticas: Utilizadas en inteligencia empresarial y análisis histórico, estas bases de datos son de solo lectura, lo que significa que se pueden consultar, pero no modificar.	Bibliográficas: Contienen material de lectura, como libros y revistas, organizado mediante datos clave como el autor, el editor, el año de publicación, el área temática o el título del libro, entre otros.
Bases de datos dinámicas: Además de permitir consultas, estas bases de datos gestionan procesos de actualización, reorganización, adición y eliminación de información.	De texto completo: Almacenan textos históricos o documentales que deben preservarse completamente y se consideran fuentes primarias.
	Directorios: Consisten en grandes listados de datos personalizados, como direcciones de correo electrónico o números de teléfono. Por ejemplo, las empresas de servicios suelen gestionar extensos directorios de clientes.
	Especializadas: Diseñadas para proporcionar información hiper especializada o técnica a un público específico que consume este tipo de datos según sus necesidades particulares.



Modelo de base de datos

Un modelo de base de datos ilustra la organización lógica de una base de datos, delineando las relaciones y normativas que rigen cómo los datos pueden ser almacenados, estructurados y gestionados. Cada aplicación de base de datos se apoya en un modelo de datos particular, los cuales son diseñados conforme a las reglas y conceptos de un modelo de datos más amplio adoptado por el sistema.

El manejo de las bases de datos se realiza a través de sistemas de gestión (conocidos como DBMS por sus siglas en inglés: *Database Management Systems* o Sistemas de Gestión de Bases de Datos), los cuales son actualmente digitales y automatizados. Estos sistemas facilitan el almacenamiento organizado y la rápida recuperación de la información, representando el fundamento de la informática moderna. Los sistemas gestores de bases de datos más populares y que se usan en la mayoría de los proyectos online son los siguientes:

MySQL

MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS) que opera bajo un modelo cliente-servidor. Es el sistema de gestión de bases de datos relacional más ampliamente utilizado en la actualidad debido a su naturaleza de código abierto (OCI, 2024).

SQL Server

MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS) que sigue un modelo cliente-servidor. Se trata de un servidor de bases de datos relacional diseñado para servir a otras aplicaciones de software, las cuales pueden funcionar en el mismo equipo o en otros equipos a través de una red, incluso en Internet (Hughes, 2021).



1.2.4 Estado del arte

Construcción de una base de datos para un sistema web en apoyo al proceso enseñanza-aprendizaje de la Licenciatura en Ciencias Computacionales.

Esta tesis presentada en la Universidad Autónoma del estado de Hidalgo, México se centra en un análisis sobre la falta de tiempo para la asistencia de asesoría o clase, así como el aspecto económico este sistema de base de datos es un ambiente *web* como apoyo a las funciones pedagógicas y cognoscitivas que realizan catedráticos y alumnos dentro de la Lic. Sistemas Computacionales del Instituto de Ciencias Básicas e Ingeniería, donde se integra información que sea útil para la elaboración de proyectos y evaluaciones correspondientes a las materias que forman parte del curso escolar, el cual se benefician a los alumnos con cuestionarios referentes a lo expuesto en clase, obteniendo una idea de cómo se encuentra en dicha materia llevando un control de los apuntes, proyectos y/o tareas manejados durante todo el semestre.

Para su funcionamiento de este proyecto se elaboró diagramas esquemas en el área de diseño. Así mismo, utilizando PHP y la base de datos se realizó en PHP *MyAdmin*, facilitando la elaboración de tablas y HTML el cual contiene texto, especificaciones de fuentes y otras instrucciones de formato.

En una de sus dificultades presentaron en la validación de los usuarios al utilizar la criptografía md5 de PHP. Desarrollado por *RSA Data Security*.

En conclusión, se probó la aplicación completa basándose en los criterios de aceptación satisfaciendo las necesidades de los usuarios finales (Lora, 2004).

Modelo de evaluación de metodologías de desarrollo de software de web

Esta tesis, presentada en la universidad de coruña 2021, se enfoca en un estudio sobre las metodologías ágiles e híbridas para el desarrollo de aplicaciones web, que



garantizan un proceso de desarrollo óptimo. Su objetivo principal de esta tesis es la investigación de un modelo de evaluación, para su valoración, eficiencia de proyectos basado en sus fases y características la metodología híbrida SWIRL a través de un sistema web. Utilizando modelo de evaluación llamado “MEASS” (Modelo de Evaluación de Aplicación Según SWIRL) que muestra los porcentajes de los resultados.

En conclusión, para el desarrollo de una aplicación web es necesario gestionar de manera correcta la selección de una metodología evaluando los datos necesarios del usuario final. Para su elaboración beneficiando la calidad del proyecto, satisfaciendo el cliente final, reducción de riesgos, optimización de procesos (Ríos, 2020).

Things

Es un gestor de tareas creado por *Cultured Code*, software en *Stuttgart*, Alemania, basada en la filosofía GTD (*Getting Things Done*), este sistema fue lanzado para *MAC* en Julio del 2007 cuando se lanzó la *app store*, admite las plataformas de *iOS*, *MACOs*, *WatchOS*, *iPadOS*.

Unas de sus características principales es su sencillez de uso y sincronización entre dispositivos compatibles, soporte para etiquetas *Tags* y poca curva de aprendizaje.

La aplicación ofrece a los administradores la posibilidad de enviar listas de tareas a una bandeja de entrada mediante una interfaz intuitiva de arrastrar y soltar. Además, permite a los usuarios etiquetar tareas, establecer fechas de inicio o plazos, y utilizar encabezados para organizar categorías o hitos. La función de búsqueda rápida integrada permite a los empleados buscar proyectos, listas de verificación o etiquetas en tiempo real.

Things también posibilita a los administradores sincronizar datos con *Apple Calendar* para visualizar eventos o listas de tareas pendientes provenientes de diversas aplicaciones de terceros, como *Google Calendar*, *Microsoft Exchange* y *Outlook*. Para una organización eficiente como se muestra en la ilustración (1), las partes interesadas pueden utilizar el módulo de áreas para crear secciones



dedicadas a distintos aspectos de la vida, como escuela, salud, familia o trabajo, y así almacenar proyectos relacionados y listas de tareas pendientes en un único repositorio centralizado.

En conclusión, este sistema es una aplicación de gestión de tareas de primer nivel teniendo equilibrio perfecto entre la facilidad de uso y principalmente por su diseño amigable (THINGS, s.f.).

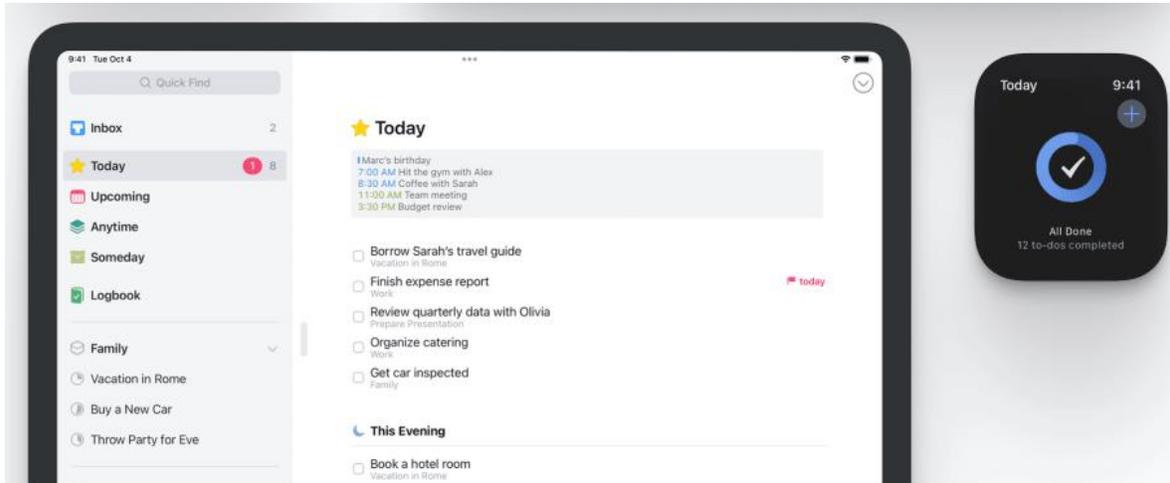


Ilustración 1 Logo de Things Fuente: (THINGS s.f).

1.2.5 Internet como instrumento de trabajo

A través de Internet, los equipos de trabajo pueden acceder a herramientas de desarrollo *web*, plataformas de gestión de proyectos en línea, sistemas de almacenamiento de datos en la nube y otras soluciones tecnológicas que respaldan el proceso de desarrollo y ejecución de proyectos *web*.

Además, Internet permite la comunicación y colaboración en tiempo real entre los miembros del equipo, incluso si están ubicados en diferentes lugares geográficos. Esto facilita la implementación de metodologías ágiles y la realización de reuniones, revisiones y actualizaciones de manera eficiente.

Un proceso que se suma a la generalización de internet como instrumento de trabajo, es que ha generado cambios notables en la construcción de redes de control de actividades, con evidencias de un gran aumento en el ámbito laboral.



Frente a la idea mantenida durante la década de 1990 de la poca fiabilidad en la comunicación mediada por internet, los trabajos de estos autores muestran que la incorporación de las tecnologías de internet ofrece la posibilidad de modelar múltiples instancias de la producción de datos, como el anonimato, la geografía o la sincronía de la comunicación, a través de las cuales desplegar las condiciones necesarias para acceder a determinados colectivos y fenómenos cuyos miembros resultarían difícilmente accesibles mediante técnicas de investigación convencionales.

La mayoría de las interacciones sociales que existen para la producción de datos son en contexto cara a cara y tiene una naturaleza efímera de modo que solo se obtienen los datos tomando notas, realizando grabaciones entre otras formas y de esa manera es posible preservar los registros de esa interacción, en cambio las interacciones sociales que son realizadas con ayuda de una herramienta tecnológica como es el internet producen registros materiales que son normalmente permanentes ya que no pueden sufrir fácilmente pérdida y son fácilmente accesibles a ellos. Aunque la extensión de esa permanencia es muy variable e impredecible, y puede ir de los minutos a los años, la materialidad de esas interacciones les dota de singulares propiedades, a esa condición de inmutabilidad y movilidad se añaden la permanencia y accesibilidad (Octavio Islas Carmona, 2000).



Capítulo II: Marco referencial

2.1 Metodologías

En la actualidad, se emplean diversas metodologías para el desarrollo de *software*, las cuales se seleccionan en función del tipo de sistema a desarrollar. Estas metodologías pueden agruparse en categorías comunes, tales como aplicaciones de escritorio, móvil y *web*. Estas metodologías permiten organizar, comunicar y comprender. En la actualidad, las metodologías han sido fundamentales para el desarrollo y diseño de plataformas, ya que guían el proceso de creación. Estas metodologías no solo aseguran la correcta implementación en diseño y programación, sino que también contribuyen a la optimización del rendimiento, la usabilidad y la accesibilidad de las plataformas. Al seguir metodologías adecuadas, se logra una experiencia de usuario mejorada y una mayor probabilidad de alcanzar los objetivos establecidos del sistema. Las siguientes metodologías son enfoques diferentes para el desarrollo de *software*, cada una con sus propias características y principios.

2.1.1 Metodología de desarrollo de aplicaciones *web*

El proceso de crear y diseñar aplicaciones *web* implica llevar a cabo las visiones, metas o requisitos de los clientes en línea, utilizando tecnologías apropiadas para cada proyecto en particular. Estas aplicaciones ofrecen diversos servicios a los usuarios de internet que las acceden a través de un navegador *web*.

El desarrollo de aplicaciones *web* se debe al impacto global del internet como plataforma de difusión de información y servicios. Con el avance tecnológico, el desarrollo de estas aplicaciones se ha vuelto más complejo, enfrentando con grandes cantidades de datos.

Cuando un método es más complejo en sus etapas, suele haber menos errores en términos de calidad y coherencia en los datos.

Las metodologías para el desarrollo de aplicaciones *web* pueden variar en la extensión de sus etapas según el enfoque dado. Estas incluyen:



Diseño Conceptual: En esta fase se trata de la especificación del dominio del problema, definiendo sus aspectos y relaciones.

Diseño Navegacional: Se enfoca en el acceso y la visualización de los datos.

Diseño de la Presentación o Interfaz: Se concentra en cómo la información será presentada a los usuarios. En esta etapa, el cliente desempeña un papel crucial al definir los requisitos, mientras que los usuarios expresan como desean interactuar con el sistema.

Implementación: Consiste en la construcción del *software* utilizando los artefactos generados en las fases anteriores (Molina Ríos, J.R., Zea Ordóñez, M.P., Contento Segarra, M.J. y García Zerda, F.G., 2018).

2.1.1 Metodología de Gestión de Relaciones RMM (*Relationship Management Methodology*)

La metodología fue concebida por *Isakowitz, Stohr y Balasubramanian*, y de acuerdo con ellos, es especialmente adecuada para dominios que presentan estructuras regulares, es decir, con clases de objetos bien definidas y relaciones claras entre esas clases. Por ejemplo, catálogos o interfaces de bases de datos convencionales. Esta metodología se enfoca en problemas con datos dinámicos que cambian con frecuencia, en lugar de entornos estáticos.

Se define como un proceso que abarca el análisis, diseño y desarrollo de aplicaciones hipermedia (Pérez Rodríguez, 2007).

El enfoque propone un sistema de comunicación que facilita la descripción de los elementos presentes en el ámbito específico, incluyendo sus conexiones y las formas en que la aplicación hipermedia permite la navegación entre ellos. Los elementos en este ámbito son delineados mediante la colaboración de entidades, atributos y relaciones asociativas, destacando las siguientes características principales:



- Enfoque para la creación de páginas *web*, adoptando un enfoque centrado en la información.
- Modelado de sitios web a un nivel lógico, que abarca el dominio de información, las estructuras de navegación y los elementos de presentación.
- Incorporado dentro de un marco de trabajo de desarrollo.
- Ayuda a organizar páginas *web* complejas que incluyen elementos de diversas entidades (múltiples vistas).
- Genera enlaces más eficaces y flexibles.
- Mantiene la coherencia durante la navegación.

2.1.1.2 Metodología UWE (UML-BASE WEB)

UWE es una metodología propuesta de Ingeniería *Web*, que mejora la especificación durante la creación de una aplicación *web*. Utiliza una notación estándar basada en UML (Lenguaje Unificado de Modelado) para sus modelos y procedimientos, lo que simplifica la transición. La metodología proporciona una guía clara para la construcción de cada elemento del modelo.

Definición de métodos: Definición de los pasos para la construcción de los diferentes modelos.

Especificación de Restricciones: Se recomienda el uso de restricciones escritas (OCL: Lenguaje de restricciones de objetos) para aumentar la exactitud de los modelos.

Proceso unificado para modelar aplicaciones web, tiene como objetivo cubrir el ciclo de vida, desarrollando procesos de modelado para generar aplicaciones *web* (Gabriel, Metodologías de desarrollo de software, 2015).





Ilustración 2 Logo de UML Based Web Fuente: (Daniel, 2015)

Esta metodología consta de fases fundamentales las cuales son el análisis de requisitos, el diseño conceptual y el diseño de la navegación, los cuales se describen cada fase con una presentación de los modelos en la ilustración (3).

Análisis de requisitos: En esta etapa se establecen los requisitos funcionales que necesita la aplicación *web* para representarlos en un modelo de casos de uso.

Diseño conceptual: Se construye el modelo conceptual del sistema considerando los requisitos hechos en el modelo de casos de uso. El resultado es el diagrama de clases de dominio.

Diseño navegacional: Se obtiene el modelo de espacio de navegación y el de estructura de navegación, que muestra como navegar a través del espacio. El resultado son diagramas de clases que representan estos modelos.

Diseño de presentación: De este paso se obtiene una serie de Interfaz de usuario que se presentan mediante diagramas de interacción UML (Fontela, 2011).

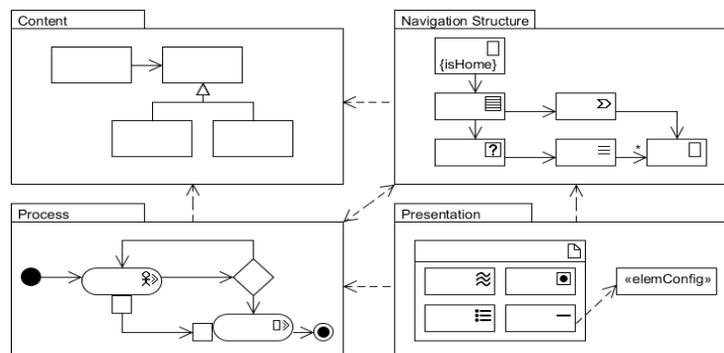


Ilustración 3 Presentación de Modelos UWE Fuente: (UML).



2.1.2 SOHDM

La Metodología de Diseño de Escenarios Orientado a Objetos en Hipermedia se centra en la creación de escenarios para el desarrollo del *software* y pone especial atención en el manejo de los requisitos. Las etapas de esta metodología tienen cierta correspondencia con las fases de *RMM*, *OOHDM* y *EROM*. Esta metodología consta de seis fases: Análisis, Modelado de Objetos, Diseño de Visitas, Diseño Navegacional, Diseño de la Implementación y Construcción. En la siguiente ilustración (4) se describe detalladamente cada fase (Ríos, Jimmy Molina, 2020).



Ilustración 4 Fases de SOHDM Fuente: (Ríos).

2.1.2 Metodología estructurada

Es un enfoque organizado para crear sistemas de información destinados a diversas organizaciones. Algunas de las características destacadas de esta metodología incluyen lo siguiente: Es organizada, lo cual se debe a dos razones fundamentales: Emplea diversos métodos y técnicas estructuradas que son propias de la Ingeniería de la Programación y que han demostrado ser las más eficientes y efectivas para el desarrollo de sistemas programados.



Tiene como objetivo emplear las metodologías de análisis y diseño estructurado para su uso con herramientas *CASE*, incrementando la productividad en el desarrollo e implementación de sistemas de información y entre ellas se puede encontrar a *Kendall & Kendall* (Agustín).

Esta metodología crea los modelos de manera descendente. Son las orientadas a procesos, a datos y las mixtas. Intentan aplicar formas ingenieriles para solucionar problemas técnicos al obtener un sistema de información, proponen la creación de modelos, flujos y estructuras mediante un *top-Down*. Es la primera aproximación al problema. Está orientada a procesos, es decir, se centra en especificar y descomponer la funcionalidad del sistema. Se utilizan varias herramientas:

- **Diagramas de flujo de datos (DFD):** Estos representan la forma en la que los datos se mueven y se transforman, incluyendo; procesos, flujos de datos y almacenes de datos. Los procesos individuales se pueden a su vez componer en otros DFD de nivel superior.
- **Especificaciones de procesos:** Es lo que se escribe para uno de los procesos definidos en el DFD cuando no se puede descomponer más. Puede hacerse en pseudocódigo, con tablas de decisión o en un lenguaje de programación.
- **Diccionario de datos:** Son los nombres de todos los tipos de datos y almacenes de datos junto con sus definiciones (Gabriela, 2009).
- **Diagramas de transición de estados:** Modelan el proceso que dependen del tiempo.
- **Diagramas entidad-relación:** Los elementos del modelo E/R se corresponden con almacenes de datos en el DFD. En este diagrama se muestran las relaciones entre dichos elementos (Gabriela, 2009).



2.1.2.1 Modelo de Cascada

La primera descripción formal de la metodología de cascada fue elaborada en el artículo de *Wintson. W Royce* en 1970 sobre la gestión de desarrollo de *software*. Sin embargo, se reconoce que el concepto de cascada se originó en los campos de fabricación y la construcción. Esto implica que no se permite regresar a una fase anterior del proceso (Stsepanets, 2024). Como se muestra en la ilustración (5) cada fase indica siempre hacia adelante con una flecha, nunca se retrocede.

Está basado en la mantenibilidad de línea de ensamblaje, que consta de 5 Fases:

- **Análisis y definición de requerimientos.** Mediante consultas con los usuarios se definen los servicios, restricciones y metas del sistema.
- **Diseño del software y del sistema.** Divide los requerimientos en sistemas de *software* y de abstracciones fundamentales del sistema de *software*.
- **Implementación y pruebas unitarias.** Durante esta etapa, el diseño de software se lleva a cabo como un conjunto de unidades, lo cual implica verificar que cada unidad cumpla con sus especificaciones.
- **Integración y pruebas del sistema.** Los programas o unidades individuales se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del *software*.
- **Funcionamiento y Mantenimiento.** Es la fase más larga del ciclo de vida. El sistema se instala y se pone en funcionamiento práctico, el mantenimiento implica corregir errores no descubiertos



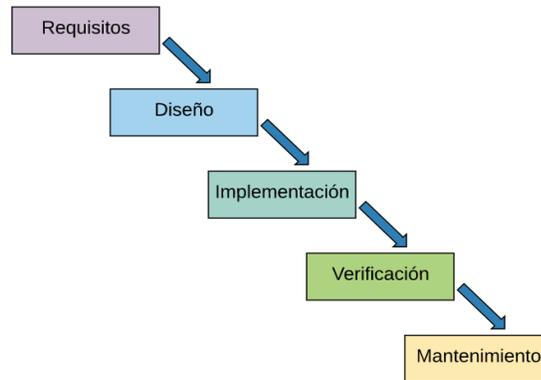


Ilustración 5 Fases de Modelo de Cascada

2.1.2.2 Modelo Espiral

Uno de los modelos más conocidos, especialmente dirigido a la reducción de riesgos, es el denominado modelo en espiral, del año 1986.

Barry W. Boehm presentó la idea del modelo de desarrollo en espiral en su ensayo titulado *"A Spiral Model of Software Development and Enhancement"*. En aquel momento, el modelo de desarrollo en cascada era el más predominante, lo que llevó a una amplia discusión sobre las limitaciones asociadas con dicho enfoque. A diferencia de otros modelos como *"code and fix"* o el modelo en cascada, el desarrollo en espiral se fundamenta en la gestión del riesgo.

En la ilustración (6), las fases no se llevan a cabo de manera lineal y secuencial, sino que se repiten varias veces en un ciclo espiral. A través de esta repetición iterativa, el proyecto avanza hacia su objetivo de manera gradual, lo que ayuda a reducir significativamente el riesgo de fracaso del proceso de desarrollo mediante controles regulares (IONOS, 2023).



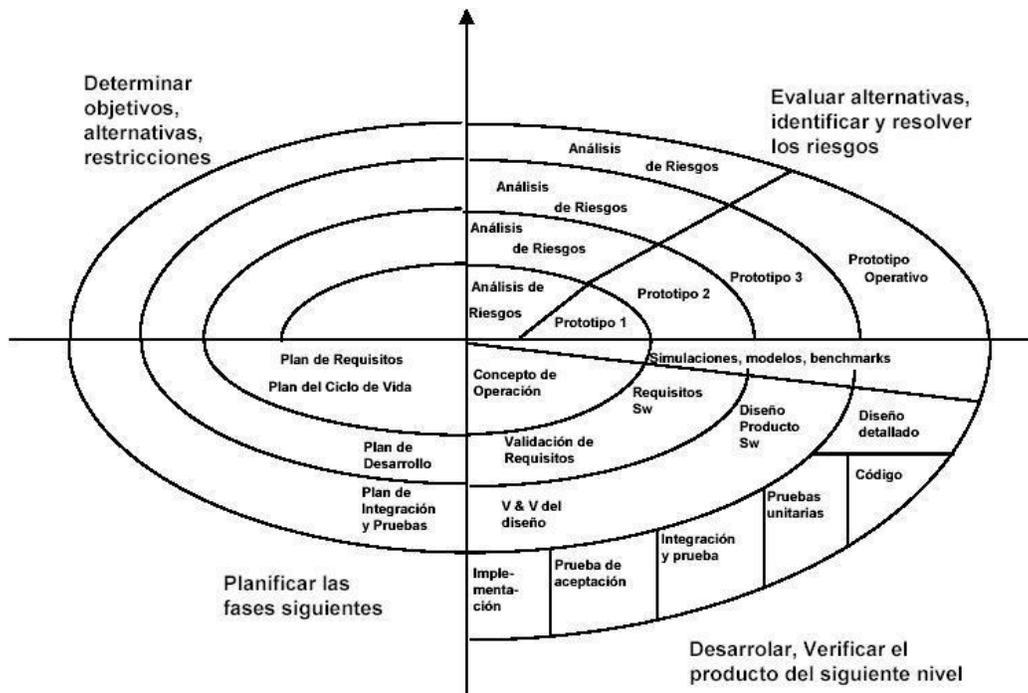


Ilustración 6 Etapas del Modelo espiral

2.1.3 Metodología orientada al objeto

El diseño orientado a objetos desarrolla una representación del campo del problema en el mundo real y lo hace corresponder con el ámbito de la solución, que es el *software*. El diseño orientado a objetos produce un diseño que interconecta objetos de datos y operaciones de una forma que empata la información y el procesamiento.

Cuando se utiliza el diseño orientado a objetos, el diseñador puede desarrollar sus propios tipos abstractos de datos y abstracciones finales y hacer corresponder el campo del mundo real con dichas abstracciones creadas por el mismo programador. Esta correspondencia se torna la mayoría de las veces más natural, ya que el rango de tipos abstractos de datos que puede inventar el diseñador es virtualmente ilimitado. Más aún, el diseño del *software*, sistemas o aplicaciones se desliga de los detalles de representación, sin que ello afecte al sistema global.



La naturaleza única del diseño orientado a objetos queda reflejada en su capacidad de construir sobre tres pilares conceptuales importantes del diseño de *software*, sistemas y/o aplicaciones:

- Abstracción
- Ocultamiento de información
- Modularidad

Abstracción

Se enfoca en la visión externa de un objeto, separa el comportamiento específico de un objeto, a esta división que realiza se le conoce como la barrera de abstracción, la cual se consigue aplicando el principio de mínimo compromiso.

Ocultamiento de información

Es la ocultación de decisiones de diseño en un programa susceptible de cambios con la idea de proteger a otras partes del código si estos se producen. Proteger una decisión de diseño supone proporcionar una interfaz estable que proteja al resto del programa de la implementación el cuál es susceptible a cambios. En los lenguajes de programación modernos el principio de ocultación de información se manifiesta de diferentes maneras, como, por ejemplo, la encapsulación.

Modularidad

Esta consiste en dividir un programa en módulos que puedan compilarse por separado, sin embargo, se tiene conexiones con otros módulos. Esta etapa se compone de tres módulos:

- **Capacidad de descomponer un sistema complejo**
Recuerdas el principio de “Divide y Vencerás”, en este procedimiento se realiza algo similar, ya que descomponen un sistema en subprogramas (recuerda llamarlos módulos), el problema en general lo divides en problemas más pequeños.



- **Capacidad de componer a través de sus módulos**

Indica la posibilidad de componer el programa desde los problemas más pequeños completando y resolviendo el problema en general, particularmente cuando se crea *software* se utilizan algunos módulos existentes para poder formar lo que nos solicitan, estos módulos que se integran a la aplicación deben de ser diseñados para ser reusables.

- **Comprensión de sistema en partes**

El poder tener cada parte separada nos ayuda a la comprensión del código y del sistema, también a la modificación de este, recordemos que si el sistema necesita modificaciones y no hemos trabajado con módulos definitivamente eso será un caos (Román, 2013).

2.1.3.1 Desarrollo Rápido de Aplicaciones (RAD)

Históricamente, el Desarrollo Rápido de Aplicaciones (RAD) ha sido reconocido como un modelo de desarrollo de software en el cual los distintos módulos de la aplicación se desarrollan simultáneamente y luego se integran para formar un producto finalizado.

La metodología involucra el desarrollo interactivo, la creación de prototipos y la utilización de herramientas *CASE* (Ingeniería de *Software* Asistida por Computadora). Las fases dentro de la metodología RAD son las siguientes:

Planificación de necesidades: En esta fase inicial, se establecen y analizan las necesidades del proyecto, incluyendo los requerimientos de la aplicación y el alcance del proyecto. Se identifican los principales objetivos y se definen los criterios de éxito (Castro, 2015).

Diseño y *feedback* con el usuario: Durante esta etapa, se realiza un diseño preliminar de la arquitectura del sistema. Se crean prototipos y se obtiene retroalimentación de los usuarios y otras partes interesadas. Esta retroalimentación



es crucial para refinar el diseño y garantizar que cumpla con los requisitos del usuario final.

Construcción: En esta fase, se lleva a cabo la implementación real del software. Se desarrollan los componentes de la aplicación, se realiza la codificación y se llevan a cabo pruebas de integración. El objetivo es construir un producto funcional.

Transición: Implica la entrega del producto final al entorno de producción en vivo. Aquí se realizan pruebas finales y se asegura que el *software* esté listo para su implementación. También se proporciona soporte y formación necesaria para los usuarios finales.

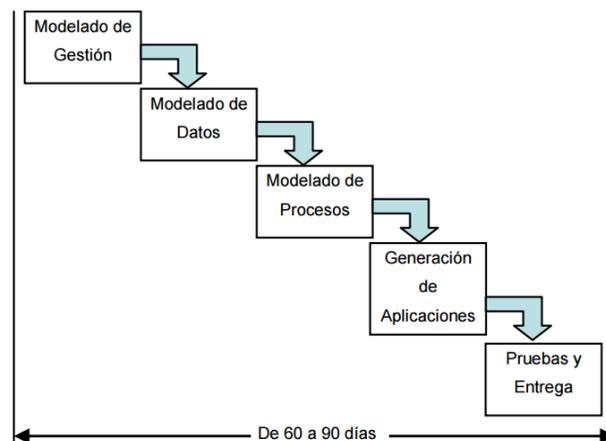


Ilustración 6 Modelo RAD

2.1.4 Metodología ágil

Este enfoque surge como una solución a las limitaciones que las metodologías tradicionales pueden generar y se sustenta en dos elementos esenciales: la postergación de decisiones y la planificación adaptativa. Se fundamenta en la capacidad de adaptación de los procesos de desarrollo. Un modelo de desarrollo ágil, por lo general, se caracteriza por ser un proceso incremental (con entregas frecuentes y ciclos rápidos), cooperativo (donde clientes y desarrolladores colaboran de manera constante mediante una comunicación detallada y continua), sencillo (el método es fácil de aprender y ajustar para el equipo) y, en última



instancia. Adaptativo (capaz de permitir cambios de último momento) (Gabriel, Maida Esteban, 2015).

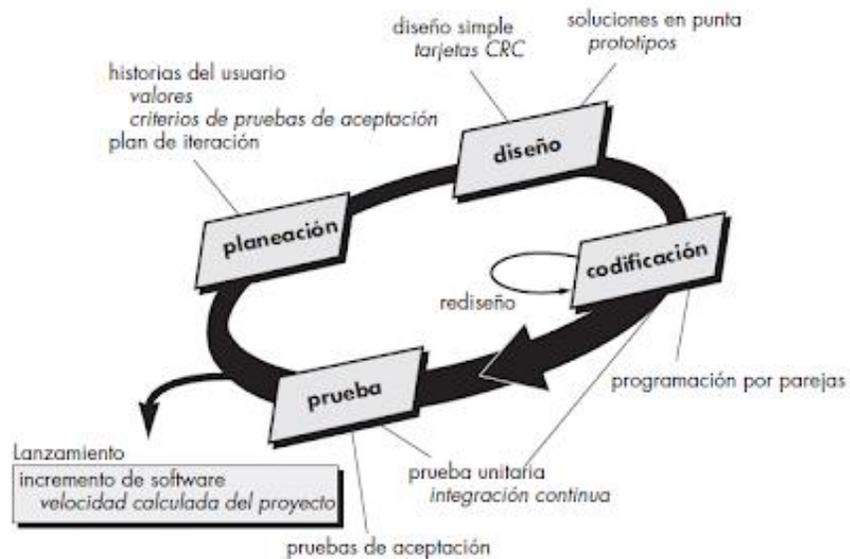


Ilustración 7 Metodología ágil Fuente: (Pressman 2005).

2.1.4.1 Extreme Programming

El origen de XP se remonta a los años 90, cuando *Kent Beck* –quien más tarde se convertiría en uno de los autores del Manifiesto Ágil– lo creó cuando fue contratado para liderar el equipo del Sistema Integral de Compensación de *Chrysler*.

La programación extrema (XP) es una metodología de desarrollo de *software* que pertenece al grupo de lo que se conoce comúnmente como metodologías ágiles. XP se fundamenta en valores, principios y prácticas, y busca posibilitar que equipos de tamaño reducido o mediano produzcan software de excelente calidad y se ajusten a los requisitos que cambian y evolucionan con el tiempo (Venema, 2024).



Consta de 4 fases:

- **Codificación**
- **Prueba**
- **Evaluación**
- **Diseño**

Lista del equipo XP:

Clientes: Establecen prioridades y marca del proyecto. Suelen ser usuarios finales del producto y quienes marcan las necesidades.

Programadores: Encargados de Desarrollar XP *Extreme Programming*.

Testers: Ayuda al cliente sobre los requisitos del producto.

Encargado: Asesoran al resto de los componentes del equipo y marcan el rumbo del proyecto.

Manager: Ofrecen recursos, es el responsable de la comunicación externa y quién coordina las actividades.

2.2 Comparación de metodologías

A continuación en la tabla (4), se presenta un resumen de una tabla comparativa entre metodologías estructuradas, orientadas a objetos, *UML-based web* y metodologías ágiles:

Tabla 4 Análisis de comparación de metodologías Fuente: Elaboración propia.

CARACTERÍSTICAS	METODOLOGÍA UML-BASE WEB	METODOLOGÍA ESTRUCTURADA	METODOLOGÍA A ORIENTADA A OBJETOS	METODOLOGÍA ÁGIL
Paradigma principal	Orientado a objetos	Procedimental	Orientado a objetos	Iterativo e incremental
Unidades de desarrollo	Objetos y componentes web	Funciones o procedimientos	Objetos y clase	Historias de usuario iteraciones
Reutilización de código	Alta	Limitada	Alta	Alta
Modificación y mantenimiento	Facilitado por UML	Complejo	Generalmente más sencillo	Adaptativo y flexible
Relación con UML	Total, integración	Escasa	Integración parcial	Utilización común de diagramas



Adaptabilidad a desarrollo web	Diseñada específicamente para desarrollo web	Menos adaptable	Adaptable, pero puede requerir ajustes	Muy adaptable y centrada en el cliente
Ejemplos de metodologías	Desarrollo ágil y Rational Unified Process (RUP)	Modelo en cascada o modelo en espiral	Desarrollo rápido de aplicaciones (RAD), SCRUM	Scrum, Kanban, XP, Crystal. Etc.

Esta tabla ofrece una visión general de las diferencias clave entre las metodologías estructuradas, orientadas a objetos, *UML-Based Web* y las metodologías ágiles en términos de paradigma, unidades de desarrollo, reutilización de código, mantenimiento, abstracción, relación con UML y adaptabilidad al desarrollo *web*. Cabe destacar que las metodologías ágiles se centran en la flexibilidad, adaptación continua y colaboración con el cliente.

2.3 Elección de metodología

Como se observa anteriormente existen diferentes tipos de metodologías para lo que es el desarrollo de aplicaciones del tipo *WEB* y son utilizadas dependiendo del tipo de aplicación y la forma de modelar la estructura de la aplicación. Dentro de cada tipo de metodología integran varias metodologías específicas.

A continuación, en la tabla 5 se expone una comparación sobre las Metodologías de desarrollo de aplicaciones *web* que integran Metodología de Gestión de Relaciones (RMM), Metodología *UWE (UML-BASE WEB)* y *SOHDM*.

Tabla 5 Análisis de comparación de metodologías de desarrollo para aplicaciones web Fuente: Elaboración propia.

Aspecto	RMM	<i>UWE (UML-BASE WEB)</i>	SOHDM
Enfoque	Centrado en la gestión de relaciones y la	Utiliza UML como lenguaje de modelado principal, con énfasis en el	Centrado en la ingeniería del <i>software</i> orientada a



	interacción con los clientes.	diseño de aplicaciones web.	servicios y en el diseño de aplicaciones <i>web</i> .
Base Teórica	Se basa en la teoría de la gestión de relaciones y la satisfacción del cliente.	Se basa en los principios de UML para el modelado y diseño de sistemas web.	Se basa en la ingeniería del <i>software</i> y en los servicios <i>web</i> orientados a la semántica.
Enfoque de Desarrollo	Orientado a la gestión de relaciones con clientes y usuarios.	Orientado al modelado y diseño de sistemas web utilizando UML.	Orientado a la ingeniería del <i>software</i> y al diseño de aplicaciones <i>web</i> basadas en servicios.
Componentes principales	Gestión de relaciones, interacción con clientes, análisis de necesidades.	Modelado UML, diagramas de casos de uso, diagramas de clases, diagramas de secuencia.	Ingeniería del <i>software</i> , diseño de servicios <i>web</i> , diseño orientado a la semántica.
Herramientas de apoyo	Software CRM, herramientas de análisis de clientes, encuestas de satisfacción.	Herramientas de modelado UML (<i>como Enterprise Architect, Visual Paradigm</i>).	Herramientas de ingeniería del <i>software</i> (como Eclipse, <i>NetBeans</i>), herramientas de servicios <i>web</i> .
Ventajas	Enfoque centrado en el cliente, mayor satisfacción del cliente, mejor gestión de relaciones.	Utilización de un lenguaje estándar para el modelado, diseño más estructurado y comprensible.	Orientación a servicios, diseño flexible y escalable, adaptabilidad a cambios.
Desafíos	Mayor enfoque en la gestión de relaciones puede descuidar otros aspectos del desarrollo.	Requiere conocimientos avanzados de UML, curva de aprendizaje para algunos desarrolladores.	Requiere experiencia en ingeniería del <i>software</i> , integración de servicios <i>web</i> puede ser compleja.

Observando que las características de las metodologías son similares. Así mismo, se eligió la metodología llamada ***UML-Based-Web o UWE***. Esta metodología se adapta al desarrollo de sistema de control de innovación empleando un desarrollo ágil, dicha metodología buscando aprovechar el poder del modelado para comprender, planificar y diseñar la arquitectura de la aplicación web de manera completa y detallada, proporcionando una variedad de diagramas (de clase, de



secuencia y de actividad) que permita visualizar de manera clara la estructura y el flujo de la aplicación *web*, facilitando la comunicación entre miembros del equipo, *stakeholders* y otros interesados, mejorando la comprensión del sistema, estableciendo una base sólida para la arquitectura de la aplicación *web* desde el principio.



Capítulo III: Análisis

3.1 Análisis del Sistema

En el área de Coordinación es necesario un sistema de control para llevar a cabo reportes de dicha área. Necesita sistematizar sus procesos en un sistema. Su evolución y desarrollo del sistema se utiliza la metodología “*Based-Web*”.

Identificando las necesidades y problemática el sistema en la Licenciatura de Ciencias Computacionales se abordaron aspectos como:

Tabla 6 Análisis de la problemática en el sistema y necesidad Fuente: Elaboración propia.

NECESIDAD	PROBLEMÁTICA
Sistema específico para el control de Actividades de Innovación Educativa.	Documentos manuales.
	Documentos digitales en diferentes formatos digitales (PDF, Hojas de cálculo).
	Procesos de ejecución tediosos.
	Rendimiento
	Control de datos ineficiente.

3.1.1 Requisitos de funcionalidad

Para garantizar el óptimo desempeño y funcionamiento del sistema, es esencial abordar unas series de puntos. En este sentido con base al sistema de Innovación Educativa para el Alumnado debe de contar con una interfaz amigable, fácil de usar, donde los usuarios finales son: coordinador, profesor y alumnos puedan hacer un fácil uso del sistema. Al mencionar lo anterior, debe permitir la autenticación y acceso al sistema, gestionando la monitorización y ejecución de acciones.



RF01: Registrar usuarios: El sistema permitirá al usuario registrarse ingresando información sobre el usuario, datos personales como su nombre, dirección de correo electrónico, teléfono, sexo, datos de contacto número de empleado, usuario, contraseña, perfil.

RF01: Autenticación de usuario: Se deberá identificarse el usuario para poder acceder al sistema.

RF03: Registros de Datos: El usuario deberá ingresar los datos correspondientes al hacer el reporte.

RF05: Consultar Información: El sistema ofrecerá al usuario información general acerca de los reportes:

- Consultar reportes: Muestra información de reportes enviados.
- Consultar horarios: Muestra información de horarios en general.

RF06: Roles de Seguridad: Este rol determinará los objetos a los que podrán acceder y las acciones que podrán llevar a cabo dentro del sistema.

3.1.2 Requisitos no funcionales

Los requisitos no funcionales son una parte fundamental en el desarrollo de sistema de control que va más allá de la funcionalidad directa. Al ejecutar una operación o reporte el rendimiento debe ser que no se degrade ya que se maneja datos en ciertas cantidades.

- **Seguridad:** Garantizar a los usuarios un rendimiento satisfactorio, así como la seguridad y la fiabilidad del sistema. Esto implica que los datos almacenados puedan ser actualizados y consultados al mismo tiempo sin que afecte la velocidad de respuesta.
- **Fiabilidad:** Garantizar que la interfaz sea amigable e intuitiva.
- **Disponibilidad:** Asegurar el funcionamiento continuo para el usuario.



3.1.3 Perspectiva del sistema

El sistema de control de Innovación funcionará con la metodología “*Based Web*” Al diseñar un sistema de control utilizando una metodología basada en *web*, es importante priorizar el diseño centrado en el usuario, la escalabilidad, la seguridad y el uso de tecnologías *web* modernas, mientras se mantiene un enfoque iterativo y colaborativo en el desarrollo del sistema.

3.1.4 Funcionalidad del sistema

Al diseñar el sistema de control utilizando la metodología “*Based Web*” la funcionalidad abarca gestión de usuario puede abarcar una amplia gama de características y capacidades, dependiendo de los requisitos específicos del sistema y las necesidades de los usuarios finales. Algunas de las funcionalidades comunes del sistema de control son:

Gestión de usuario y perfil: Permitir la creación de cuentas de usuario con diferentes accesos y roles dentro del sistema.

Módulos: Tendrá modulo de editar información personal del usuario registrado, modulo de información de horarios disponibles de los docentes para el alumnado.

Generación de informes y análisis: Generación de reporte y análisis de datos para ayudar a los docentes y a la coordinación de Ciencias Computacionales.

Registro de Actividades: Registrar todas las actividades realizadas en el sistema. Acciones por los usuarios (docente, coordinador) y cualquier otro evento relevante.



3.1.5 Especificación de requerimientos

Tabla 7 Análisis de requisito funcional registro de usuario Fuente: Elaboración propia.

Número de requisito	RF1
Nombre de requisito	Registro de usuarios
Tipo	Requisitos
Prioridad del requisito	Alta/Esencial
Descripción	Esto generalmente implica recopilar información sobre el usuario, datos personales como su nombre, dirección de correo electrónico y de contacto nombre de usuario deseado y establecer una contraseña inicial.

Tabla 8 Análisis de requisito funcional autenticación de usuario Fuente: Elaboración propia.

Número de requisito	RF2
Nombre de requisito	Autenticación de usuarios
Tipo	Requisitos
Prioridad del requisito	Alta/Esencial
Descripción	Proceso de verificar la identidad de un usuario que intenta acceder al sistema, que ayuda a garantizar que solo las personas autorizadas puedan utilizar el sistema y acceder a sus recursos.

Tabla 9 Análisis de requisito funcional Registro de actividades Fuente: Elaboración propia.

Número de requisito	RF3
Nombre de requisito	Registros de Actividades
Tipo	Requisitos
Prioridad del requisito	Alta/Esencial
Descripción	Registrar todas las actividades realizadas en el sistema. Acciones por los usuarios y cualquier otro evento relevante.



Tabla 10 Análisis de requisito funcional generación de reportes Fuente: Elaboración propia.

Número de requisito	RF4
Nombre de requisito	Generación de Informes
Tipo	Requisitos
Prioridad del requisito	Alta/Esencial
Descripción	El sistema ofrecerá al usuario información general.

Tabla 11 Análisis de requisito funcional roles de seguridad Fuente: Elaboración propia.

Número de requisito	RF5
Nombre de requisito	Roles de Seguridad
Tipo	Restricción
Prioridad del requisito	Alta/Esencial
Descripción	Este rol determinará los objetos a los que podrán acceder y las acciones que podrán llevar a cabo dentro del sistema.

3.1.6 Proceso de ingeniería de requerimientos

Adquisición y análisis de requerimientos

La ingeniería de requerimientos desempeña un papel fundamental para comprender las necesidades del cliente, definir los objetivos del sistema y especificar los requisitos funcionales y no funcionales. Con la creciente disponibilidad de herramientas y plataformas en línea, la metodología "Based Web" ofrece un enfoque colaborativo y accesible para llevar a cabo este proceso.

- Definición de los objetivos del sistema
- Recopilación y Análisis de Requerimientos
- Roles de Seguridad



Desde la identificación de los interesados hasta la validación de los requisitos y la gestión de cambios, las herramientas en línea ofrecen una plataforma centralizada para realizar todo el ciclo de vida de la ingeniería de requerimientos.

A través de un enfoque moderno y orientado a la colaboración, esta metodología busca optimizar la comunicación entre los diferentes interesados y mejorar la calidad y precisión de los requisitos identificados, lo que resulta en un sistema de control que cumple de manera efectiva con las necesidades del cliente y los usuarios finales.

Roles de seguridad

Los usuarios tienen ciertos roles de seguridad en el sistema, según el nivel de protección que tendrá en el flujo del sistema, teniendo autorización y limitaciones que se asignan como permisos al usuario dentro del sistema una vez registrado. La siguiente tabla (12) se describe cada rol que tiene en el sistema.

Tabla 12 Roles de seguridad Elaboración propia

Rol	Funciones	Transacciones de Acceso
Administrador del sistema	Encargado de administrar el control de usuarios, asignación de roles y permisos en el sistema.	Acceso total
Capturista	Responsable de gestionar los reportes dentro del sistema.	Tanto como el docente y el coordinador tendrá que capturar los reportes de las Actividades de Innovación Educativa. Sin embargo, solo el coordinador tendrá el acceso a ver los reportes realizados por el docente.



Capítulo IV: Diseño

En esta etapa se elaboran los modelos requeridos para organizar el contenido, datos y la navegación a través de las páginas, lo que resulta fundamentalmente para entender cómo se desarrollará el diseño y la funcionalidad del sistema al crear. Los modelos incluidos en este proceso son: El modelo conceptual, El modelo navegacional, prototipos de la interfaz de usuario y los modelos UML.

4.1 Base de datos

Las bases de datos juegan un papel fundamental en un sistema de control, ya que son la columna vertebral que permite almacenar, organizar y gestionar la información necesaria, el cual mantiene integridad y consistencia de los datos. Esto significa que los datos almacenados en la base de datos están sujetos a reglas y restricciones que garantizan su calidad y coherencia, evitando la duplicación o la pérdida de información.

Tablas y campos

La estructura de la base de datos ha sido diseñada con un enfoque lógico que se sustenta en varias tablas interrelacionadas. Estas tablas han sido cuidadosamente planificadas para garantizar la eficiencia, coherencia y fiabilidad del sistema. Cada tabla representa una entidad específica o un conjunto de entidades relacionadas entre sí, lo que permite organizar y almacenar la información de manera ordenada y accesible. El diseño de estas tablas se basa en las necesidades y requerimientos del sistema. Las cuales son 5 tablas, estas son: tabla usuarios, tabla de reportes, tabla de tipo de reportes, tabla materias de reportes y tabla de semestre.



4.1.1 Modelo Entidad-Relación (E-R)

En este modelo se desarrollan las tablas de la base de datos para resguardar la información que se genere por el usuario, del sistema *web*. En la tabla usuario se almacenará los datos personales como atributos. La tabla reportes tiene como atributos un identificador de reporte y los nombres de los reportes y existe una relación de uno a muchos.

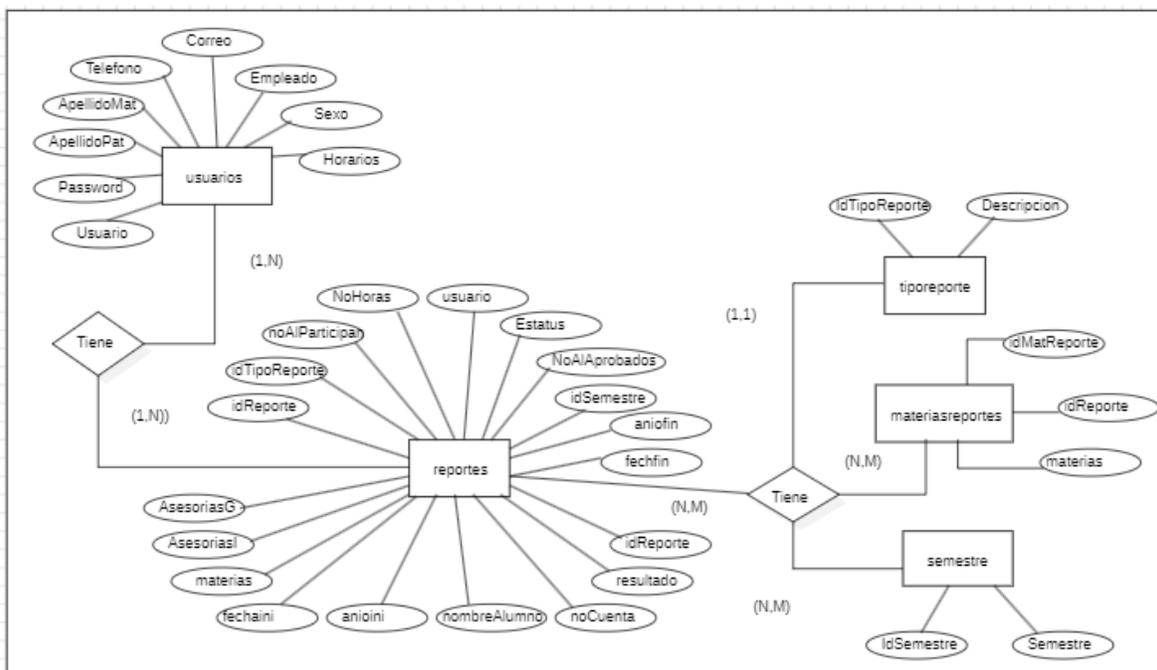


Ilustración 8 Modelo Entidad-Relación E-R Fuente: Elaboración propia.

4.1.2 Modelo Relacional

Su idea fundamental es el uso de relaciones. Las relaciones se pueden establecer entre diferentes tablas a través de claves primarias y claves externas, permitiendo así representar las relaciones entre entidades del mundo real.

En el sistema de control de reportes de Innovación se identifican los actores: profesor, coordinador, alumno donde cada actor tiene privilegios al ingresar al sistema.



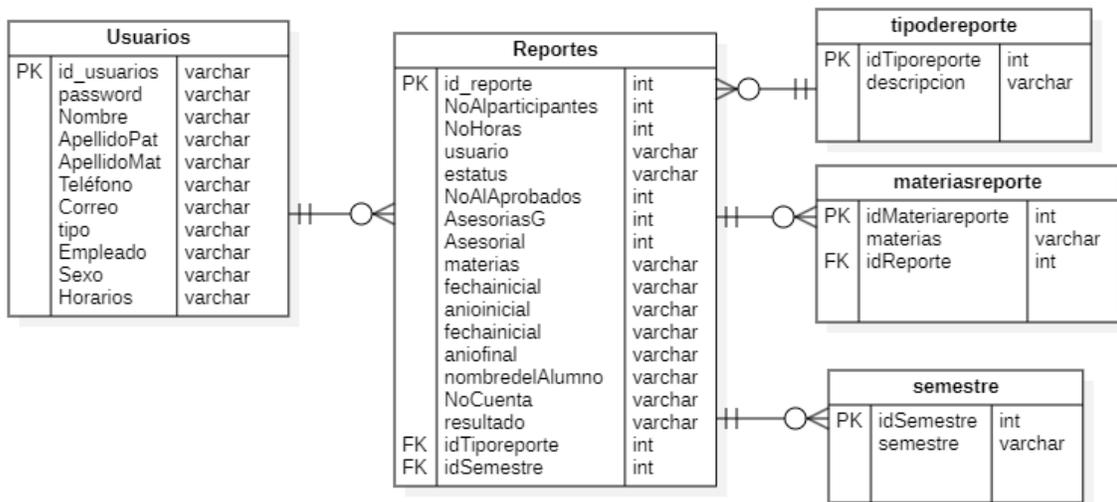


Ilustración 9 Modelo Relacional Fuente: Elaboración propia.

Usuarios

Esta tabla almacenará la información personal y de contacto como también los horarios en el que se encontrará el docente. Los cuales se muestran en la siguiente tabla (13) que se describe los atributos.

- **Usuario:** Nombre del usuario con el que ingresará al sistema.
- **Password:** Ingresar una contraseña para ingresar al sistema.
- **Nombre:** Apellido paterno del usuario.
- **Apellidoopat:** Apellido materno del usuario.
- **Apellidomat:** Número telefónico del usuario.
- **Teléfono:** Correo institucional o personal del usuario.
- **Correo:** Número de empleado del usuario.
- **Sexo:** Genero del usuario.
- **Horarios:** Horarios del usuario y lugar en donde se puede encontrar.

Tabla 13 Análisis para la tabla usuarios Fuente: Elaboración propia.



usuarios		
ÍNDICE	CAMPO	TIPO
PK	Usuario	varchar
	Pastor	varchar
	Nombre	varchar
	ApellidoPat	varchar
	ApellidoMat	varchar
	teléfono	varchar
	Correo	varchar
	Empleado	varchar
	Sexo	varchar
	Horarios	varchar

Reportes

En esta tabla se registran los reportes con los siguientes atributos.

- **idReporte:** Identificador de número de reporte.
- **idTipoReporte:** Identificador de tipo de reporte.
- **NoAlParticipa:** Número total de alumnos participantes.
- **NoHoras:** Número de horas que durara el curso
- **Usuario:** Docente que está creando reporte.
- **Estatus:** Estado del reporte si ya fue terminado o pendiente.
- **NoAlAprobados:** Número de alumnos aprobados.
- **IdSemestre:** Identificador de número de semestre.
- **asesoriasG:** Numero de asesorías grupales.
- **asesoriasI:** Numero de asesorías individuales.
- **materias:** Nombre de la materia o asignatura.
- **fechaini:** Fecha en la que inicio curso o asesoría.
- **anioini:** Año en el que inicio curso o asesoría.
- **fechafin:** Fecha en la que finalizo curso o asesoría.
- **aniofin:** Año en la que finalizo curso o asesoría.
- **nombreAlumno:** Nombre del alumno
- **noCuenta:** Número de cuenta
- **resultado:** Calificación obtenida



Tabla 14 Análisis de la tabla reportes Fuente: Elaboración propia.

reportes		
ÍNDICE	CAMPO	TIPO
PK	idReporte	int
	idTipoReporte	int
	NoAlParticipa	int
	NoHoras	int
	usuario	varchar
	Estatus	varchar
	NoAlAprobados	int
	IdSemestre	int
	AsesoriasG	int
	AsesoriasI	int
	materias	varchar
	fechaini	varchar
	anioini	varchar
	fechafin	varchar
	aniofin	varchar
	nombreAlumno	varchar
	noCuenta	varchar
	resultado	varchar

Tipo reporte: En esta tabla (15), se presenta los diferentes tipos de reportes y su identificador.

- **idTipoReporte:** Identificador de Numero de reporte
- **Descripcion:** Describe el tipo de reporte que es.

Tabla 15 Análisis de la tabla de tipo reporte Fuente: Elaboración propia.

tiporeporte		
ÍNDICE	CAMPO	TIPO
PK	idTipoReporte	int
	descripción	varchar

Materias reporte: Se tienen las materias en la tabla (16), que se impartieron o se aplicaron en el curso.

- **IdMatReporte:** Identificador de materias del reporte



- **IdReporte:** Identificador de número de reporte
- **materias:** Nombre de la materia o asignatura

Tabla 16 Análisis de la tabla materia reportes Fuente: Elaboración propia.

materiasreportes		
ÍNDICE	CAMPO	TIPO
PK	idMatReporte	int
	idReporte	int
	materias	varchar

Semestre: Los reportes tienen un semestre de acuerdo con el curso o actividad educativa en el que se aplicara. En la siguiente tabla (17), se describe el atributo.

- **idSemestre** Identificador del semestre
- **Semestre** Semestre en el que se aplicara el curso o actividad educativa.

Tabla 17 Análisis de la tabla de semestre Fuente: Elaboración propia.

semestre		
ÍNDICE	CAMPO	TIPO
PK	idSemestre	int
	semestre	varchar

4.2 Diagramas UML

La introducción de los diagramas UML (Lenguaje Unificado de Modelado) en la metodología "Based Web" en el diseño de un sistema de control proporciona una estructura visual y comprensible para representar la arquitectura, la navegación y la interacción del sistema. Estos diagramas UML ayudan a los diseñadores y desarrolladores a conceptualizar, comunicar y documentar aspectos clave del sistema, lo que facilita la comprensión y la colaboración entre los diferentes equipos involucrados en el proyecto.



Algunos de los diagramas UML que pueden ser útiles en este contexto incluyen:

4..1 Diagrama de Caso de Uso

Esta herramienta sirve para tener una visión clara y detallada de las interacciones entre los actores y el sistema de control, identificando las funciones y características principales, desde la perspectiva del usuario como se visualiza en la ilustración (10). Describiendo cada acción de la función que hace el actor.

1.Registro de Usuario:

Actores:

- Actor1 Coordinador
- Actor2 Profesor

Descripción: Permite a un usuario crear una cuenta en el sistema proporcionando la información necesaria.

2.Inicio de Sesión:

Actores:

- Actor1 Coordinador
- Actor2 Profesor

Descripción: Permite a los usuarios autenticarse en el sistema utilizando sus credenciales.

3.Ver Perfil:

Actores:

- Actor1 Coordinador
- Actor2 Profesor

Descripción: Permite al usuario ver su información personal, como nombre, número de empleado, etc.



4.Modificar Perfil:

Actores:

- Actor1 Coordinador
- Actor2 Profesor

Descripción: Permite al usuario editar su información personal, como nombre, número de empleado, etc.

5.Crear Reporte:

Actores:

- Actor1 Coordinador
- Actor2 Profesor

Descripción: Permite crear a los usuarios reporte.

6.Visualización Detalles de Reporte:

Actores:

- Actor1 Coordinador

Descripción: Permite al usuario ver detalles específicos, como visualizar reportes.

7. Modificar reportes:

Actores:

- Actor1 Coordinador
- Actor2 Profesor

Descripción: Permite a los usuarios modificar los datos de los reportes, antes de mandar el reporte final.

8.Visualización de horarios:

Actores:



- Actor1 Coordinador
- Actor2 Profesor
- Actor3 Alumnado

Descripción: Permite a los usuarios ver detalles de horarios disponibles.

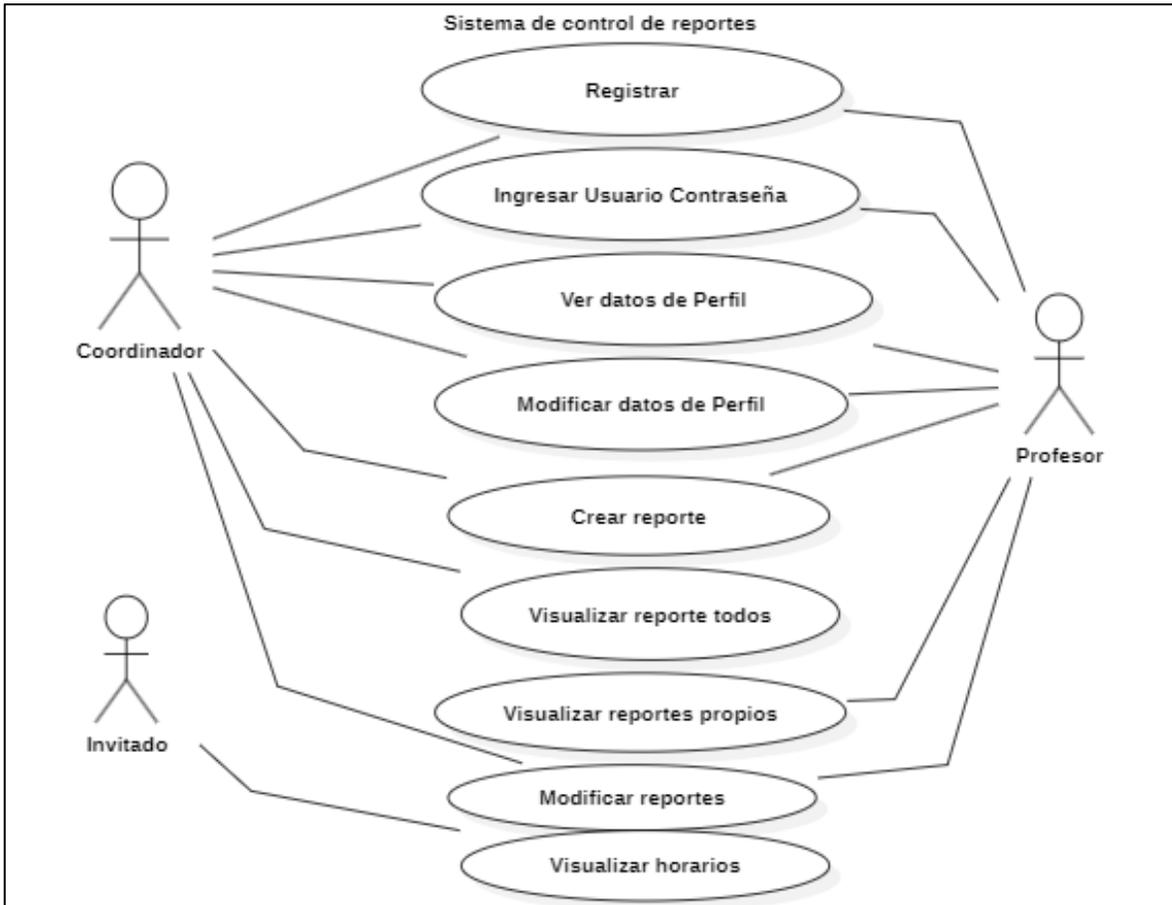


Ilustración 10 Diagrama de Caso de Uso Fuente: Elaboración propia.

4.2.2 Diagrama Navegacional

Se centra en la estructura y la navegación de un sitio web. Este modelo describe cómo están conectadas entre sí las diferentes páginas o secciones del sitio, así como las interacciones que pueden ocurrir entre ellas. Utiliza nodos para representar las páginas individuales del sitio web y enlaces para mostrar las conexiones entre estas páginas. El objetivo principal de este modelo es proporcionar una representación clara y comprensible de la arquitectura de navegación del sitio



web, lo que facilita el diseño y la comprensión de cómo los usuarios pueden moverse a través del sitio (UWE, 2016).

Los componentes de modelado incluyen:

Clases de navegación.

El usuario realiza el registro en el sistema donde ingresa datos personales, elige rol en el sistema y genera su perfil, elige llenar extras del perfil, o cerrar sesión, si el elige datos extras, guarda la información. El usuario crea un reporte en el sistema modificando y visualizarlo.

Enlaces de navegación.

Muestran las conexiones directas entre las clases de navegación del sistema de control de la Licenciatura de Ciencias Computacionales.

Rutas de navegación alternativas.

Que se presentan con el estereotipo <<menú>>.

Primitivas de acceso.

Se utilizan para acceder a múltiples instancias de una clase de navegación (<<índice>> o <<visita guiada>>) o para seleccionar ítems (<<consulta>>).

Clases de procesos

Modelan los puntos de entrada y salida de los procesos de negocio, cada una asociada a un caso de uso de proceso.



Como se puede ver en la siguiente ilustración (11), se En esta representan la conexión entre las clases de proceso y navegación.

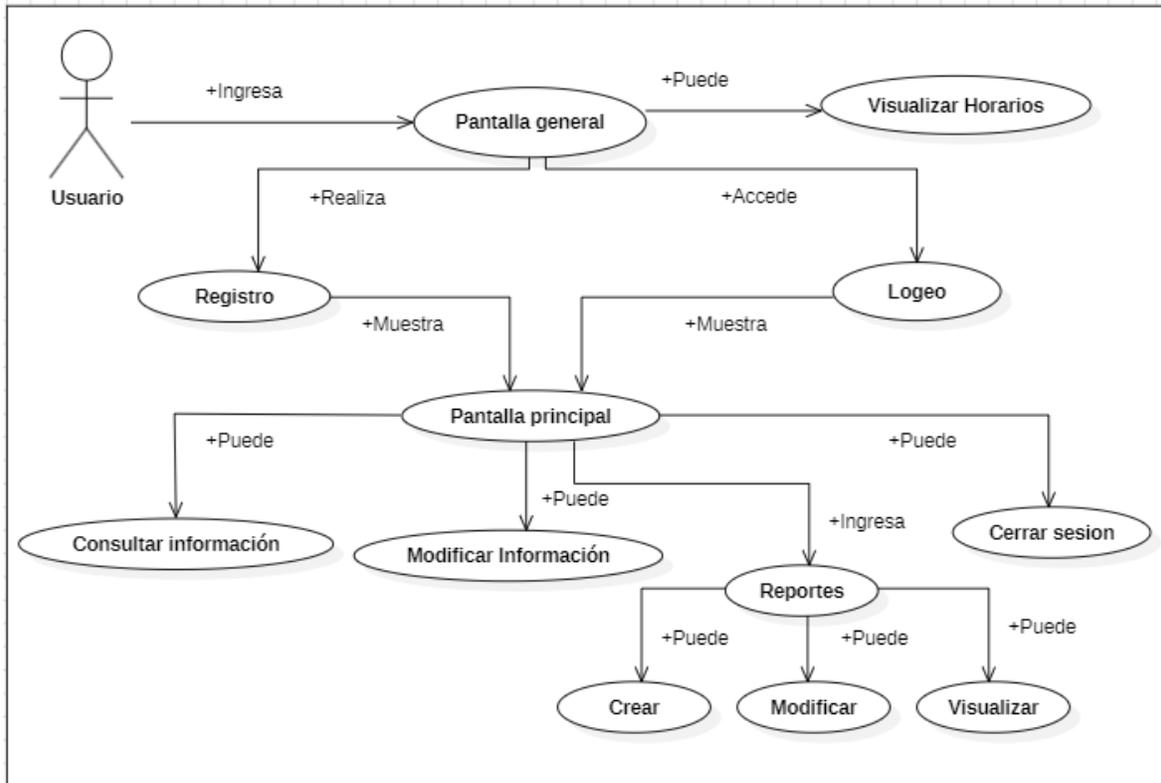


Ilustración 11 Diagrama navegacional Fuente: Elaboración propia

4.2.3 Diagrama de Actividad

Un diagrama de actividad, perteneciente al lenguaje de modelado unificado (UML), es una representación gráfica del flujo de trabajo de las actividades dentro de un sistema, utilizando símbolos y reglas definidas formalmente. Principalmente empleado en el modelado de sistemas de software, los diagramas de actividad en UML se consideran herramientas de comportamiento, ya que delimitan las acciones que deben ocurrir en los sistemas, funcionando como modelos de estos (Servicenow, s.f.).

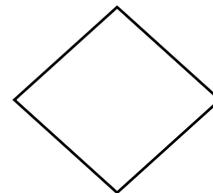
Presenta el flujo de trabajo o los procesos del sistema, representando acciones y decisiones.



Este diagrama de Actividades en la ilustración (13), demuestra la lógica del algoritmo, ilustrando los flujos del trabajo entre Usuarios, Coordinador, Docente y sistema, simplificando y mejorando los procesos.

Actividades

- Actividad Inicial: Ingresar al sistema
- Actividad1: Ingresar datos
- Actividad2: Elegir perfil
- Actividad3: Visualizar datos
- Actividad4: Agregar horarios
- Actividad5: Modificar reportes
- Actividad6: Guardar datos
- Actividad7: Cerrar sesión
- Actividad8: Coordinador
- Actividad9: Profesor
- Actividad10: Crear reporte
- Actividad 11: Buscar reporte
- Actividad 12: Guardar reporte
- Actividad 13: Reporte terminado
- Actividad 14 Visualizar reporte



Decisiones

- Decisión 1: Registrarse o Elegir perfil
- Decisión 2: Agregar horarios o no
- Decisión 3: Crear o buscar reporte
- Decisión 4: Reporte terminado



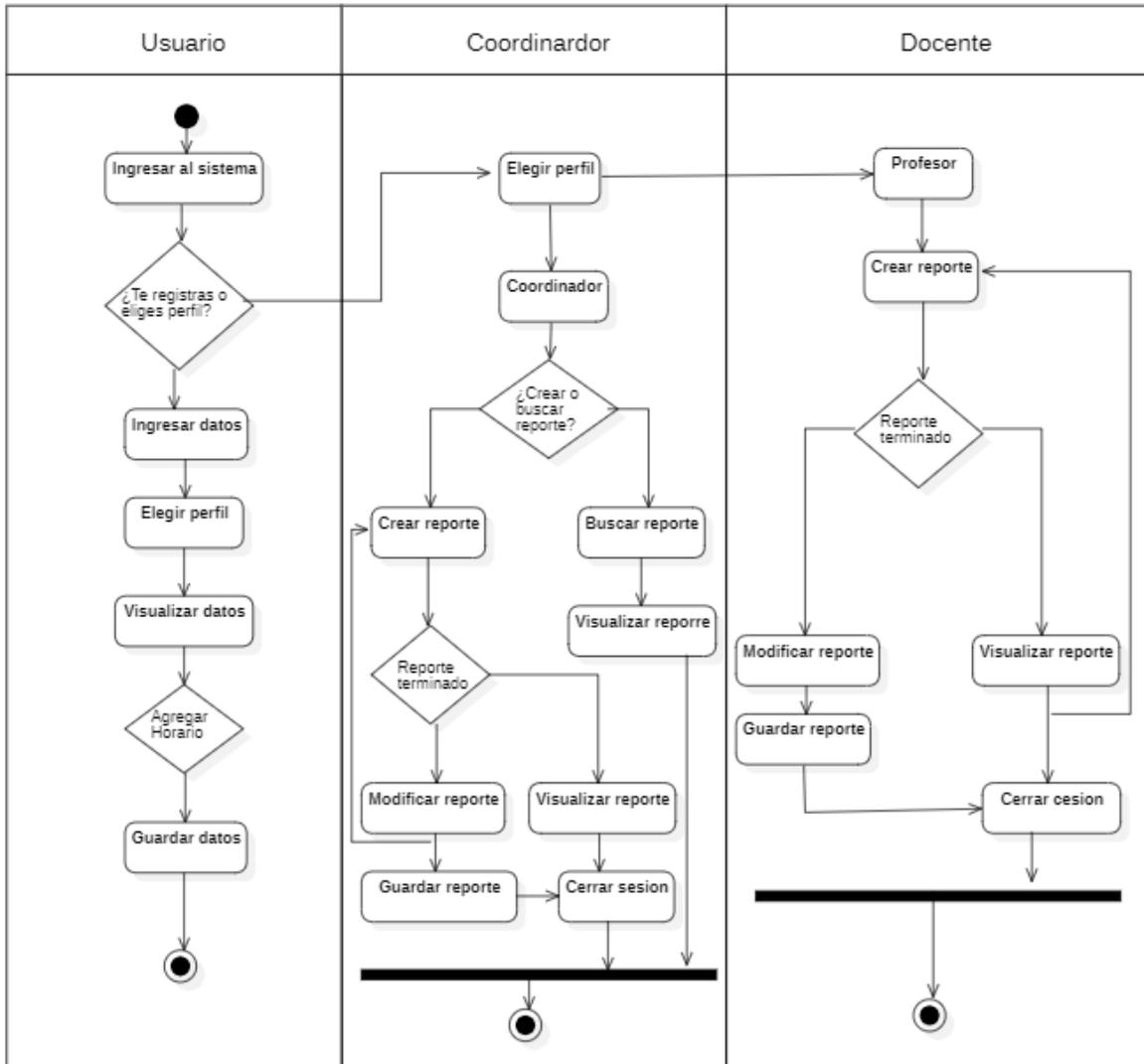


Ilustración 12 Diagrama de Actividades Fuente: Elaboración propia.

4.2.4 Diagrama de Clases

Los diagramas de clases del Lenguaje de Modelado Unificado (UML) desempeñan un papel similar al de un plano en el desarrollo de software. Estos diagramas, al igual que un plano, son utilizados tanto al inicio como durante la actualización del desarrollo de un software. Forman parte de una serie de diagramas UML que representan distintos aspectos del desarrollo, todos ellos empleando un mismo lenguaje para facilitar su interpretación.



Para representar la estructura estática del sistema de control de Innovación Educativa para la Licenciatura de ciencias computacionales, se ocuparán las siguientes clases, atributos y relaciones entre ellas.

1. Clases 1: Usuarios

- Atributo 1 Usuario
- Atributo 2 Password
- Atributo 3 Nombre
- Atributo 4 Apellido Paterno
- Atributo 5 Apellido Materno
- Atributo 6 Teléfono
- Atributo 7 Correo
- Atributo 8 Empleo
- Atributo 9 Sexo
- Atributo 10 Horarios

2. Clases 2: Reportes

- Atributo 1 Id reporte
- Atributo 2 Id tipo de Reporte
- Atributo 3 Número de Alumnos participantes
- Atributo 4 Número de Horas
- Atributo 5 Usuario
- Atributo 6 Estatus
- Atributo 7 Número de Alumnos Aprobados
- Atributo 8 Id semestre
- Atributo 9 Asesorías Grupales
- Atributo 10 Asesorías Individuales
- Atributo 11 Materias
- Atributo 12 Fecha Inicial
- Atributo 13 Año Inicial
- Atributo 14 Fecha Final
- Atributo 15 Año Final



- Atributo 16 Nombre de Alumno
- Atributo 17 Número de Cuenta
- Atributo 18 Resultado

3. Clase 3: Tipo de reporte

- Atributo 1 Id Tipo de Reporte
- Atributo 2 Descripción del Reporte

4. Clase 4: Materia reportes

- Atributo 1 Id Materia de Reporte
- Atributo 2 Id Reporte
- Atributo 3 Materias

5. Clase 5: Semestre

- Atributo 1 Id Semestre
- Atributo 2 Semestre

Relaciones

- El usuario tiene relación de uno a muchos con la clase reportes.
- La clase reportes tiene relación de muchos a muchos con la clase materia reportes y la clase semestre.
- La clase reportes tiene relación de uno a uno con la clase tipo reporte.



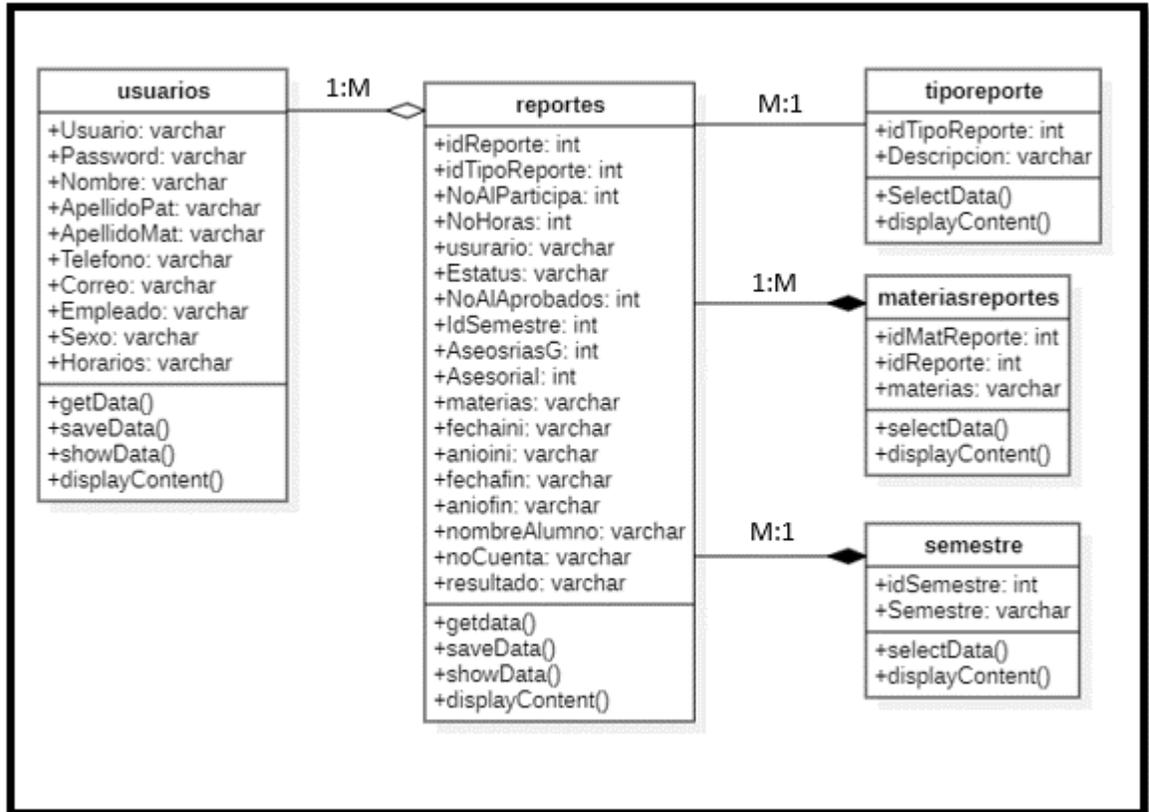


Ilustración 13 Diagrama de clase Fuente: Elaboración propia



4.3 Prototipo de Interfaz del sistema de control de Innovación Educativa para el Alumnado

Interfaz Principal del Sistema

Las interfaces de sistema de control de Innovación educativa como prototipo se van a basar en la portada principal del sistema en donde se presentará el escudo de la universidad nombre de la aplicación y tres menús ya sea para registrarse o elegir el perfil de Coordinador o Profesor, y horarios disponibles de los docentes. Como se presenta en la siguiente ilustración (15).

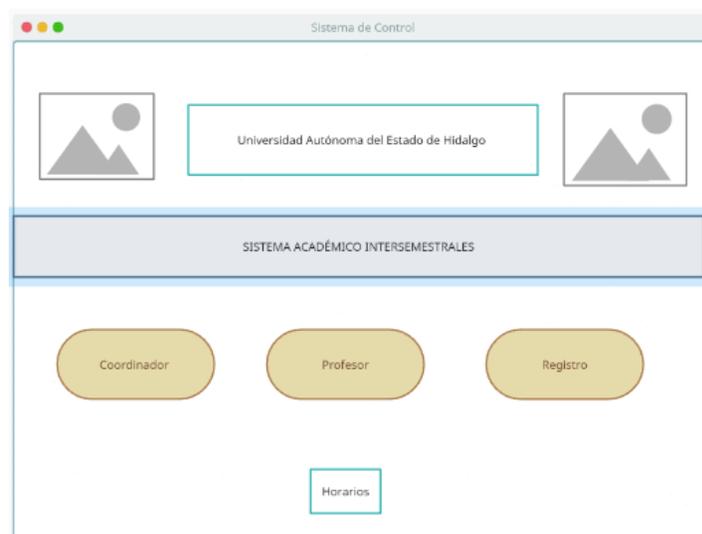


Ilustración 14 Prototipo de Interfaz Inicial Fuente Elaboración propia.



Interfaz de Registro

Cuando el usuario ingrese al apartado de registro, en donde se muestra la ilustración (16), tiene que ingresar sus datos personales y datos de contacto.



Ilustración 15 muestra un prototipo de la interfaz de registro. El título de la ventana es "Registro". En la parte superior, se muestra el logo de la "UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO" flanqueado por dos imágenes de paisaje. Debajo de esto, hay una barra de navegación con el texto "Reistro". El formulario principal contiene cinco campos de texto: "Nombre", "Apellido Paterno", "Apellido Materno", "Teléfono" y "Correo". En la parte inferior del formulario, hay un botón rojo con el texto "Continuar".

Ilustración 15 Prototipo de Interfaz de registro Fuente: Elaboración propia.

Interfaz de Inicio de Sesión

El usuario ingresará el nombre de usuario y password el sistema realizará la validación de sus credenciales.



Ilustración 16 muestra un prototipo de la interfaz de inicio de sesión. El título de la ventana es "Acceso". En la parte superior, se muestra el logo de la "UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO" flanqueado por dos imágenes de paisaje. Debajo de esto, hay una barra de navegación con el texto "Acceso". El formulario principal contiene dos campos de texto: "Usuario" y "Contraseña". Debajo de los campos, hay tres botones de inicio de sesión: "Coordinador", "Profesor" y "Registro". En la parte inferior del formulario, hay un botón con el texto "Horarios".

Ilustración 16 Prototipo de Interfaz Inicio de sesión Fuente: Elaboración propia.



Interfaz de Reportes

El usuario de acuerdo con su perfil creará o visualizará los reportes de Actividades de Innovación Educativo



Ilustración 17 Prototipo de Interfaz de reportes Fuente: Elaboración propia.

Interfaz de Modificación de Información de Usuario

El usuario podrá modificar los datos personales a excepción de usuario ya que es una llave única que idéntica cada usuario.



Ilustración 18 Prototipo de Interfaz modificación de datos usuario Fuente: Elaboración propia.



Conclusiones

Desde los primeros momentos de la existencia en la Tierra hasta hoy, ha sido evidente que todo lo que se crea, ya sean sistemas, servicios o productos innovadores, tiene su origen en una necesidad fundamental, que se busca satisfacer al crearlo con diversas etapas para su elaboración.

Centrándose en las etapas de análisis y diseño son aspectos para llevar a cabo el sistema hacía el éxito. Estas etapas no solo permiten comprender las necesidades y requisitos del usuario, sino también identificar riesgos, optimizar recursos y garantizar que el sistema sea eficiente, seguro y adaptable a los cambios. Al centrarse en estas etapas iniciales, se establece una base sólida, para el futuro desarrollo e implementación del sistema de control, que cumpla con las expectativas del usuario, mejore la eficiencia operativa y contribuya al éxito general de la coordinación. De igual manera, se implementa una metodología que proporciona una estructura organizada para facilitar la medición del desempeño, lo que permite la adaptación a cambios en el entorno.

Al elaborar futuramente el sistema, unos de sus beneficios tendrán una mejor visión clara, optimizando procesos más eficientes y actualizados, ya que hoy en día, el uso de nuevas herramientas tecnológicas es fundamental porque al menos una vez al día, el ser humano tiene contacto con un dispositivo electrónico.



Trabajos futuros

El análisis y diseño son etapas críticas en el desarrollo e implementación de cualquier proyecto, especialmente cuando se trata de soluciones destinadas a un área específica, en este caso en la coordinación de la Licenciatura de Ciencias Computacionales. Estas fases son fundamentales para comprender a fondo todos los requisitos, desafíos y objetivos del área, se espera que, con esto ayude a su desarrollo e implementación en el área indicada.



Glosario

UML: acrónimo de Lenguaje de Modelado Unificado, es una herramienta genérica empleada en el análisis, diseño e implementación de sistemas de software. Su objetivo primordial es ofrecer un método estandarizado y accesible para representar visualmente las características arquitectónicas fundamentales de un sistema de software.

Sistema de Control: Conjunto de componentes y procesos diseñados para supervisar y regular el funcionamiento de un sistema, máquina o proceso.

Interfaz de Usuario (UI): La parte del sistema que permite a los usuarios interactuar con él. En un sistema de control web, esto podría incluir formularios, botones y paneles de control.

Navegación: El proceso de moverse entre diferentes partes de un sistema de control web. Esto implica la estructura y la disposición de los enlaces y botones que permiten a los usuarios moverse de una página a otra.

Autenticación: Proceso de verificación de la identidad de un usuario antes de permitirle acceder al sistema de control. Esto puede incluir la entrada de credenciales como nombre de usuario y contraseña.

Autorización: Determinación de los permisos y privilegios que tiene un usuario dentro del sistema de control. Esto puede incluir la capacidad de ver, modificar o eliminar ciertos datos o configuraciones.

Panel de Control: Una interfaz de usuario que muestra información y permite al usuario controlar diferentes aspectos del sistema de control. Puede incluir gráficos, tablas y controles deslizantes para ajustar configuraciones.

Registro de Actividades: Un registro que guarda un historial de las acciones realizadas por los usuarios dentro del sistema de control. Esto puede incluir cambios en la configuración, eventos de alarma y otras actividades relevantes.



Notificaciones: Mensajes automáticos que se envían a los usuarios para informarles sobre eventos importantes o cambios en el sistema de control. Esto puede incluir alertas de errores, recordatorios de tareas pendientes, etc.

Seguridad: Conjunto de medidas y protocolos diseñados para proteger el sistema de control contra accesos no autorizados, ataques cibernéticos y otros riesgos de seguridad.



Referencias

- Agustín, G. C. (s.f.). *Ingeniería del software Práctica del software*. RA-MA.
- arsys. (15 de 06 de 2023). *Tipos de bases de datos que existen* . Obtenido de <https://www.arsys.es/blog/tipos-de-bases-de-datos-que-existen>
- Campos, M. S. (3 de 10 de 2021). *Virtualpro*. Obtenido de Control de sistemas: ejemplos y aplicaciones: <https://www.virtualpro.co/noticias/control-de-sistemas--ejemplos-y-aplicaciones>
- Castro, M. (2015). *Incentro*. Obtenido de Metodología RAD o DRA. El Desarrollo Rápido de Aplicaciones: https://www.incentro.com/es-ES/blog/metodologia-rad-desarrollo-rapido-aplicaciones#toc_link_FasesdentrodeunprocesoconmetodologaRAD
- Comas Rodríguez, R. N., & Medina León, A. (Mayo-Agosto de 2013). *Ciencias de la Información*. Obtenido de Análisis evolutivo de los sistemas de información y su marco conceptual: <https://www.redalyc.org/pdf/1814/181430077002.pdf>
- Digital guide IONOS. (18 de 1 de 2023). *Ventajas e inconvenientes del modelo de base de datos orientada a objetos*. Obtenido de <https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/base-de-datos-orientada-a-objetos/>
- Equipo editorial, Etecé. (19 de noviembre de 2023). *Concepto*. Obtenido de "Base de datos": <https://concepto.de/base-de-datos/>
- Fontela, C. (2011). *UML modelado de software para profesionales*. Alfa Omega.
- Gabriel, M. E. (2015). Obtenido de Metodologías de desarrollo de software: <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Gabriel, M. E. (2015). *Metodologías de desarrollo de software*. Obtenido de <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Gabriel, M. E. (s.f.). *Metodologías de desarrollo de software*. Obtenido de <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Gabriel, Maida Esteban. (2015). Obtenido de Metodologías de desarrollo de software: <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Gabriela, A. (2009). *Metodologías estructuradas*. Obtenido de <http://metodologiasestructuradas.blogspot.com/2009/02/metodologias-estructuradas.html>



- GoogleCloud. (s.f.). *¿Qué es una base de datos relacional?* Obtenido de <https://cloud.google.com/learn/what-is-a-relational-database?hl=es-419#:~:text=Una%20base%20de%20datos%20relacional%20es%20una%20colecci%C3%B3n%20de%20informaci%C3%B3n,estructuras%20de%20datos%20entre%20s%C3%AD.>
- Hughes, A. (Abril de 2021). *Microsoft SQL Server*. Obtenido de <https://www.computerweekly.com/es/definicion/Microsoft-SQL-Server>
- IONOS. (12 de Septiembre de 2023). *IONOS*. Obtenido de Modelo en espiral: el modelo para la gestión de riesgos en el desarrollo de software: <https://www.ionos.mx/startupguide/productividad/modelo-en-espiral/>
- Lora, A. R. (Noviembre de 2004). Construcción de una base de datos para un sistema web en apoyo al proceso de enseñanza.
- Mario Alberto, A. P. (2007). *Departamento de Electrónica y Automática*. Obtenido de Introducción a los sistemas de control y modelo matemático para sistemas lineales: <http://dea.unsj.edu.ar/control1/apuntes/unidad1y2.pdf>
- Molina Ríos, J.R., Zea Ordóñez, M.P., Contento Segarra, M.J. y García Zerda, F.G. (Marzo de 2018). *3 ciencias*. Obtenido de Comparación de metodologías en aplicaciones web: <https://3ciencias.com/wp-content/uploads/2018/03/art1.pdf>
- OCI. (2024). *¿Qué es MySQL?* Obtenido de <https://www.oracle.com/mx/mysql/what-is-mysql/>
- Octavio Islas Carmona, G. G. (2000). *Internet: el medio inteligente*. México: Continental México.
- Oracle. (2024). *¿Qué es una base de datos?* . Obtenido de <https://www.oracle.com/mx/database/what-is-database/>
- Pérez Rodríguez, Y. S. (2007). *Repositorio.uci.cu*. Obtenido de La metodología RMM aplicable al desarrollo del software educativo multimedia topografía: https://repositorio.uci.cu/jspui/bitstream/ident/TD_0851_07/1/TD_0851_07.pdf
- Potter, J. (1999). *Aprenda rápido Internet* . Alfaomega.
- Ramos Martín, A. R. (2015). *¿Que es un sistema web?* Obtenido de <https://www.addappto.com/que-es-un-sistema-web/>
- Ríos, J. M. (2020). Modelo de evaluación de metodologías de desarrollo de software web.
- Ríos, Jimmy Molina. (2020). Obtenido de Modelo de evaluación de metodologías de desarrollo de software web.
- Román, L. L. (2013). *Metodología de la programación orientada a objetos*. México: Alfa Omega.



- SDI. (2022). *Sistemas de control*. Obtenido de <https://sdindustrial.com.mx/blog/sistemas-de-control/>
- Servicenow. (s.f.). *¿Qué es un diagrama de actividad?* Obtenido de <https://www.servicenow.com/es/workflows/creator-workflows/what-is-activity-diagram.html>
- Stsepanets, A. (2024). *Ganttpro*. Obtenido de Modelo de cascada (Waterfall): qué es y cuándo conviene usarlo: <https://blog.ganttpro.com/es/metodologia-de-cascada/>
- Telefónica S.A. (s.f.). *¿Qué es la Web 3.0 y cuáles son sus características?* Obtenido de <https://www.telefonica.com/es/sala-comunicacion/blog/que-es-la-web-3-0-y-cuales-son-sus-caracteristicas/>
- THINGS. (s.f.). *Getapp*. Obtenido de <https://www.getapp.com.mx/software/2035443/things>
- Torresburriel . (22 de 11 de 2022). *Web 1.0, web 2.0 y web 3.0*. Obtenido de <https://torresburriel.com/weblog/web1-web2-web3-2/>
- Trasobares, A. H. (2003). *Los sistemas de información: Evolución y Desarrollo*.
- Venema, M. (19 de Febrero de 2024). *Nimble Humanize Work*. Obtenido de *¿Qué es la programación extrema (XP)? Son Valores, Principios y Prácticas:* <https://www.nimblework.com/agile/extreme-programming-xp/>
- World Wide Web foundation. (2008). *World Wide Web foundation*. Obtenido de *Historia de la Red:* <https://webfoundation.org/about/vision/history-of-the-web/>



ANEXOS

LEVANTAMIENTO DE REQUERIMIENTOS DE INFORMACIÓN SOBRE LOS REPORTES

ENTREVISTA

1. ¿En qué consiste las Actividades de Innovación Educativa?
R=Las Actividades de Innovación Educativa consiste en cursos de preparación para el alumno en sus materias.
2. ¿Cuál es el objetivo de los reportes de Actividades de Innovación Educativa?
R=Apoyo hacía el proceso de nivelación de los alumnos.
3. ¿Tiene que elaborar algún informe o reporte?
R=Si, se elabora un reporte para representar cada actividad de Innovación Educativa como evidencia de ejecución.
4. ¿Cuál es el proceso de entrega de los reportes de Actividades de Innovación Educativa?
R= Se registran, se cursan, se finaliza, después se elaboran los reportes de las actividades realizadas, de forman manual y se entrega la documentación físicamente en papel a la coordinación para su validación.
5. ¿Cuántas Actividades de Innovación Educativa existen?
R= Existen 5 tipos de Actividades
Actividades de Homologación
Actividades de Nivelación acompañamiento académico de preparación para presentar el examen extraordinario.
Actividades de avance académico cursos de verano para adelantar asignaturas
Actividades de avance académico evaluación de acreditación por competencias
Actividades de avance académico acompañamiento académico (preparación para presentar evaluación de acreditación por competencias).
6. ¿Qué tipo de datos se necesitan para elaborar los reportes de Actividades de Innovación Educativa?
R=En las Actividades se piden los datos de horas que se van a dar, el número de alumnos registrados, alumnos que se aprobaron y no aprobaron, las materias, las fechas de inicio y final del curso, la calificación, asesorías Individuales o Grupales
7. ¿En qué periodo se elaboran los reportes de Actividades de Innovación Educativa?
R=Los periodos se presentan dos semanas, antes del inicio de curso normal.
8. ¿Maneja o existe, algún sistema o plataforma a la entrega de sus reportes?
R=En específicos no hay ningún sistema solo se entrega en formatos digitales o manuales.

