

INSTITUTO COMERCIAL SUPERIOR DE LA NACIÓN
“Tte. ARMANDO PALACIOS”
INCOS – LA PAZ
CARRERA DE SISTEMAS INFORMÁTICOS



PROYECTO DE GRADO
SISTEMA WEB DE GESTIÓN ACADÉMICA. CASO: INSTITUTO
TECNOLÓGICO DEL OCCIDENTE

POSTULANTE: FREDDY ALEJANDRO MAGNE LOZA

TUTOR: Lic. EGBERTO TOLA FLORES

Proyecto de Grado para Optar al Título de Técnico Superior en
Sistemas Informáticos

La Paz – Bolivia

2024

DEDICATORIA

Nada de esto habría sido posible sin el apoyo constante de mi familia, la incondicionalidad de mis amigos, y todas las personas que el destino puso en mi camino a lo largo de la vida. A todos ustedes, les agradezco desde lo más profundo de mi corazón.

AGRADECIMIENTO

Agradezco, en primer lugar, a Dios por brindarme la fortaleza y sabiduría necesarias para completar esta etapa de mi formación académica. A mi familia, por su apoyo constante y motivación en cada paso del camino.

A mis compañeros, por su colaboración y camaradería a lo largo de este proceso.

A mis docentes, quienes me guiaron y compartieron sus conocimientos, permitiéndome crecer tanto personal como profesionalmente.

Finalmente, expreso mi gratitud a todas las personas que, de una u otra manera, contribuyeron al logro de este importante objetivo.

ÍNDICE

MARCO INTRODUCTORIO	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	2
1.2.1. ANTECEDENTES DE LA INSTITUCIÓN	2
1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES.....	3
1.3. PLANTEAMIENTO DEL PROBLEMA	4
1.3.1. PROBLEMA CENTRAL.....	4
1.3.2. PROBLEMAS SECUNDARIOS.....	4
1.4. OBJETIVOS	5
1.4.1. OBJETIVO GENERAL.....	5
1.4.2. OBJETIVOS ESPECÍFICOS	5
1.5. JUSTIFICACIÓN	5
1.5.1. JUSTIFICACIÓN TECNOLÓGICA	5
1.5.2. JUSTIFICACIÓN ECONÓMICA	6
1.5.3. JUSTIFICACIÓN OPERATIVA.....	6
1.6. ALCANCES Y LIMITACIONES	6
1.6.1. ALCANCES	6
1.6.2. LÍMITES.....	7
1.7. ENFOQUE METODOLÓGICO	8
1.7.1. METODOLOGÍA DE LA INVESTIGACIÓN	8
1.7.2. METODOLOGÍA DE DESARROLLO	8
MARCO TEÓRICO	9
2.1. CONCEPTOS GENERALES.....	9
2.1.1. APLICACIÓN WEB.....	9
2.1.2. INTERFAZ DE USUARIO.....	9

2.1.3.	LÓGICA DE NEGOCIO	9
2.1.4.	BASE DE DATOS.....	10
2.1.5.	AUTENTICACIÓN DE USUARIOS	10
2.1.6.	SEGURIDAD INFORMÁTICA	10
2.1.7.	DISEÑO RESPONSIVO	10
2.2.	INTEGRACIÓN DE SISTEMAS	10
2.2.1.	INTERFAZ DE APLICACIÓN DE APLICACIONES (API)	11
2.2.2.	ARQUITECTURA ORIENTADA A SERVICIOS (SOA)	11
2.3.	METODOLOGÍA DE DESARROLLO	12
2.3.1.	METODOLOGÍA ÁGIL.....	12
2.3.2.	METODOLOGÍA XP	12
2.4.	LENGUAJE UNIFICADO DE MODELADO	14
2.4.1.	DIAGRAMAS UML	14
2.4.2.	DIAGRAMA DE CASOS DE USO	14
2.4.3.	DIAGRAMA DE CLASES	15
2.4.4.	DIAGRAMA DE SECUENCIAS	15
2.4.5.	DIAGRAMA DE COMPONENTES	15
2.5.	INGENIERÍA DE SOFTWARE	16
2.5.1.	MÉTRICAS DE CALIDAD - ISO 9126.....	16
2.5.2.	PROCESO DEL SOFTWARE	17
2.5.3.	ESTRUCTURA DEL PROCESO	17
2.6.	MODELO DE DISEÑO DE BASE DE DATOS	18
2.6.1.	BASE DE DATOS.....	18
2.6.2.	ELEMENTOS DEL MODELO RELACIONAL	18
2.6.3.	GESTOR DE BASE DE DATOS	20
2.6.4.	MYSQL WORKBENCH	20

2.7.	LENGUAJES DE PROGRAMACIÓN	21
2.7.1.	PHP	21
2.7.2.	JAVASCRIPT	22
2.8.	REPORTES.....	22
2.8.1.	REPORTE ACADÉMICO DE NOTAS	23
2.9.	NORMAS Y LEYES.....	24
2.9.1.	LEY N°164.....	24
2.10.	HERRAMIENTAS DE DESARROLLO	26
2.11.	MÉTODO DE ANÁLISIS DE COSTO Y BENEFICIO.....	28
2.11.1.	COCOMO II	28
	MARCO APLICATIVO	30
3.1.	ANÁLISIS DE LA SITUACIÓN ACTUAL	30
3.2.	PLANIFICACIÓN.....	30
3.2.1.	ANÁLISIS DE TIPOS DE USUARIO	30
3.2.2.	REQUERIMIENTOS FUNCIONALES	32
3.2.3.	REQUERIMIENTOS NO FUNCIONALES.....	33
3.2.4.	DIAGRAMA DE CASO DE USO DEL SISTEMA.....	35
3.3.	APLICACIÓN DE LA METODOLOGÍA XP.....	36
3.3.1.	FASE I – PLANIFICACIÓN.....	36
3.3.2.	PLANIFICACIÓN DE ITERACIONES.....	44
3.4.	FASE II – DISEÑO	45
3.4.1.	PRIMERA ITERACIÓN.....	45
3.4.2.	SEGUNDA ITERACIÓN	52
3.4.3.	TERCERA ITERACIÓN.....	57
3.4.4.	CUARTA ITERACIÓN	59
3.5.	FASE III - CODIFICACIÓN.....	61

3.6.	FASE IV - PRUEBAS	62
3.6.1.	CRITERIOS DE ACEPTACIÓN.....	62
MARCO EVALUATIVO		67
4.1.	Métricas de Calidad.....	67
4.1.1.	Usabilidad.....	67
4.1.2.	Funcionalidad	69
4.1.3.	Confiabilidad.....	73
4.1.4.	Mantenibilidad	74
4.1.5.	Portabilidad.....	75
4.1.6.	Calidad del Proyecto	75
4.2.	Costo de Desarrollo.....	76
4.2.1.	Costo del Software Desarrollado.....	76
MARCO CONCLUSIVO		78
5.1.	CONCLUSIONES.....	78
MARCO BIBLIOGRÁFICO		80
6.1.	BIBLIOGRAFÍA	80

ÍNDICE DE FIGURAS

FIGURA 3.1 DIAGRAMA GENERAL DE CASO DE USO DEL SISTEMA	35
FIGURA 3.2 DIAGRAMA DE BASE DE DATOS DE LOS USUARIOS.....	45
FIGURA 3.4 INTERFAZ WEB DE INICIO DE SESIÓN	46
FIGURA 3.5 INTERFAZ WEB PARA REGISTRAR USUARIOS	46
FIGURA 3.6 INTERFAZ WEB PARA LISTAR LOS DIFERENTES USUARIOS	47
FIGURA 3.7 DIAGRAMA DE BASE DE DATOS PARA GESTIONES Y CARRERAS	48
FIGURA 3.8 INTERFAZ WEB PARA LA ADMINISTRACIÓN DE GESTIONES	48
FIGURA 3.9 INTERFAZ WEB PARA LA ADMINISTRACIÓN DE CARRERAS.....	49
FIGURA 3.10 DIAGRAMA DE BASE DE DATOS PARA LOS CURSOS	50
FIGURA 3.11 INTERFAZ WEB PARA EL REGISTRO DE CURSOS.....	50
FIGURA 3.12 INTERFAZ WEB PARA ADMINISTRAR LOS CURSOS	51
FIGURA 3.13 DIAGRAMA DE BASE DE DATOS PARA LA GESTIÓN DE ESTUDIANTES-CURSO	52
FIGURA 3.14 INTERFAZ WEB PARA REGISTRO DE ESTUDIANTES	52
FIGURA 3.15 INTERFAZ WEB PARA ADMINISTRAR DATOS DE LOS ESTUDIANTES	53
FIGURA 3.16 DIAGRAMA DE BASE DE DATOS PARA LA GESTIÓN DE CALIFICACIONES	54
FIGURA 3.17 INTERFAZ WEB PARA LA GESTIÓN DE CALIFICACIONES.....	55
FIGURA 3.18 INTERFAZ WEB PARA LA VISUALIZACIÓN DE NOTAS DE PARTE DEL ESTUDIANTE....	56
FIGURA 3.19 DISEÑO DE BASE DE DATOS PARA EL REGISTRO DE DOCENTES	57
FIGURA 3.20 INTERFAZ WEB PARA EL REGISTRO DE DOCENTES	58
FIGURA 3.21 INTERFAZ WEB PARA LA ADMINISTRACIÓN DE DOCENTES.....	59
FIGURA 3.22 DIFERENTES GRÁFICAS ADMINISTRATIVAS	60
FIGURA 3.23 DIAGRAMA GENERAL DE BASE DE DATOS.....	61

ÍNDICE DE TABLAS

TABLA 2.1	TIPOS DE API	11
TABLA 2.2	ROLES DE XP	13
TABLA 2.4	EVENTOS DE XP	13
TABLA 2.5	CARACTERÍSTICAS DE LAS MÉTRICAS DE CALIDAD	16
TABLA 2.6	OBJETIVOS DE LA ESTRUCTURA DEL PROCESO.....	17
TABLA 2.7	ELEMENTOS DEL MODELO RELACIONAL.....	19
TABLA 2.8	CARACTERÍSTICAS DE UNA DBMS.....	20
TABLA 2.9	HERRAMIENTAS DE DESARROLLO PARA EL PROYECTO.....	26
TABLA 3.1	IDENTIFICACIÓN DE ROLES Y TAREAS	30
TABLA 3.1	REQUERIMIENTOS FUNCIONALES DEL SISTEMA.....	32
TABLA 3.2	REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA	33
TABLA 3.3	HISTORIAS DE USUARIO DEL PROYECTO.....	36
TABLA 3.4	HISTORIA DE USUARIO 1: HU-01	37
TABLA 3.5	TAREAS A REALIZAR: HU-01	37
TABLA 3.6	HISTORIA DE USUARIO 2: HU-02	38
TABLA 3.7	TAREAS A REALIZAR: HU-02	38
TABLA 3.8	HISTORIA DE USUARIO 3: HU-03	39
TABLA 3.9	TAREAS A REALIZAR: HU-03	39
TABLA 3.10	HISTORIA DE USUARIO 4: HU-04	40
TABLA 3.11	TAREAS A REALIZAR: HU-04	40
TABLA 3.12	HISTORIA DE USUARIO 5: HU-05	41
TABLA 3.13	TAREAS A REALIZAR: HU-05	41
TABLA 3.14	HISTORIA DE USUARIO 6: HU-06	42
TABLA 3.15	TAREAS A REALIZAR: HU-06	42
TABLA 3.16	HISTORIA DE USUARIO 7: HU-07	43
TABLA 3.17	TAREAS A REALIZAR: HU-07	43
TABLA 3.18	PLANIFICACIÓN DE ITERACIONES DEL PROYECTO.....	44
TABLA 3.19	CRITERIO DE ACEPTACIÓN: HU-01	62
TABLA 3.20	CRITERIO DE ACEPTACIÓN: HU-02.....	62
TABLA 3.21	CRITERIO DE ACEPTACIÓN: HU-03.....	63
TABLA 3.22	CRITERIO DE ACEPTACIÓN: HU-04.....	64

TABLA 3.23	CRITERIO DE ACEPTACIÓN: HU-05	64
TABLA 3.24	CRITERIO DE ACEPTACIÓN: HU-06	65
TABLA 3.25	CRITERIO DE ACEPTACIÓN: HU-07	66
TABLA 4.1	ENCUESTA REALIZADA A DIFERENTES USUARIOS	68
TABLA 4.2	CONTEO TOTAL PARA PUNTO FUNCIÓN	70
TABLA 4.3	VALORES DE AJUSTE DE COMPLEJIDAD	71
TABLA 4.4	CALIDAD DEL PROYECTO	75

RESUMEN

Actualmente, los sistemas de gestión académica son componentes esenciales para el éxito y la eficiencia de las instituciones educativas. La evolución tecnológica ha impulsado la demanda de herramientas que optimicen los procesos administrativos y académicos, mejorando la experiencia tanto del personal como de los estudiantes. En Bolivia, diversas instituciones utilizan sistemas privados para facilitar estas tareas. El Instituto Tecnológico del Occidente, que ofrece cursos de mecánica, electrónica e informática, enfrenta el desafío de optimizar su gestión mediante un sistema adaptado a sus necesidades.

En la actualidad, se emplean métodos manuales y múltiples programas que dificultan la integración de información, generando retrasos y errores en la gestión. Este proyecto tiene como objetivo desarrollar un sistema web que centralice todas las funciones clave, permitiendo una administración más eficiente y mejorando la experiencia de usuarios como docentes, estudiantes y administradores.

Para abordar esta problemática, se adoptará la metodología ágil XP, reconocida por su enfoque iterativo que promueve la flexibilidad y la colaboración. Se utilizarán tecnologías modernas como PHP y MySQL, garantizando un sistema robusto y flexible, con una interfaz intuitiva. Esta combinación asegurará que el sistema no solo resuelva los desafíos actuales, sino que también facilite el crecimiento futuro del Instituto.

1. PRIMER CAPÍTULO

MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

Actualmente, todo sistema de gestión académica se ha convertido en un componente vital para el éxito y la eficiencia de las instituciones educativas. La evolución tecnológica ha generado una demanda creciente de sistemas que realicen procesos administrativos y académicos, permitiendo un flujo de trabajo más ágil y una mejor experiencia tanto para el personal como para los estudiantes. Por ende, las instituciones y universidades de Bolivia cuentan con diferentes sistemas privados que facilitan tales procesos académicos. Considerando la importancia de este aspecto, el Instituto Tecnológico del Occidente, reconocido por ofrecer diversos cursos cortos de aprendizaje sobre mecánica, electrónica e informática, se enfrenta al desafío de optimizar su gestión mediante la implementación de un sistema administrativo robusto y adaptado a sus necesidades particulares.

Actualmente, en el instituto se utilizan diversos métodos manuales y una variedad de programas para gestionar las actividades académicas y administrativas. Esta diversidad de herramientas dificulta la integración de la información y los procesos, lo que genera retrasos y errores en la gestión. Por lo tanto, el objetivo principal de este proyecto es desarrollar un sistema de gestión académica que integre todas las funciones esenciales, permitiendo una administración más eficiente y mejorando la experiencia de los usuarios involucrados sean el administrador, plantel docente y estudiantes.

Para enfrentar esta problemática, se implementará la metodología ágil XP, reconocida por su enfoque en la mejora continua, la colaboración constante con el cliente y la entrega frecuente de software funcional a través de iteraciones cortas y ajustables. Por otro lado, en cuanto a la implementación técnica, se utilizarán tecnologías modernas como *PHP* y *MySQL* para el desarrollo del *backend*, aprovechando su robustez y flexibilidad para la gestión de datos y lógica de negocio, además del diseño de la interfaz de usuario, garantizando una experiencia intuitiva y atractiva para los usuarios finales. La combinación de estas tecnologías y metodologías asegurará una implementación del proyecto innovadora y eficiente.

1.2. ANTECEDENTES

1.2.1. ANTECEDENTES DE LA INSTITUCIÓN

El Instituto Tecnológico del Occidente (ITO) surge como una respuesta innovadora y visionaria a las demandas de formación profesional en la región. Fundado en el año 2022, se ha destacado por ser pionero en la provisión de cursos cortos orientados a diversas áreas del conocimiento, principalmente en las áreas de Mecánica, Electrónica, Informática y Gastronomía. Su compromiso con la excelencia académica y su enfoque en la capacitación práctica han posicionado al instituto como un referente de calidad en el ámbito educativo regional.

❖ MISIÓN

“Nuestra misión es proporcionar una educación técnica de alta calidad, centrada en el estudiante y orientada hacia el futuro. A través de programas académicos innovadores y actualizados, así como de prácticas de aprendizaje práctico, nos comprometemos a preparar a nuestros estudiantes para enfrentar los desafíos del mundo tecnológico actual y futuro. Nos esforzamos por fomentar un ambiente de aprendizaje inclusivo, colaborativo y orientado a resultados, donde la excelencia académica y el desarrollo personal se promuevan en igual medida.”

❖ VISIÓN

Ser la institución tecnológica líder en la formación de profesionales altamente capacitados y comprometidos con la innovación y el desarrollo tecnológico. Buscamos inspirar y capacitar a nuestros estudiantes para que se conviertan en líderes en sus respectivas áreas, contribuyendo así al progreso de la sociedad y al avance de la industria a nivel nacional e internacional.

❖ UBICACIÓN

El instituto se encuentra ubicada en la ciudad de El Alto – Bolivia, específicamente en la Zona Bolívar “A”, Calle Evadidos del Paraguay Nro. 77. (Ver Anexos)

1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES

TÍTULO: SISTEMA ACADÉMICO PARA EL REGISTRO Y SEGUIMIENTO ACADÉMICO. CASO: ACADEMIA DE FORMACIÓN PROFESIONAL “BELA”

AUTOR: JHANNET MONICA CALAMANI SALAS

INSTITUCIÓN: UNIVERSIDAD PÚBLICA DE EL ALTO

AÑO: 2020

Resumen. - El sistema académico se desarrolló utilizando tecnologías adecuadas basadas en servicios web para su interacción futura con otros proyectos de índole educacional. Para el análisis y diseño, se aplicó la metodología UWE¹, una herramienta para modelar aplicaciones web. Simultáneamente, para el desarrollo del software² se utilizaron PHP³ y MySQL⁴, lo que permitió crear interfaces sencillas de comprender y manipular para los usuarios. Considerando que la Academia de Formación Profesional “Bela”, institución dedicada a la formación de profesionales en estilismo y estética facial, se beneficia en gran medida con la implementación del software al cubrir aspectos de registro y seguimiento académico de sus estudiantes.

TÍTULO: SISTEMA WEB DE GESTIÓN ACADÉMICA. CASO: UNIDAD EDUCATIVA “INICUA”

AUTOR: CONSUELO ASPE ALANOCA

INSTITUCIÓN: UNIVERSIDAD PÚBLICA DE EL ALTO

AÑO: 2020

¹ UWE (UML-based Web Engineering) es una metodología para diseñar y desarrollar aplicaciones web utilizando diagramas UML.

² *Software* hace referencia a un conjunto programas y aplicaciones informáticas que se utilizan en un sistema informático.

³ PHP es un lenguaje de programación interpretado del lado del servidor y de uso general que se adapta al desarrollo web.

⁴ MySQL es un sistema de administración de bases de datos relacionales que puede ser utilizado para proyectos web.

Resumen. - Se realizó un sistema web de gestión académica para la unidad educativa “Inicua”, empleando la metodología *OOHDM*⁵, la cual propone el diseño de aplicaciones hipermedia y web a través de un proceso de cinco fases: obtención de requerimientos, diseño conceptual, diseño navegacional, diseño abstracto de la interfaz y su implementación. El desarrollo del proyecto se lleva a cabo utilizando *PHP* como lenguaje de programación y *MariaDB* como gestor de base de datos para almacenar toda la información. Además, se hace uso del *Framework CodeIgniter* y *Bootstrap* para adaptar la interfaz del sistema. El proyecto garantiza el procesamiento integral y confiable de la información, incorporando mecanismos de seguridad para prevenir sustracciones, modificaciones o falsificaciones por parte de personas no autorizadas.

1.3. PLANTEAMIENTO DEL PROBLEMA

1.3.1. PROBLEMA CENTRAL

¿De qué manera se puede optimizar la gestión académica del Instituto Tecnológico del Occidente?

1.3.2. PROBLEMAS SECUNDARIOS

- ❖ El registro manuscrito de calificaciones por parte del personal educativo dificulta la organización y la rápida disponibilidad de la información académica.
- ❖ El riesgo asociado al registro de datos personales de estudiantes y docentes mediante Excel compromete la privacidad y seguridad, lo que puede provocar filtraciones de información confidencial.
- ❖ La dificultad en el proceso de registro de nuevos estudiantes en la institución, según la carrera elegida, provoca desorganización en los registros académicos y administrativos.
- ❖ Tanto estudiantes como docentes no tienen la opción de acceder y consultar las calificaciones a través de una cuenta personal.

⁵ *OOHDM* (Object-Oriented Hypermedia Design Method) es una metodología para diseñar aplicaciones web y sistemas de hipermedia utilizando principios de la programación orientada a objetos

- ❖ El uso de Excel para la generación de reportes académicos consume una cantidad significativa de tiempo y recursos administrativos, lo que reduce la eficiencia en la producción de informes.

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar un sistema web de gestión académica para el Instituto Tecnológico del Occidente que proporcione una administración eficiente de los datos académicos de estudiantes como del personal administrativo.

1.4.2. OBJETIVOS ESPECÍFICOS

- ❖ Integrar un módulo web que permita al personal administrativo gestionar las calificaciones y rendimiento académico de los estudiantes.
- ❖ Establecer una base de datos relacional que almacene de manera segura la información de cada estudiante, docente y personal administrativo, para la organización y seguimiento de las calificaciones y cursos.
- ❖ Diseñar un módulo de registro de nuevos estudiantes de acuerdo a sus datos personales y carrera elegida.
- ❖ Crear una plataforma de acceso personal para que los estudiantes y docentes puedan consultar las calificaciones a través de su cuenta de usuario.
- ❖ Generar reportes sobre el rendimiento académico y administrativo para facilitar el seguimiento de los estudiantes.

1.5. JUSTIFICACIÓN

1.5.1. JUSTIFICACIÓN TECNOLÓGICA

El instituto tecnológico cuenta con la infraestructura suficiente en el área de computación, que incluye equipos necesarios para la implementación del sistema web de administración de calificaciones. La elección de utilizar software de código abierto para el desarrollo del software de información asegura una solución económica y accesible, sin la necesidad de realizar inversiones adicionales significativas en licencias o

tecnología propietaria. Además, dado que el instituto cuenta con el hardware necesario para soportar todas las fases del proyecto, se asegura una implementación sin contratiempos y se optimizan los recursos disponibles para cumplir de manera efectiva con los objetivos del proyecto.

1.5.2. JUSTIFICACIÓN ECONÓMICA

El desarrollo del sistema académico reducirá los costos operativos asociados con los procesos manuales. La digitalización de la generación y distribución de boletines y otros documentos académicos disminuirá el consumo de papel y otros recursos físicos, resultando en ahorros sustanciales a largo plazo. Adicionalmente, la eficiencia administrativa mejorada permitirá una mejor asignación de recursos humanos y financieros, optimizando el presupuesto institucional. La inversión inicial en tecnología y capacitación será recuperada rápidamente gracias a los ahorros en tiempo y recursos, y al incremento en la eficiencia operativa.

1.5.3. JUSTIFICACIÓN OPERATIVA

Al proporcionar un acceso sencillo y rápido a la información académica y personal, se incrementará la transparencia y se reducirá el estrés asociado con la obtención de datos importantes. La centralización de la información promoverá una comunicación más efectiva entre todos los actores del proceso educativo, mejorando la colaboración y el apoyo mutuo. Además, la automatización de procesos administrativos liberará tiempo para que el personal se enfoque en actividades más significativas, como el apoyo directo a los estudiantes, contribuyendo al desarrollo integral de la comunidad educativa.

1.6. ALCANCES Y LIMITACIONES

1.6.1. ALCANCES

- ❖ La plataforma web contará con una interfaz de usuario intuitiva y sencilla de comprender.
- ❖ Se automatizarán procesos como la generación de boletines y entre otros documentos académicos.
- ❖ El sistema podrá ser utilizado tanto vía internet como en un área de red local.

- ❖ Se implementarán sistemas de seguridad para proteger la información personal de estudiantes y docentes.
- ❖ Los administradores podrán acceder a estadísticas y análisis sobre el rendimiento académico de los estudiantes, facilitando la toma de decisiones, a través de herramientas de visualización de datos, como gráficos y tablas interactivas, que mejorarán el análisis de la información.
- ❖ Los usuarios podrán acceder a sus cuentas personales para visualizar sus calificaciones, historial académico y otra información relevante, y podrán descargar estos datos en formato PDF.
- ❖ El sistema será desarrollado con un diseño responsive, permitiendo su uso en diferentes dispositivos como celulares, tablets y computadoras.

1.6.2. LÍMITES

- ❖ Solo se almacenará información personal relevante para fines académicos, evitando guardar datos no prioritarios para la institución.
- ❖ No se generarán reportes de asistencia de los estudiantes.
- ❖ La plataforma no ofrecerá recursos educativos ni materiales de estudio físicos, como aulas, equipos, materiales didácticos, libros impresos o cualquier otro recurso físico necesario para la enseñanza presencial
- ❖ El sistema solo gestionará tres tipos de usuarios sea el administrador, docente y el estudiante, sin incluir otros roles.
- ❖ El sistema no gestionará aspectos relacionados con pagos o costos de las carreras, ni abarcará ningún servicio de carácter económico.
- ❖ La evaluación del personal docente no será parte del sistema.
- ❖ El sistema no contará con un módulo de mensajería para la comunicación directa entre estudiantes y docentes.
- ❖ No se incluirá la planificación o programación de cursos y asignaturas dentro del sistema.

- ❖ No se incluirán funciones para el seguimiento o gestión de estudiantes egresados.
- ❖ La implementación del sistema web estará limitada al Instituto Tecnológico del Occidente y sus necesidades lo cual no se extenderá a otras instituciones educativas.

1.7. ENFOQUE METODOLÓGICO

1.7.1. METODOLOGÍA DE LA INVESTIGACIÓN

Para este proyecto se empleará el *método descriptivo* como enfoque principal en el desarrollo del sistema web diseñado para administrar las calificaciones del instituto. A través de esta metodología se proporcionará una descripción detallada y precisa de las diversas partes del sistema, utilizando tanto datos cuantitativos como cualitativos. Se recopilarán datos mediante técnicas como encuestas, cuestionarios, entrevistas, observación directa y análisis de documentos, para identificar las características y comportamientos actuales de los estudiantes, docentes y personal administrativo en la gestión de calificaciones del instituto. El análisis de estos datos incluirá análisis estadístico para los datos cuantitativos y análisis temático para los datos cualitativos. Los resultados se presentarán en forma de tablas, gráficos y narrativas detalladas, facilitando una comprensión exhaustiva de las necesidades específicas de los usuarios. Por último, este enfoque garantizará una gestión eficiente y efectiva de las calificaciones académicas y asegurará que las soluciones desarrolladas respondan directamente a las expectativas y requerimientos de los usuarios involucrados.

1.7.2. METODOLOGÍA DE DESARROLLO

Para abordar este proyecto, se utilizará la metodología ágil Extreme Programming (XP) como enfoque principal de desarrollo. Esta metodología es un marco de trabajo altamente efectivo que se centra en la mejora continua, la adaptación rápida a los cambios y la entrega frecuente de versiones funcionales del software. El proyecto se dividirá en iteraciones cortas y frecuentes, donde se trabajará en el diseño, desarrollo y prueba de pequeñas funcionalidades. Al final de cada iteración, se realizará una revisión para demostrar los avances, evaluar el producto y ajustar el plan según la retroalimentación recibida.

2. SEGUNDO CAPÍTULO

MARCO TEÓRICO

En este capítulo se describe las metodologías, técnicas y herramientas empleadas en el desarrollo del sistema web del proyecto. Por ende, se busca proporcionar una visión clara y completa del proyecto, considerando el enfoque estratégico adoptado, las técnicas específicas aplicadas y la justificación de las herramientas tecnológicas utilizadas.

2.1. CONCEPTOS GENERALES

2.1.1. APLICACIÓN WEB

Un sistema web o también llamado aplicación web, es un conjunto de programas y servicios que se ejecutan en un servidor y se acceden a través de un navegador web. Este tipo de sistema está diseñado para realizar una amplia variedad de tareas, como la gestión de información, la comunicación, el comercio electrónico y la educación.

Según Berners-Lee y Fischetti (2019), un sistema web puede definirse como una colección de software que interactúa mediante protocolos web para proporcionar servicios a los usuarios a través de la red. Esta definición enfatiza la naturaleza distribuida y accesible de los sistemas web, que permiten a los usuarios interactuar con la aplicación desde cualquier lugar con una conexión a internet.

2.1.2. INTERFAZ DE USUARIO

La interfaz de usuario es la parte del sistema con la que interactúan los usuarios finales. Esta interfaz se desarrolla utilizando tecnologías como HTML, CSS y JavaScript, lo que permite crear páginas web interactivas y dinámicas.

2.1.3. LÓGICA DE NEGOCIO

Es la capa que contiene las reglas y procesos específicos del sistema, gestionando operaciones como la validación de datos, el procesamiento de transacciones y la aplicación de reglas de negocio. Esta capa se implementa generalmente en lenguajes de programación del lado del servidor, como PHP, Python, Ruby o Java.

2.1.4. BASE DE DATOS

La base de datos es el componente encargado de almacenar y gestionar la información que utiliza el sistema. Los sistemas de gestión de bases de datos relacionales (RDBMS) como MySQL, PostgreSQL y SQL Server son comúnmente utilizados para este propósito (Sommerville, 2020).

2.1.5. AUTENTICACIÓN DE USUARIOS

La autenticación es el proceso de verificar la identidad de los usuarios que intentan acceder al sistema, mientras que la autorización determina qué acciones pueden realizar dentro del sistema. Estos mecanismos son fundamentales para garantizar que solo los usuarios autorizados puedan acceder a ciertas funcionalidades y datos sensibles (Smith, 2020).

2.1.6. SEGURIDAD INFORMÁTICA

La seguridad informática se refiere a las medidas y controles implementados para proteger la información y los sistemas del acceso no autorizado, el uso indebido, la divulgación, la interrupción, la modificación o la destrucción. En el contexto de un sistema académico, es crucial garantizar la confidencialidad, integridad y disponibilidad de los datos de calificaciones y de los usuarios (Sotomayor, 2018).

2.1.7. DISEÑO RESPONSIVO

El diseño responsivo se refiere a la técnica de desarrollo web que permite que una aplicación web se adapte de manera óptima a diferentes tamaños de pantalla y dispositivos, como computadoras, tabletas y smartphones. Es especialmente importante para garantizar una buena experiencia de usuario en un sistema académico accesible desde múltiples dispositivos (Marcotte, 2017).

2.2. INTEGRACIÓN DE SISTEMAS

La integración de sistemas es un proceso fundamental en el ámbito de la tecnología de la información que permite que diferentes sistemas y aplicaciones trabajen juntos de manera cohesiva. Este proceso es crucial para mejorar la eficiencia operativa, la calidad de los datos y la toma de decisiones en las organizaciones.

2.2.1. INTERFAZ DE APLICACIÓN DE APLICACIONES (API)

Las API⁶s permiten que diferentes aplicaciones se comuniquen entre sí mediante la definición de un conjunto de reglas y protocolos. Además, son extremadamente flexibles y pueden ser utilizadas para integrar una amplia variedad de sistemas, desde aplicaciones web hasta servicios en la nube. Estas interfaces permiten a las empresas expandir la funcionalidad de sus aplicaciones existentes sin tener que realizar cambios significativos en el código base (Medjaoui et al., 2020).

Tabla 2.1

Tipos de API

API	DESCRIPCIÓN
API REST	Permite la comunicación entre sistemas mediante el uso de HTTP, siguiendo principios de arquitectura REST.
API SOAP	Utiliza el protocolo SOAP para intercambiar mensajes XML estructurados entre sistemas.
API RPC	Facilita la ejecución de procedimientos remotos a través de una red, permitiendo que un programa ejecute un procedimiento en otro espacio de direcciones.

Nota. Descripción de los tipos de API

2.2.2. ARQUITECTURA ORIENTADA A SERVICIOS (SOA)

Esta metodología permite que los componentes de software se expongan como servicios, que pueden ser descubiertos y utilizados por otras aplicaciones a través de la red. Esta arquitectura promueve la reutilización y la flexibilidad, ya que los servicios pueden ser combinados y orquestados para crear nuevas aplicaciones y procesos de negocio. Adicionalmente, es particularmente beneficiosa para organizaciones que buscan una alta escalabilidad y adaptabilidad en sus sistemas de TI (Papazoglou, 2019).

⁶ API se refiere a una pieza de código que permite a diferentes aplicaciones comunicarse entre sí y compartir información y funcionalidades.

2.3. METODOLOGÍA DE DESARROLLO

La metodología de desarrollo se refiere al marco estructurado que se utiliza para planificar, controlar y ejecutar un proyecto de desarrollo de software. Estas metodologías ayudan a los equipos a organizar su trabajo, gestionar recursos y garantizar la entrega de productos de calidad. Existen diversas metodologías, cada una con sus propias características y enfoques.

2.3.1. METODOLOGÍA ÁGIL

La metodología ágil es un enfoque iterativo e incremental para el desarrollo de software, diseñado para ser más flexible y adaptable que los métodos tradicionales. Se enfoca en la colaboración, la retroalimentación continua y la entrega de software funcional en cortos períodos de tiempo. Este proceso permite a los equipos responder rápidamente a los cambios en los requisitos y a las prioridades del cliente.

El Manifiesto Ágil publicado en 2001, establece los principios fundamentales de este enfoque, enfatizando la importancia de individuos e interacciones sobre procesos y herramientas, software funcionando sobre documentación extensiva, colaboración con el cliente sobre la negociación de contratos y respuesta al cambio sobre seguir un plan (Beck et al., 2001).

2.3.2. METODOLOGÍA XP

Extreme Programming (XP) es una metodología ágil que busca mejorar la calidad del software y la capacidad de respuesta a los cambios mediante prácticas de desarrollo intensivas y colaborativas. XP se enfoca en la entrega continua de software funcional, la mejora de la calidad del código y la adaptación a las necesidades del cliente a través de ciclos cortos de desarrollo y retroalimentación constante. A continuación, se presentan las tablas con definiciones explícitas sobre los principales componentes de XP.

Tabla 2.2*Roles de XP*

ROL	DESCRIPCIÓN
CLIENTE	Proporciona las historias de usuario y define las prioridades del producto. Participa activamente en el desarrollo para asegurar que el producto final cumpla con sus expectativas.
DESARROLLADOR	Encargado de escribir el código y realizar las pruebas necesarias. Trabaja en parejas y sigue las prácticas de XP como el desarrollo guiado por pruebas (TDD) y la integración continua.
COACH	Facilita la adopción de las prácticas de XP dentro del equipo, ayuda a resolver problemas y asegura que el equipo siga las prácticas de programación extrema.
TRACKER	Monitorea el progreso del proyecto, mide el rendimiento y asegura que el equipo esté alcanzando los objetivos establecidos.

Nota. Descripción de los roles de *Extreme Programming* (XP)

Tabla 2.3*Eventos de XP*

EVENTOS	DESCRIPCIÓN
PLANIFICACIÓN DE ITERACIÓN	Reunión al inicio de cada iteración donde el equipo define qué historias de usuario se implementarán y cómo se organizarán las tareas.
REUNIONES DIARIAS	Breves reuniones diarias donde el equipo revisa el progreso, identifican obstáculos y ajusta el plan según sea necesario.

DEMOSTRACIÓN DE ITERACIÓN	Al final de cada iteración, el equipo presenta el incremento de producto al cliente y recibe retroalimentación.
RETROSPECTIVA	Reunión al final de cada iteración donde el equipo reflexiona sobre el proceso, identifica áreas de mejora y planifica ajustes para la próxima iteración.

Nota. Descripción de los eventos de *Extreme Programming* para utilizarlos en las iteraciones

2.4. LENGUAJE UNIFICADO DE MODELADO

El Lenguaje Unificado de Modelado (UML) es una notación estándar utilizada en el desarrollo de sistemas de software para especificar, visualizar, construir y documentar los artefactos de un sistema. UML permite a los diseñadores y desarrolladores crear modelos visuales que representan diversos aspectos del sistema, facilitando la comunicación y la comprensión entre los miembros del equipo de desarrollo (Fowler, 2004).

2.4.1. DIAGRAMAS UML

Los diagramas UML son herramientas visuales esenciales que ayudan a describir diferentes aspectos del sistema en desarrollo. Cada tipo de diagrama ofrece una perspectiva única del sistema, permitiendo a los desarrolladores y otras partes interesadas entender mejor el funcionamiento y la estructura del sistema.

2.4.2. DIAGRAMA DE CASOS DE USO

Los diagramas de casos de uso se utilizan para capturar los requisitos funcionales del sistema desde la perspectiva del usuario. Representan los casos de uso, que son escenarios que describen cómo los usuarios interactúan con el sistema para alcanzar un objetivo específico.

Este tipo de diagrama es fundamental para identificar las funcionalidades que el sistema debe proporcionar y para definir las expectativas de los usuarios finales. Los diagramas de casos de uso también ayudan a documentar las interacciones entre los actores (usuarios u otros sistemas) y el sistema en cuestión (Jacobson, Booch, & Rumbaugh, 1999).

2.4.3. DIAGRAMA DE CLASES

Los diagramas de clases ofrecen una visión estática del sistema, representando las clases que forman el sistema, junto con sus atributos, métodos y las relaciones entre ellas. Estos diagramas son esenciales para el diseño de la arquitectura del sistema y la implementación del código, ya que permiten modelar la estructura de datos y las relaciones entre las entidades del sistema (Booch, Rumbaugh, & Jacobson, 2005).

2.4.4. DIAGRAMA DE SECUENCIAS

Los diagramas de secuencias se enfocan en el aspecto dinámico del sistema, mostrando cómo los objetos interactúan a lo largo del tiempo para realizar un caso de uso específico. Estos diagramas destacan el flujo de mensajes entre los objetos y el orden en que se producen las interacciones, proporcionando una visión clara del proceso de ejecución de las funcionalidades del sistema. Son particularmente útiles para comprender el comportamiento del sistema en tiempo real y para identificar posibles problemas en la comunicación entre los componentes (Fowler & Scott, 1997).

2.4.5. DIAGRAMA DE COMPONENTES

El diagrama de componentes es una herramienta esencial en la ingeniería de software, utilizada para modelar la estructura de alto nivel de un sistema mostrando cómo los distintos módulos de software, o componentes, se relacionan entre sí a través de interfaces definidas. Esta herramienta permite a los desarrolladores visualizar la arquitectura del sistema de manera modular, facilitando la identificación de dependencias y la planificación para la reutilización de componentes. Los componentes, representados como rectángulos, encapsulan funcionalidad específica y se conectan mediante interfaces, indicadas por círculos, que definen los puntos de interacción entre ellos.

Este tipo de diagramas es especialmente útil en sistemas complejos, ya que proporciona una visión clara de cómo los diferentes elementos del sistema trabajan en conjunto para cumplir con los requisitos del software (Booch, 2007).

2.5. INGENIERÍA DE SOFTWARE

La ingeniería de software es una disciplina que se encarga de todos los aspectos relacionados con la producción de software, desde la concepción inicial del sistema hasta su mantenimiento continuo. Este campo abarca un amplio conjunto de actividades, incluyendo el análisis de requisitos, diseño, codificación, pruebas y gestión de proyectos. El objetivo principal es desarrollar software de alta calidad que cumpla con las necesidades del usuario y sea fiable, eficiente y mantenible (Pressman & Maxim, 2018).

2.5.1. MÉTRICAS DE CALIDAD - ISO 9126

De acuerdo al estándar ISO⁷ 9126, las métricas de calidad del software son herramientas cruciales para evaluar y mejorar la calidad del software durante su desarrollo. Estas métricas proporcionan una manera cuantitativa de medir diferentes aspectos del software, como su eficiencia, mantenibilidad, fiabilidad y usabilidad. Entre las métricas más comunes se encuentran:

Tabla 2.4

Características de las Métricas de Calidad

CONCEPTO	DESCRIPCIÓN
FUNCIONALIDAD	Evalúa si el software cumple con los requisitos especificados y proporciona las funcionalidades necesarias.
CONFIABILIDAD	Mide la capacidad del software para funcionar correctamente bajo condiciones específicas durante un periodo determinado.
USABILIDAD	Determina la facilidad con la que los usuarios pueden aprender a utilizar el software y su eficiencia durante el uso.
EFICIENCIA	Evalúa el rendimiento del software en términos de consumo de recursos, como tiempo de procesamiento y memoria.

⁷ ISO hace referencia a la Organización Internacional de Normalización, el cual es una organización para la creación de estándares internacionales compuesta por diversas organizaciones nacionales de normalización.

MANTENIBILIDAD	Mide la facilidad con la que el software puede ser modificado para corregir errores, mejorar el rendimiento o adaptarse a un entorno cambiante.
-----------------------	---

Nota. Descripción de las métricas de calidad, las cuales tienen diferentes fórmulas para ser desarrollados. Estas ecuaciones se analizarán en el marco evaluativo.

2.5.2. PROCESO DEL SOFTWARE

El proceso del software es un conjunto de actividades, métodos, prácticas y transformaciones que se utilizan para desarrollar y mantener el software. Este proceso se puede modelar a través de diferentes enfoques, siendo algunos de los más conocidos el modelo en cascada, el modelo en espiral, el desarrollo ágil y el modelo de integración continua (Pressman & Maxim, 2018).

2.5.3. ESTRUCTURA DEL PROCESO

La estructura del proceso del software se organiza generalmente en varias fases o etapas, cada una con objetivos y actividades específicas:

Tabla 2.5

Objetivos de la Estructura del Proceso

CONCEPTO	DESCRIPCIÓN
RECOPILACIÓN DE REQUISITOS	Esta fase implica entender y documentar lo que los usuarios necesitan del software.
DISEÑO	En esta etapa, se crea una arquitectura para el software, definiendo cómo se organizarán y estructurarán sus componentes.
IMPLEMENTACIÓN	Consiste en la codificación del software, transformando el diseño en un sistema funcional.

PRUEBAS	Se realiza una serie de pruebas para identificar y corregir errores, asegurando que el software cumple con los requisitos especificados.
DESPLIEGUE	El software se pone en producción y se entrega a los usuarios finales.
MANTENIMIENTO	Incluye todas las actividades necesarias para corregir errores, mejorar el software y adaptarlo a nuevos requisitos o entornos (Sommerville, 2019).

Nota. Descripción de todas las fases del proceso estructural de un software

2.6. MODELO DE DISEÑO DE BASE DE DATOS

El diseño de base de datos es una fase crítica en el desarrollo de sistemas de información, encargada de estructurar y organizar los datos de manera eficiente. Un modelo de diseño de base de datos proporciona una representación conceptual de cómo se almacenarán, gestionarán y accederán los datos en un sistema. Este proceso implica la creación de un esquema que define la estructura de la base de datos y cómo interactúan los diferentes elementos dentro de ella (Elmasri & Navathe, 2015).

2.6.1. BASE DE DATOS

Una base de datos es un conjunto organizado de datos almacenados y gestionados electrónicamente en un sistema informático. Las bases de datos permiten la recuperación rápida y eficiente de información y están diseñadas para soportar operaciones de consulta, actualización y administración de datos. Las bases de datos pueden ser de diferentes tipos, como bases de datos relacionales, orientadas a objetos o NoSQL, cada una con sus propias características y ventajas (Date, 2004).

2.6.2. ELEMENTOS DEL MODELO RELACIONAL

El modelo relacional es un enfoque para organizar datos en forma de tablas que se relacionan entre sí mediante claves. Los elementos fundamentales del modelo relacional incluyen:

Tabla 2.6*Elementos del Modelo Relacional*

CONCEPTO	DESCRIPCIÓN
TABLAS	Las tablas son estructuras fundamentales en el modelo relacional que almacenan datos en filas y columnas. Cada tabla representa una entidad del mundo real, como empleados o productos. Las tablas permiten la organización de datos en un formato estructurado y fácil de consultar.
FILAS (TUPLAS)	Cada fila en una tabla representa una instancia individual de la entidad que la tabla describe. Por ejemplo, en una tabla de clientes, cada fila representaría un cliente específico.
COLUMNAS	Las columnas de una tabla representan los atributos de la entidad. Cada columna tiene un nombre y un tipo de dato asociado, que define el tipo de información que puede almacenarse en ella (por ejemplo, nombre, dirección, fecha de nacimiento).
CLAVES PRIMARIAS	Una clave primaria es un atributo o un conjunto de atributos que identifican de manera única cada fila en una tabla. Las claves primarias son cruciales para garantizar la unicidad de los registros y para establecer relaciones entre tablas.
CLAVES FORÁNEAS	Las claves foráneas son atributos en una tabla que se utilizan para establecer una relación con la clave primaria de otra tabla. Esto permite la creación de vínculos entre diferentes tablas y facilita la integridad referencial en la base de datos (Codd, 1970).

Nota. Descripción de los elementos del modelo relacional para organizar los datos del proyecto

2.6.3. GESTOR DE BASE DE DATOS

Un Gestor de Base de Datos (DBMS) es un software especializado que permite la creación, gestión y manipulación de bases de datos. Los DBMS proporcionan un entorno controlado para definir estructuras de datos, realizar operaciones de consulta y actualización, y mantener la integridad y seguridad de los datos. Entre las principales funciones se encuentran:

Tabla 2.7

Características de una DBMS

CONCEPTO	DESCRIPCIÓN
DEFINICIÓN DE UNA BASE DE DATOS	Permite la creación y modificación de esquemas de base de datos, incluyendo tablas, índices y restricciones.
MANIPULACIÓN DE DATOS	Facilita la inserción, actualización, eliminación y consulta de datos a través de lenguajes de consulta como SQL.
ADMINISTRACIÓN DE TRANSACCIONES	Gestiona transacciones para asegurar que las operaciones sobre la base de datos sean atómicas, consistentes, aisladas y duraderas (propiedades ACID).
SEGURIDAD Y CONTROL DE ACCESO	Proporciona mecanismos para proteger los datos contra accesos no autorizados y para definir permisos y roles de usuario.

Nota. Descripción de todas las fases del proceso estructural de un software

2.6.4. MYSQL WORKBENCH

MySQL Workbench es una herramienta visual unificada que permite diseñar, modelar, desarrollar, y administrar bases de datos MySQL. Ofrece un entorno gráfico intuitivo para trabajar con bases de datos, facilitando la creación y administración de esquemas, la ejecución de consultas SQL, y la visualización de resultados. Entre sus principales características se destacan los siguientes puntos:

✓ **Diseño y modelado de bases de datos**

MySQL Workbench permite a los usuarios crear modelos de bases de datos visualmente, lo que simplifica la planificación y el diseño de estructuras complejas.

✓ **Ejecución de consultas y desarrollo SQL**

Proporciona un editor SQL avanzado con resaltado de sintaxis y autocompletado, lo que facilita la escritura y depuración de consultas.

✓ **Administración de servidores MySQL**

Permite la gestión de usuarios, la configuración de permisos, la monitorización del rendimiento del servidor y la realización de copias de seguridad, todo desde una interfaz gráfica.

✓ **Migración de bases de datos**

MySQL Workbench soporta la migración desde diversos sistemas de gestión de bases de datos hacia MySQL, haciendo el proceso más eficiente y menos propenso a errores.

2.7. LENGUAJES DE PROGRAMACIÓN

Los lenguajes de programación son herramientas esenciales en el desarrollo de software, ya que permiten a los programadores escribir instrucciones que las computadoras pueden ejecutar. Cada lenguaje de programación tiene sus propias características y paradigmas, que influyen en cómo se diseñan y desarrollan las aplicaciones. Los lenguajes de programación pueden ser clasificados en varios tipos, como lenguajes de bajo nivel, de alto nivel, compilados e interpretados. La elección del lenguaje adecuado depende de los requisitos del proyecto, el entorno de ejecución y la experiencia del desarrollador (Sommerville, 2011).

2.7.1. PHP

PHP (Hypertext Preprocessor) es un lenguaje de programación de código abierto ampliamente utilizado para el desarrollo de aplicaciones web. Fue diseñado inicialmente para el procesamiento de formularios en el servidor y para generar contenido dinámico en sitios web. PHP se integra fácilmente con bases de datos y otros servicios web, lo que lo hace ideal para desarrollar aplicaciones web interactivas y sitios web dinámicos.

PHP se ejecuta en el servidor, lo que significa que el código PHP se procesa en el servidor antes de que la página web se envíe al navegador del usuario. Esto permite generar contenido HTML dinámico basado en la lógica de negocio y los datos almacenados en bases de datos. PHP es conocido por su simplicidad y flexibilidad, así como por su amplia comunidad de desarrolladores y recursos disponibles. Las versiones modernas de PHP también incluyen características avanzadas como la programación orientada a objetos, manejo de excepciones y soporte para varios sistemas de bases de datos (Welling & Thomson, 2016).

2.7.2. JAVASCRIPT

JavaScript es un lenguaje de programación interpretado que se utiliza principalmente para crear contenido interactivo en páginas web. Desarrollado originalmente para su uso en navegadores web, JavaScript permite a los desarrolladores añadir funcionalidades dinámicas y responder a eventos del usuario sin necesidad de recargar la página. JavaScript se ejecuta en el lado del cliente, lo que significa que el código se procesa en el navegador del usuario.

JavaScript es un lenguaje versátil que soporta múltiples paradigmas de programación, incluyendo programación imperativa, orientada a objetos y funcional. Además de su uso en el desarrollo web, JavaScript ha evolucionado con el tiempo y se ha extendido a entornos de desarrollo de servidor mediante tecnologías como Node.js. Esto permite a los desarrolladores utilizar JavaScript tanto en el front-end como en el back-end de las aplicaciones, creando un entorno de desarrollo más coherente y eficiente. La popularidad de JavaScript también ha llevado al desarrollo de numerosos frameworks y bibliotecas, como React y Angular, que facilitan la creación de aplicaciones web complejas (Flanagan, 2020).

2.8. REPORTES

Los reportes son documentos o presentaciones que consolidan y presentan datos relevantes de manera estructurada para facilitar la toma de decisiones y el análisis. En el contexto de un sistema académico para la administración de calificaciones, los reportes juegan un papel crucial al proporcionar una visión clara del rendimiento

académico de los estudiantes, la efectividad de los docentes y el estado general del proceso educativo.

Además, los reportes pueden incluir una variedad de información, como estadísticas, gráficos, y resúmenes, y son utilizados por administradores, docentes, y estudiantes para evaluar y mejorar el desempeño académico. Las principales características de los reportes incluyen:

✓ **Consolidación de datos**

Agregan información proveniente de diferentes fuentes y la presentan de manera coherente.

✓ **Visualización de datos**

Utilizan gráficos y tablas para facilitar la interpretación de los datos y la identificación de tendencias.

✓ **Personalización y filtrado**

Permiten ajustar el contenido y el formato del reporte según las necesidades específicas de los usuarios o del contexto.

✓ **Generación automática**

En algunos sistemas, los reportes se generan automáticamente a partir de datos actualizados, reduciendo la carga de trabajo manual y minimizando errores.

2.8.1. REPORTE ACADÉMICO DE NOTAS

Es un tipo específico de reporte que se centra en la presentación de las calificaciones y el rendimiento académico de los estudiantes. Este reporte proporciona un resumen detallado de las calificaciones obtenidas por los estudiantes en diferentes asignaturas o periodos académicos, y suele ser utilizado para la evaluación y seguimiento del progreso académico.

✓ **Resumen de calificaciones**

Muestra las notas obtenidas por el estudiante en cada asignatura o módulo, y puede incluir promedios generales y detalles adicionales como la escala de calificación utilizada.

✓ **Comentarios y observaciones**

Permite incluir observaciones de los docentes sobre el rendimiento del estudiante, destacando fortalezas y áreas de mejora.

✓ **Comparación de rendimientos**

Facilita la comparación del desempeño del estudiante a lo largo del tiempo o con respecto a sus compañeros, mediante gráficos y tablas.

✓ **Formato estandarizado**

Puede seguir un formato estándar o personalizado según las políticas de la institución educativa, asegurando la consistencia y claridad en la presentación de los datos.

2.9. NORMAS Y LEYES

2.9.1. LEY N°164

✓ **ARTICULO 1 (OBJETO)**

Establecer las normas para el acceso, uso y desarrollo de las Tecnologías de Información y Comunicación (TIC), de acuerdo con el título IV de la Ley N° 164 del 8 de agosto de 2011, sobre Telecomunicaciones, Tecnologías de Información y Comunicación.

✓ **ARTICULO 2 (ÁMBITO DE APLICACIÓN)**

Este Reglamento será aplicable a individuos o entidades, tanto públicas como privadas, que realicen actividades o brinden servicios relacionados con la certificación digital, gobierno electrónico, software libre, correo electrónico y el uso de documentos y firmas en el Estado Plurinacional de Bolivia. (Constitución Política del Estado, 2013)

✓ **ARTICULO 3 (DEFINICIONES)**

Además de las definiciones técnicas establecidas en la Ley N° 164, se adoptan las siguientes definiciones para la aplicación de este Reglamento:

I. Respetto al desarrollo de contenido y aplicaciones

a. Contenido Digital

Información producida en cualquier formato que puede ser distribuida a través de medios electrónicos y forma parte de un mensaje que el sistema de transferencia o almacenamiento no examina ni modifica, excepto para la convención durante su transporte.

b. Aplicaciones Digitales

Programas de software modulares, específicos e interactivos, ya sea para usuarios individuales o múltiples, utilizados en plataformas de servicios digitales o equipos destinados a comunicaciones personales, educación, producción o entretenimiento, entre otros.

II. Respetto al Software Libre

a. Programa o Software

Cualquier conjunto de instrucciones finitas que un dispositivo de procesamiento de datos utiliza para realizar una tarea específica o resolver un problema, incluyendo todas las dependencias necesarias para su funcionamiento completo.

b. Código Fuente o Programa Fuente

El conjunto completo de instrucciones y archivos digitales originales, legibles para el ser humano, escritos por el programador en un lenguaje de programación específico, junto con todos los archivos de soporte necesarios, como tablas de datos, imágenes, especificaciones, documentación y cualquier otro elemento necesario para crear el programa ejecutable.

c. Software Libre

El software licenciado por su autor bajo una licencia de código de fuente abierta permite al usuario ejercer las siguientes libertades sea, ejecutar el software para cualquier

propósito sin restricción alguna, estudiar cómo funciona el software y modificarlo para cumplir un objetivo específico, mediante el acceso al código fuente y todos los componentes que posibilitan su funcionamiento. El acceso al código fuente es una condición necesaria e imprescindible para estas libertades.



2.10. HERRAMIENTAS DE DESARROLLO

Las herramientas de desarrollo son aplicaciones y entornos diseñados para facilitar el proceso de creación de software. Estas herramientas abarcan una amplia gama de utilidades, desde entornos de desarrollo integrados (IDEs) hasta sistemas de control de versiones y plataformas de gestión de proyectos. Las herramientas de desarrollo proporcionan a los programadores y equipos de desarrollo los recursos necesarios para escribir, probar, depurar y mantener el código de manera eficiente y efectiva.

Tabla 2.8

Herramientas de Desarrollo para el Proyecto

LOGO	NOMBRE	DESCRIPCIÓN
	BALSAMIQ	Esta herramienta será utilizada para el maquetado de la aplicación e interfaces.
	BIZAGI MODELER	Servirá como herramienta para modelar y documentar procesos.
	DIAGRAMS.NET	Es un software alojado en la nube que permite diseñar cualquier tipo de diagramas como organigramas, de red, mapas conceptuales, entre otros.

 Office	<i>MICROSOFT OFFICE</i>	<p>Se utilizará para la realización de la documentación.</p>
 Visual Studio Code	<i>VISUAL STUDIO CODE</i>	<p>Es un editor de código fuente independiente que se ejecuta en Windows, macOS y Linux, con una gran cantidad de extensiones para admitir casi cualquier lenguaje de programación.</p>
 XAMPP	<i>XAMPP</i>	<p>Es un paquete de software gratuito y de código abierto que proporciona una solución completa para el desarrollo de aplicaciones web. Incluye un servidor web Apache, una base de datos MySQL (o MariaDB), y soporte para PHP.</p>
 BOOTSTRAP	<i>BOOTSTRAP</i>	<p>Se utilizará para desarrollar interfaces de usuario responsivas y atractivas de manera rápida y eficiente.</p>

Nota. Herramientas de desarrollo que se utilizarán para el proyecto, se colocaron las más esenciales

2.11. MÉTODO DE ANÁLISIS DE COSTO Y BENEFICIO

El método de análisis de costo y beneficio generalmente incluye la identificación y estimación de todos los costos y beneficios asociados con el proyecto, la evaluación de los impactos financieros y no financieros, y la realización de una evaluación comparativa. Las técnicas de análisis pueden variar, pero comúnmente incluyen métodos de cálculo de retorno de inversión (ROI), valor presente neto (VPN) y análisis de costo-beneficio (Schmid, 2017).

2.11.1. COCOMO II

COCOMO II (Constructive Cost Model II) es un modelo de estimación de costos de software desarrollado por Barry Boehm y su equipo. Es una evolución del modelo COCOMO original y se utiliza para estimar el esfuerzo, el costo y el tiempo necesario para desarrollar un proyecto de software. Además, este modelo proporciona una metodología detallada y estructurada para realizar estas estimaciones, basándose en factores como el tamaño del software, la complejidad del proyecto y las características del equipo de desarrollo.

En este modelo de estimación se utilizan las siguientes ecuaciones.

Estimación de Esfuerzo:

$$E = a \times (KLDC)^b \times m(X) \quad (2.1)$$

Donde:

- E es el Esfuerzo Estimado en persona-mes.
- a es Parámetro que depende del tipo de proyecto.
- $KLDC$ es las Kilolíneas de código fuente (tamaño del proyecto).
- b es Exponente que también depende del tipo de proyecto.
- $m(X)$ es el Factor Multiplicativo que considera características específicas del proyecto como el riesgo y la complejidad.

Estimación de la Duración:

$$D = c \times E \times d \quad (2.2)$$

Dónde:

- D es la Duración estimada en meses
- E es la Estimación de Esfuerzo
- c, d son los Factores que afectan la duración del proyecto

Estimación de la Productividad:

$$P = \frac{E}{D} \quad (2.3)$$

Dónde:

- P es la Productividad del equipo en personas
- E es el Esfuerzo estimado
- D es la Duración estimada

3. TERCER CAPÍTULO

MARCO APLICATIVO

3.1. ANÁLISIS DE LA SITUACIÓN ACTUAL

El objetivo principal de este capítulo es desarrollar un sistema de gestión académica que integre todas las funciones esenciales que requiere el establecimiento. Además, se abordará la ingeniería del proyecto, detallando las fases de desarrollo del sistema mediante la metodología XP y el lenguaje de modelado UML en contraste con el funcionamiento general del proceso integrado de los módulos del sistema.

3.2. PLANIFICACIÓN

Se trazan principalmente las historias de usuario relevantes para la primera entrega del producto, junto con la familiarización con las herramientas, tecnologías y prácticas que se implementarán en el proyecto.

3.2.1. ANÁLISIS DE TIPOS DE USUARIO

Para el presente proyecto nos encontramos con tres tipos de usuarios con diferentes roles y una relación jerárquica entre ellos.

Tabla 3.1

Identificación de Roles y Tareas

USUARIO	DESCRIPCIÓN
ESTUDIANTE	<ul style="list-style-type: none"> ❖ Visualización de Calificaciones: El estudiante puede acceder a su perfil y visualizar las calificaciones de los cursos a cuáles se inscribió. ❖ Consulta de Boletines Académicos: Puede descargar e imprimir los boletines académicos de los cursos a cuáles se inscribió.

	<p>❖ Perfil de Usuario: El estudiante puede actualizar su información personal básica dentro de los límites permitidos por el sistema.</p>
DOCENTE	<p>I. Registrar y listar calificaciones: El docente puede registrar las calificaciones correspondientes a los bimestres de cada curso habilitado durante la gestión.</p> <p>II. Buscar Estudiante: El docente puede revisar el rendimiento académico de cierto estudiante.</p> <p>III. Perfil de Usuario: El docente puede actualizar su información personal básica dentro de los límites permitidos por el sistema.</p>
ADMINISTRADOR	<p>IV. Gestión de Usuarios: El administrador tiene la capacidad de crear, modificar y eliminar cuentas de usuarios sea de estudiantes, docentes u otros administradores.</p> <p>V. Administración de Gestiones, Carreras y Cursos: El administrador tiene la capacidad de administrar las gestiones, carreras y cursos disponibles para los estudiantes.</p> <p>VI. Gestión de los Estudiantes: El administrador puede actualizar todo tipo de información académica y personal respecto a los estudiantes.</p> <p>VII. Visualización y generación de reportes: El administrador puede visualizar diversos gráficos sobre el rendimiento estudiantil, así como generar diferentes reportes administrativos y académicos.</p>

Nota. Descripción de todos los usuarios involucrados para el sistema con sus funcionalidades y características

3.2.2. REQUERIMIENTOS FUNCIONALES

Tabla 3.2

Requerimientos Funcionales del Sistema

CÓDIGO	NOMBRE	DESCRIPCIÓN
RF-1	GESTIÓN DE USUARIOS	❖ El sistema debe permitir al administrador crear, modificar y eliminar cuentas de usuarios, incluyendo estudiantes, docentes y otros administradores. Esto incluye la gestión de permisos y roles asociados a cada cuenta.
RF-2	ADMINISTRACIÓN DE GESTIONES Y CARRERAS	❖ El sistema debe permitir administrar las gestiones y carreras del estudiantado, de este modo, se podrá gestionar los cursos.
RF-3	GESTIÓN DE CURSOS	❖ El sistema debe permitir al administrador gestionar los cursos ofrecidos, incluyendo la creación, modificación y eliminación de cursos.
RF-4	GESTIÓN DE ESTUDIANTES	VIII. El sistema debe permitir al administrador gestionar la información de los estudiantes. Esto incluye la capacidad de actualizar datos personales y académicos, así como gestionar las inscripciones en cursos.
RF-5	GESTIÓN DE HISTORIAL ACADÉMICO	IX. El sistema debe permitir la gestión del historial académico de los estudiantes, incluyendo la actualización de calificaciones, la visualización de boletines académicos y el seguimiento del

		rendimiento académico a lo largo del tiempo.
RF-6	GESTIÓN DE DOCENTES	X. El sistema debe proporcionar funcionalidades para la gestión de la información de los docentes. Esto incluye la capacidad de agregar, actualizar y eliminar perfiles de docentes.
RF-7	REPORTES DE GESTIÓN ACADÉMICA	XI. El sistema debe permitir al administrador generar reportes académicos y analíticos sobre la institución. Esto incluye reportes sobre el rendimiento académico de los estudiantes, informes de gestión y otros análisis relevantes para la administración.

Nota. Descripción de todos los requerimientos funcionales del sistema web de gestión académica

3.2.3. REQUERIMIENTOS NO FUNCIONALES

Tabla 3.3

Requerimientos No Funcionales del Sistema

CÓDIGO	NOMBRE	DESCRIPCIÓN
RNF-1	SEGURIDAD	❖ El sistema garantizara que los usuarios autenticados tengan acceso sólo a los recursos y datos que están autorizados a ver y modificar, además de que la seguridad de la información no se vea comprometida.
RNF-2	INTERFAZ DE USUARIO	❖ Para proporcionar una experiencia de usuario intuitiva y atractiva se incluirá características como un diseño atractivo,

		con una navegación sencilla, una interfaz de usuario coherente y consistente en el proyecto.
RNF-3	ESCALABILIDAD	XII. El sistema debe ser escalable para permitir un crecimiento futuro y ser capaz de aumentar o disminuir el tamaño de los recursos según las necesidades de la aplicación, utilizando al máximo los servicios PHP y Bootstrap.
RNF-4	DISEÑO RESPONSIVO	XIII. El sistema web tendrá la capacidad de adaptarse a diferentes tamaños de pantalla y dispositivos, lo que permitirá a los usuarios acceder y utilizar el sistema desde diferentes dispositivos móviles, tabletas o computadoras de escritorio.
RNF-5	DISEÑO DE PROGRAMACIÓN	XIV. Se programará el código del sistema tomando en cuenta la legibilidad, la mantenibilidad, la escalabilidad y la capacidad de reutilización del código para futuras ediciones que se requiera del proyecto.
RNF-6	MANTENIBILIDAD	XV. La plataforma web debe ser sencillo de mantener con una estructura clara, Además, debe contar con documentación detallada y actualizada que explique el funcionamiento del sistema.

RNF-7 FLEXIBILIDAD

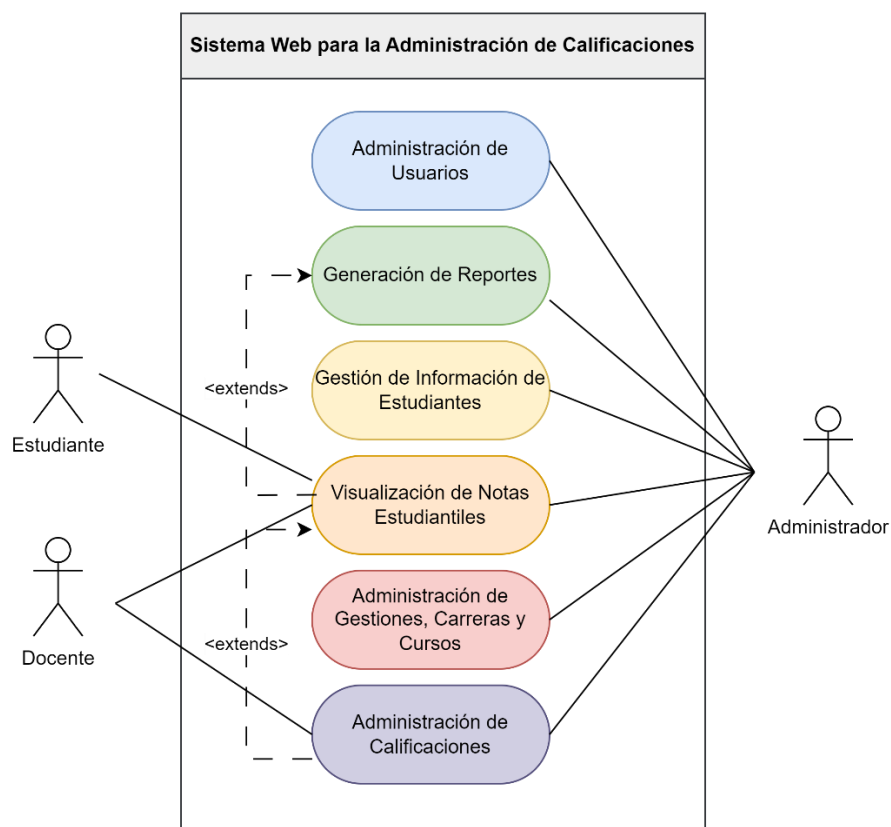
XVI. Al trabajar con el lenguaje PHP, este es compatible con una amplia gama de plataformas, lenguajes de programación y tecnologías de terceros, con esta flexibilidad se permitirá utilizar los recursos de manera efectiva para cumplir con los requerimientos del proyecto en desarrollo.

Nota. Descripción de los requerimientos no funcionales del sistema web de gestión académica

3.2.4. DIAGRAMA DE CASO DE USO DEL SISTEMA

Figura 3.1

Diagrama General de Caso de Uso del Sistema



Nota. Elaboración propia del diagrama de caso de uso general del proyecto

3.3. APLICACIÓN DE LA METODOLOGÍA XP

3.3.1. FASE I – PLANIFICACIÓN

En esta primera fase, se definen las prioridades clave del proyecto. Se comienza por identificar las historias de usuario más relevantes y se decide qué funcionalidades se implementarán en la siguiente iteración.

Tabla 3.4

Historias de Usuario del Proyecto

CÓDIGO	NOMBRE
HU-01	CONTROL DE ACCESO DE USUARIOS
HU-02	ADMINISTRACIÓN DE GESTIONES Y CARRERAS
HU-03	ADMINISTRACIÓN DE CURSOS
HU-04	ADMINISTRACIÓN DE ESTUDIANTES
HU-05	ADMINISTRACIÓN DE CALIFICACIONES
HU-06	ADMINISTRACIÓN DE DOCENTES
HU-07	GENERACIÓN DE GRÁFICAS ADMINISTRATIVAS

Nota. Historias de Usuario del Proyecto

Considerando los datos de la tabla anterior, se empieza a describir las historias de usuario individualmente, así como las tareas a realizar para cada caso. Además, se especificarán las tareas para cada historia de usuario, las cuales se desarrollarán posteriormente.

Tabla 3.5*Historia de Usuario 1: HU-01*

HISTORIA DE USUARIO	HU-01
NOMBRE DE HISTORIA: CONTROL DE ACCESO DE USUARIOS	
NUMERO: 01	USUARIO: ADMINISTRADOR
DÍAS DE DESARROLLO: 5	PRIORIDAD: ALTA
<p>DESCRIPCIÓN: El administrador puede registrar nuevos usuarios en el sistema a través del botón de registro, que dirige a un formulario destinado a tal propósito. Una vez completado el registro, todos los usuarios se podrán visualizar en una tabla, donde también estarán disponibles los botones para editar o eliminar usuarios, ya sean administradores, docentes o estudiantes.</p> <p>Por otro lado, una vez que los usuarios hayan registrado sus datos en el sistema, podrán iniciar sesión en la interfaz de login utilizando su número de carnet y contraseña. Al ingresar, se podrá acceder a las funcionalidades correspondientes. Para el administrador, estas funciones incluyen la creación de perfiles para estudiantes, docentes y otros administradores, así como la asignación de roles y permisos adecuados.</p>	
<p>OBSERVACIONES: El administrador, docente y estudiante son los únicos usuarios para el ingreso al sistema.</p>	

Nota. Primera historia de usuario con todas sus características

Tabla 3.6*Tareas a Realizar: HU-01*

NÚMERO DE TAREA	TAREA A REALIZAR
1.1.	Diseñar la base de datos para los usuarios
1.2.	Implementar una interfaz de autenticación de usuarios
1.3.	Implementar una interfaz para el registro de usuarios

1.4.	Desarrollar una interfaz para el listado de usuarios
------	--

Nota. Tareas a realizar por parte de la primera historia de usuario

Tabla 3.7

Historia de Usuario 2: HU-02

HISTORIA DE USUARIO	HU-02
NOMBRE DE HISTORIA: ADMINISTRACIÓN DE GESTIONES Y CARRERAS	
NUMERO: 02	USUARIO: ADMINISTRADOR
DÍAS DE DESARROLLO: 7	PRIORIDAD: ALTA
DESCRIPCIÓN: El administrador puede gestionar las operaciones a través de botones en la interfaz de gestiones, permitiendo registrar, editar y eliminar gestiones. Asimismo, el usuario podrá gestionar las carreras disponibles en el instituto utilizando botones en la interfaz de carreras, garantizando que los estudiantes se inscriban en la opción adecuada. Todos estos registros se almacenarán en tablas, mostrando datos relevantes para el administrador.	
OBSERVACIONES: Ninguna.	

Nota. Segunda historia de usuario con todas sus características

Tabla 3.8

Tareas a Realizar: HU-02

NÚMERO DE TAREA	TAREA A REALIZAR
2.1.	Diseñar la base de datos de las gestiones y carreras
2.2.	Realizar una interfaz para administrar las gestiones
2.3.	Realizar una interfaz para administrar los cursos

Nota. Tareas a realizar por parte de la segunda historia de usuario

Tabla 3.9*Historia de Usuario 3: HU-03*

HISTORIA DE USUARIO	HU-03
NOMBRE DE HISTORIA: ADMINISTRACIÓN DE CURSOS	
NUMERO: 03	USUARIO: ADMINISTRADOR
DÍAS DE DESARROLLO: 7	PRIORIDAD: ALTA
DESCRIPCIÓN: El administrador puede gestionar las operaciones en la interfaz de cursos, donde puede registrar, editar y eliminar los cursos relacionados con las carreras y gestiones a través de botones colocados de manera intuitiva para el usuario. Además, esta interfaz presenta una tabla con todos los cursos registrados y los botones correspondientes	
OBSERVACIONES: Ninguna.	

Nota. Tercera historia de usuario con todas sus características

Tabla 3.10*Tareas a Realizar: HU-03*

NÚMERO DE TAREA	TAREA A REALIZAR
3.1.	Diseñar la base de datos de los cursos
3.2.	Realizar una interfaz para el registro de cursos
3.3.	Realizar una interfaz para administrar los cursos

Nota. Tareas a realizar por parte de la tercera historia de usuario

Tabla 3.11*Historia de Usuario 4: HU-04*

HISTORIA DE USUARIO		HU-04
NOMBRE DE HISTORIA: ADMINISTRACIÓN DE ESTUDIANTES		
NUMERO: 04		USUARIO: ADMINISTRADOR
DÍAS DE DESARROLLO: 7		PRIORIDAD: ALTA
DESCRIPCIÓN: El administrador puede gestionar la información de los estudiantes mediante botones para registrar y editar notas, asegurando que sus datos personales y académicos estén actualizados. Además, estos botones permiten actualizar la información del perfil, gestionar inscripciones en cursos y revisar el historial académico. Todos estos datos se presentan en la interfaz de estudiantes a través de tablas accesibles para el administrador.		
OBSERVACIONES: Ninguna.		

Nota. Cuarta historia de usuario con todas sus características

Tabla 3.12*Tareas a Realizar: HU-04*

NÚMERO DE TAREA	TAREA A REALIZAR
4.1.	Diseñar la base de datos para la gestión de la información de los estudiantes y su relación con los cursos
4.2.	Implementar un módulo web para el registro de nuevos estudiantes
4.3.	Desarrollar un apartado web para la administración de información de cada estudiante

Nota. Tareas a realizar por parte de la cuarta historia de usuario

Tabla 3.13*Historia de Usuario 5: HU-05*

HISTORIA DE USUARIO		HU-05
NOMBRE DE HISTORIA: ADMINISTRACIÓN DE CALIFICACIONES		
NUMERO: 05	USUARIO: ADMINISTRADOR Y ESTUDIANTE	
DÍAS DE DESARROLLO: 7	PRIORIDAD: ALTA	
DESCRIPCIÓN: El administrador puede realizar diversas operaciones con la información académica de los estudiantes utilizando botones para registrar calificaciones específicas, teniendo en cuenta sus datos relacionados con el curso, la carrera y la gestión. Por otro lado, los estudiantes pueden acceder a su perfil para visualizar sus calificaciones e historial académico a través de una interfaz que incluye botones y funciones de búsqueda de notas.		
OBSERVACIONES: Ninguna.		

Nota. Quinta historia de usuario con todas sus características

Tabla 3.14*Tareas a Realizar: HU-05*

NÚMERO DE TAREA	TAREA A REALIZAR
5.1.	Diseñar la base de datos para la gestión de calificaciones del estudiantado de acuerdo a los requerimientos del estudiante y del administrador
5.2.	Implementar un apartado web para la administración de las calificaciones de parte del administrador

5.3.	Desarrollar un módulo web para la visualización de notas de manera gráfica y descriptiva tanto para el estudiante como para el administrador
------	--

Nota. Tareas a realizar por parte de la quinta historia de usuario

Tabla 3.15

Historia de Usuario 6: HU-06

HISTORIA DE USUARIO	HU-06
NOMBRE DE HISTORIAL: REGISTRO DE DOCENTES	
NUMERO: 06	USUARIO: ADMINISTRADOR
DÍAS DE DESARROLLO: 5	PRIORIDAD: MEDIA
DESCRIPCIÓN: El administrador puede gestionar el registro de docentes, considerando su información personal y experiencia académica, mediante botones de registro y edición que conducen a un formulario completo. Basándose en estos datos, se podrán asignar cursos a cada docente. Posteriormente, esta información se presenta en una lista que permite visualizar fácilmente a todos los docentes registrados.	
OBSERVACIONES: Ninguna.	

Nota. Sexta historia de usuario con todas sus características

Tabla 3.16

Tareas a Realizar: HU-06

NÚMERO DE TAREA	TAREA A REALIZAR
6.1.	Diseñar la base de datos para el registro de docentes
6.2.	Implementar un apartado web para el registro de docentes

6.3.	Implementar un módulo web para la administración de docentes
------	--

Nota. Tareas a realizar por parte de la sexta historia de usuario

Tabla 3.17

Historia de Usuario 7: HU-07

HISTORIA DE USUARIO	HU-07
NOMBRE DE HISTORIA: GENERACIÓN DE GRÁFICAS ADMINISTRATIVAS	
NUMERO: 07	USUARIO: ADMINISTRADOR
DÍAS DE DESARROLLO: 4	PRIORIDAD: MEDIA
DESCRIPCIÓN: El administrador puede generar boletines en formato PDF para un estudiante específico utilizando un botón de búsqueda. Además, puede visualizar datos sobre el rendimiento de los estudiantes y la gestión institucional en el panel principal del proyecto, lo que facilita la toma de decisiones y el seguimiento del desempeño académico.	
OBSERVACIONES: Ninguna.	

Nota. Séptima historia de usuario con todas sus características

Tabla 3.18

Tareas a Realizar: HU-07

NÚMERO DE TAREA	TAREA A REALIZAR
7.1.	Identificar los reportes y gráficas más relevantes a mostrar para el administrador

-
- 7.2. Implementar diferentes reportes y gráficas académicas que permitan al administrador visualizar el rendimiento de los estudiantes a través de los cursos
-

Nota. Tareas a realizar por parte de la séptima historia de usuario

3.3.2. PLANIFICACIÓN DE ITERACIONES

Una de las características más representativas de la metodología XP es la programación por iteraciones, donde cada iteración representa un ciclo de trabajo en el que se abordan una o varias historias de usuario. Dada su importancia, el proyecto se dividirá en cuatro iteraciones.

Tabla 3.19

Planificación de Iteraciones del Proyecto

ITERACIÓN	N°	HISTORIA DE USUARIO	INICIO	FIN	PRIORIDAD
PRIMERA	HU-01	REGISTRO DE USUARIOS	23/08/24	29/08/24	ALTA
	HU-02	ADMINISTRACIÓN DE GESTIONES Y CARRERAS	30/08/24	05/09/24	ALTA
	HU-03	ADMINISTRACIÓN DE CURSOS	06/09/24	12/09/24	ALTA
SEGUNDA	HU-03	ADMINISTRACIÓN DE ESTUDIANTES	13/09/24	19/09/24	ALTA
	HU-04	ADMINISTRACIÓN DE CALIFICACIONES	20/09/24	26/09/25	ALTA

TERCERA	HU-06	REGISTRO DE DOCENTES	27/09/24	01/10/24	MEDIA
CUARTA	HU-07	GENERACIÓN DE GRÁFICAS ADMINISTRATIVAS	02/10/24	06/10/24	MEDIA

Nota. Planificación de iteraciones del proyecto considerando todas las historias de usuario.

3.4. FASE II – DISEÑO

En esta fase se presentarán el desarrollo de las tareas presentadas anteriormente, además se considerarán las iteraciones, diferentes diagramas e interfaces que promete el sistema web.

3.4.1. PRIMERA ITERACIÓN

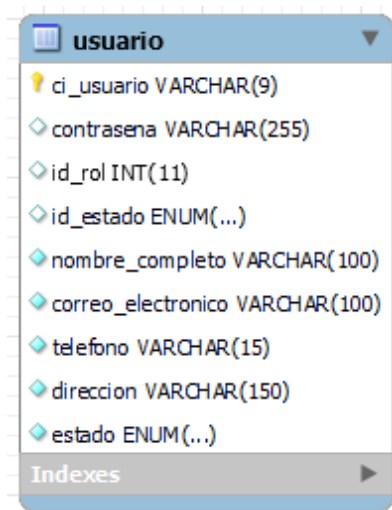
En esta primera iteración se considerará la implementación de la primera historia de usuario.

I. HISTORIA DE USUARIO 1: REGISTRO DE USUARIOS

❖ TAREA 1.1. DISEÑAR LA BASE DE DATOS PARA LOS USUARIOS

Figura 3.2

Diagrama de Base de Datos de los Usuarios



Nota. Diagrama de base de datos de los usuarios obtenida de *MySQL Workbench*

❖ TAREA 1.2. IMPLEMENTAR UNA INTERFAZ DE AUTENTICACIÓN DE USUARIOS

Figura 3.3

Interfaz web de Inicio de Sesión



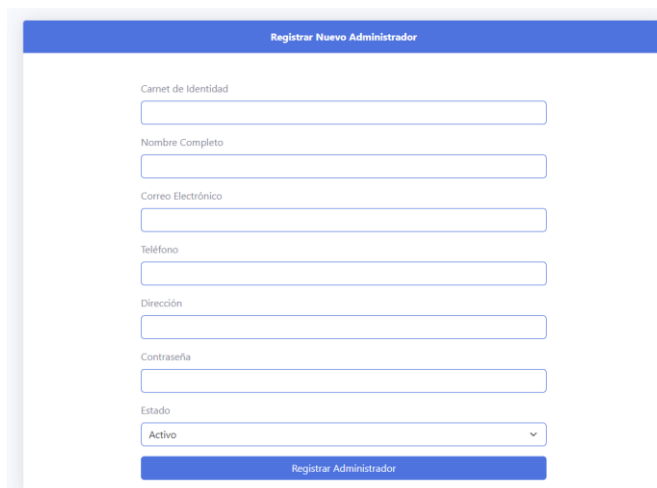
The screenshot shows a login form titled "Bienvenido" (Welcome). On the left is the logo of the Instituto Tecnológico del Occidente, which features a gear, a sun, and a mountain. The form contains two input fields: "Carnet de Identidad" (ID Card) and "Contraseña" (Password). Below these fields is a blue button labeled "Iniciar Sesión" (Log In). The background of the form is a light blue gradient with a faint image of a brain.

Nota. Inicio de sesión para los tres usuarios involucrados, podrán ingresar con los datos de su carnet de identidad y contraseña.

❖ TAREA 1.3. DESARROLLAR UNA INTERFAZ PARA EL REGISTRO DE USUARIOS

Figura 3.4

Interfaz web para registrar usuarios



The screenshot shows a registration form titled "Registrar Nuevo Administrador" (Register New Administrator). The form contains several input fields: "Carnet de Identidad" (ID Card), "Nombre Completo" (Full Name), "Correo Electrónico" (Email), "Teléfono" (Phone), "Dirección" (Address), and "Contraseña" (Password). There is also a dropdown menu for "Estado" (State) with "Activo" (Active) selected. At the bottom is a blue button labeled "Registrar Administrador" (Register Administrator).

Nota. El formulario de registro de usuarios recopila suficientes datos personales para completar el proceso de registro. Además, el registro de docentes y estudiantes se realiza automáticamente al momento de incorporarlos al sistema.

❖ TAREA 1.4. DESARROLLAR UNA INTERFAZ PARA EL LISTADO DE USUARIOS

Figura 3.5

Interfaz web para listar los diferentes usuarios

Listar Usuarios							
Administradores Registrados							
Carnet de Identidad	Nombre Completo	Correo Electrónico	Teléfono	Dirección	Estado	Rol	Acciones
123456	fred	fred123@gmail.com	7654213	Av. Perus	Activo	Administrador	✎ Editar ✖ Eliminar
912221	admin	admin@gmail.com	512312321	Calle 1234	Activo	Administrador	✎ Editar ✖ Eliminar
Estudiantes Registrados							
Carnet de Identidad	Nombre Completo	Correo Electrónico	Teléfono	Dirección	Estado	Rol	Acciones
36287495	Juan Pablo Ramirez Torres	juan.ramirez.torres@gmail.com	77823123	Av. Stadium	Activo	Estudiante	✎ Editar ✖ Eliminar
40728193	Miguel Angel Torres Morales	miguel.torrez.morales@gmail.com	77281233	Av. Villa Victoria	Activo	Estudiante	✎ Editar ✖ Eliminar
56198274	Elena Gabriela Castro Fernandez	elena.castro.fer@gmail.com	77281234	Av. Chasquipampa	Activo	Estudiante	✎ Editar ✖ Eliminar
58192734	Ana Maria Fernandez Vargas	ana.fernandez.vargas@gmail.com	7818282	Av. Siempre Viva	Activo	Estudiante	✎ Editar ✖ Eliminar
72983415	Lucia Carolina Vargas Ortiz	lucia.vargas.ortiz@gmail.com	77281233	Av. Prado	Activo	Estudiante	✎ Editar ✖ Eliminar
83492715	Carlos Andres Mamani Lopez	carlos.mamani.lopez@gmail.com	77888812	Av. Peru	Activo	Estudiante	✎ Editar ✖ Eliminar

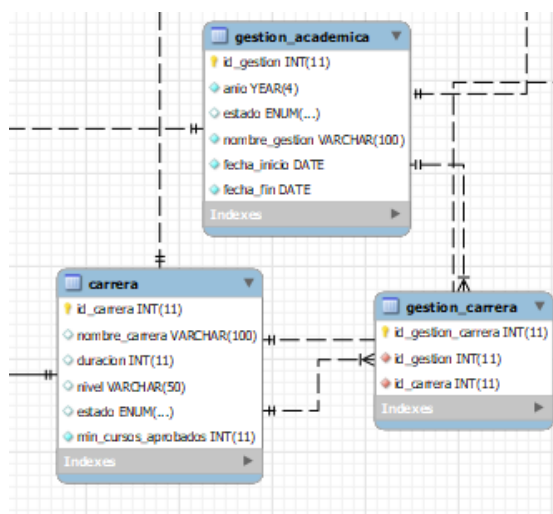
Nota. Se tiene la lista de todos los usuarios registrados en el sistema, estos se pueden añadir, editar y eliminar. Además, se muestra todos los datos esenciales del usuario como ser su nombre completo, correo electrónico, dirección y estado. Cabe mencionar que algunos de estos datos también pueden ser editados en el perfil del estudiante o docente.

II. HISTORIA DE USUARIO 2: ADMINISTRACIÓN DE GESTIONES Y CARRERAS

❖ TAREA 2.1. DISEÑAR LA BASE DE DATOS DE LAS GESTIONES Y CARRERAS

Figura 3.6

Diagrama de Base de Datos para gestiones y carreras



Nota. Diagrama de la base de datos diseñado para la gestión de las carreras en la institución, tomando en cuenta los datos fundamentales requeridos por la misma.

❖ TAREA 2.2. REALIZAR UNA INTERFAZ PARA LA ADMINISTRACIÓN DE GESTIONES

Figura 3.7

Interfaz web para la administración de gestiones

Gestiones Académicas					
Gestiones Registradas					
Nombre de la Gestión	Año	Fecha de Inicio	Fecha de Fin	Carreras Asociadas	Acciones
GESTION 2026	2026	2026-01-01	2026-12-10	SISTEMAS INFORMÁTICOS, DISEÑO GRAFICO, GASTRONOMÍA	Editar Eliminar
GESTIÓN 2025	2025	2024-09-13	2024-09-27	SISTEMAS INFORMÁTICOS, GASTRONOMÍA	Editar Eliminar
GESTIÓN 2024	2024	2024-09-03	2024-09-29	SISTEMAS INFORMÁTICOS, DISEÑO GRAFICO, GASTRONOMÍA	Editar Eliminar

Nota. Administración de todas las gestiones disponibles en el sistema, estos se pueden añadir, editar y eliminar la información.

❖ TAREA 2.3. IMPLEMENTAR UNA INTERFAZ PARA LA ADMINISTRACIÓN DE CARRERAS

Figura 3.8

Interfaz web para la administración de carreras

Listar Carreras

[+ Registrar Nueva Carrera](#)

Carreras Registradas								
Nombre de la Carrera	Duración (años)	Nivel	Estado	Min. Cursos Aprobados	Cursos Primer Año (Siglas)	Cursos Segundo Año (Siglas)	Cursos Tercer Año (Siglas)	Acciones
SISTEMAS INFORMÁTICOS	3	Técnico Superior	Activo	23	DPW-107 HDC-104 INT-103 MPI-101 OTM-106 PRG-102 TSO-105	ADS-206 BDD-208 DPW-207 EDD-203 EST-201 PDM-205 PRG-202 RDC-204	ADS-306 BDD-308 DPW-207 EMP-301 GMC-303 PDM-307 RDC-304 TMG-305	Editar Eliminar
DISEÑO GRAFICO	3	Técnico Superior	Activo	5	Ninguno	Ninguno	Ninguno	Editar Eliminar
GASTRONOMÍA	3	Técnico Superior	Activo	22	IAG-105 ING-102 ISM-101 NUT-106 OAB-107 PRP-104 SER-108 TEH-103	EMP-206 FRA-201 GIN-204 GNA-202 PAN-205 RYP-203	BAR-302 COC-301 EBG-304 ENM-303 IEG-306 MAG-305 TMG-307	Editar Eliminar

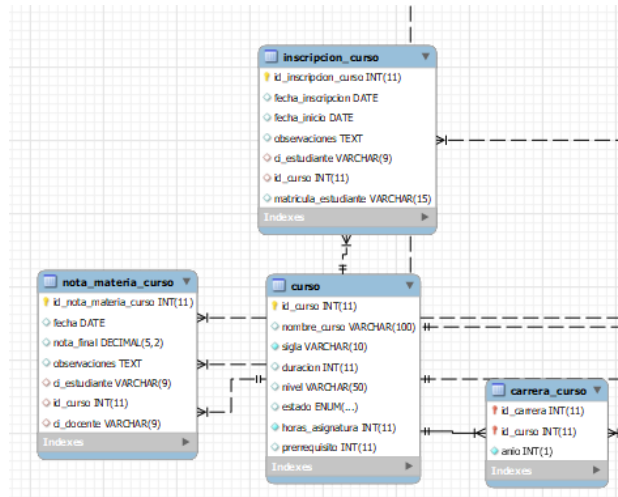
Nota. Administración de todas las carreras de la institución. Estas carreras tienen todas las materias que debe pasar el estudiante, se muestran estas materias o cursos de acuerdo al código que tiene cada uno. Además, cabe mencionar que se puede añadir, editar o eliminar una nueva carrera.

III. HISTORIA DE USUARIO 3: ADMINISTRACIÓN DE CURSOS

❖ TAREA 3.1. DISEÑAR LA BASE DE DATOS DE LOS CURSOS

Figura 3.9

Diagrama de Base de Datos para los cursos



Nota. Diagrama de la base de datos diseñado para los cursos de la institución, tomando en cuenta información crucial requeridos por la misma.

❖ TAREA 3.2. REALIZAR UNA INTERFAZ PARA EL REGISTRO DE CURSOS

Figura 3.10

Interfaz web para el registro de cursos

Registrar Nuevo Curso

Nombre del Curso

Nivel

Básico

Sigla del Curso

Estado

Activo

Horas de Asignatura

Seleccionar Carreras

☐ SISTEMAS INFORMÁTICOS
 ☐ DISEÑO GRAFICO
 ☐ GASTRONOMÍA

Prerrequisito de Materia

Ninguno

Duración (en semanas)

Registrar Curso

Nota. Formulario para el registro de nuevos cursos para la institución.

❖ TAREA 3.3. IMPLEMENTAR UNA INTERFAZ PARA ADMINISTRAR LOS CURSOS

Figura 3.11

Interfaz web para administrar los cursos

Listar Cursos

[+ Registrar Nuevo Curso](#)

Cursos Disponibles

Sigla	Nombre del Curso	Horas	Prerrequisito	Duración	Nivel	Estado	Carreras Asociadas	Acciones
INT-103	INGLÉS TÉCNICO	2	Ninguno	40	Básico	Activo	SISTEMAS INFORMÁTICOS	✎ Editar ✖ Eliminar
MPI-101	MATEMÁTICA PARA LA INFORMÁTICA	4	Ninguno	30	Básico	Activo	SISTEMAS INFORMÁTICOS	✎ Editar ✖ Eliminar
PRG-102	PROGRAMACIÓN I	8	Ninguno	30	Básico	Activo	SISTEMAS INFORMÁTICOS	✎ Editar ✖ Eliminar
HDC-104	HARDWARE DE COMPUTADORAS	4	Ninguno	30	Básico	Activo	SISTEMAS INFORMÁTICOS	✎ Editar ✖ Eliminar
TSO-105	TALLER DE SISTEMAS OPERATIVOS	4	Ninguno	30	Básico	Activo	SISTEMAS INFORMÁTICOS	✎ Editar ✖ Eliminar
OTM-106	OFIMÁTICA Y TECNOLOGÍA MULTIMEDIA	4	Ninguno	30	Básico	Activo	SISTEMAS INFORMÁTICOS	✎ Editar ✖ Eliminar

Nota. Listado de todos los cursos disponibles para la institución, estos cursos se relacionan con uno o muchas carreras de acuerdo al pensum del Ministerio de Gobierno. Se toma en cuenta también los prerrequisitos que tiene cada curso y los docentes involucrados en esas materias.

Además, que se pueden realizar ediciones de características en estos cursos.

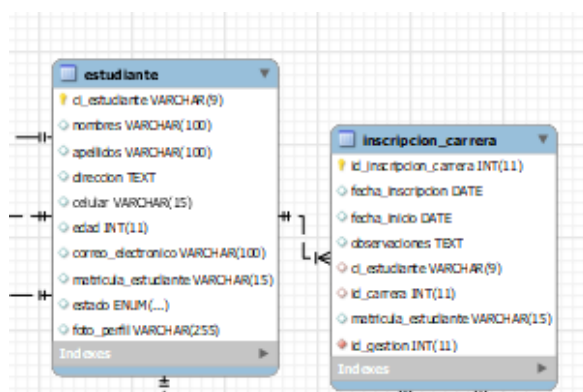
3.4.2. SEGUNDA ITERACIÓN

IV. HISTORIA DE USUARIO 4: ADMINISTRACIÓN DE ESTUDIANTES

- TAREA 4.1. DISEÑAR LA BASE DE DATOS PARA LA GESTIÓN DE LA INFORMACIÓN DE LOS ESTUDIANTES Y SU RELACIÓN CON LOS CURSOS

Figura 3.12

Diagrama de Base de Datos para la gestión de estudiantes-curso



Nota. Diagrama de la base de datos diseñado para la administración del registro y control de los estudiantes, considerando la información personal esencial del estudiantado.

- TAREA 4.2. IMPLEMENTAR UN MÓDULO WEB PARA EL REGISTRO DE NUEVOS ESTUDIANTES

Figura 3.13

Interfaz web para registro de estudiantes

Registrar Nuevo Estudiante

Datos del Estudiante	
Carnet de Identidad	Edad
<input type="text"/>	<input type="text"/>
Nombres	Correo Electrónico
<input type="text"/>	<input type="text"/>
Apellidos	Matrícula del Estudiante
<input type="text"/>	<input type="text"/>
Dirección	Carrera
<input type="text"/>	SISTEMAS INFORMÁTICOS
Celular	Gestión Académica
<input type="text"/>	GESTIÓN 2024
	Fecha de Inicio
	dd/mm/aaaa
	Observaciones (Opcional)
	<input type="text"/>

Nota. Formulario para el registro de estudiantes nuevos para la institución.








- **TAREA 4.3. DESARROLLAR UN APARTADO WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN DE CADA ESTUDIANTE**

Figura 3.14

Interfaz web para administrar datos de los estudiantes

Listar Estudiantes

[Registrar Estudiante](#)
[Exportar PDF](#)

Estudiantes Registrados									
Foto	Carnet de Identidad	Nombre Completo	Celular	Correo Electrónico	Estado	Carrera Inscrita	Fecha de Inicio	Gestiones	Acciones
	1111111	pepito debora torres	7664123	pep@gmail.com	Activo	SISTEMAS INFORMÁTICOS	11/09/2024	GESTIÓN 2024	Editar Eliminar
	36287495	Juan Pablo Ramirez Torres	77823123	juan.ramirez.torres@gmail.com	Activo	SISTEMAS INFORMÁTICOS	25/03/2024	GESTIÓN 2024, GESTIÓN 2025	Editar Eliminar
	40728193	Miguel Angel Torres Morales	77281233	miguel.torrez.morales@gmail.com	Activo	SISTEMAS INFORMÁTICOS	No registrada	GESTIÓN 2024, GESTIÓN 2025, GESTIÓN 2025	Editar Eliminar
	54321876	Maria Castillo	67727123	maria.castillo@gmail.com	Activo	GASTRONOMÍA	21/03/2024	GESTIÓN 2024	Editar Eliminar
	56198274	Elena Gabriela Castro Fernandez	77281234	elena.castro.fer@gmail.com	Activo	SISTEMAS INFORMÁTICOS	25/03/2024	GESTIÓN 2024	Editar Eliminar
	58192734	Ana Maria Fernandez Vargas	7818282	ana.fernandez.vargas@gmail.com	Activo	SISTEMAS INFORMÁTICOS	25/03/2024	GESTIÓN 2024, GESTIÓN 2025, GESTIÓN 2025	Editar Eliminar
	72983415	Lucia Carolina Vargas Ortiz	77281233	lucia.vargas.ortiz@gmail.com	Activo	SISTEMAS INFORMÁTICOS	No registrada	GESTIÓN 2024, GESTIÓN 2025, GESTIÓN 2025, GESTIÓN 2025	Editar

Nota. Listado de la información de todos los estudiantes registrados en el sistema. Estos datos pueden ser editados según sea necesario. Además, para registrar a un estudiante, se debe contar con la gestión actual, las carreras y las materias en las que el estudiante estará inscrito.

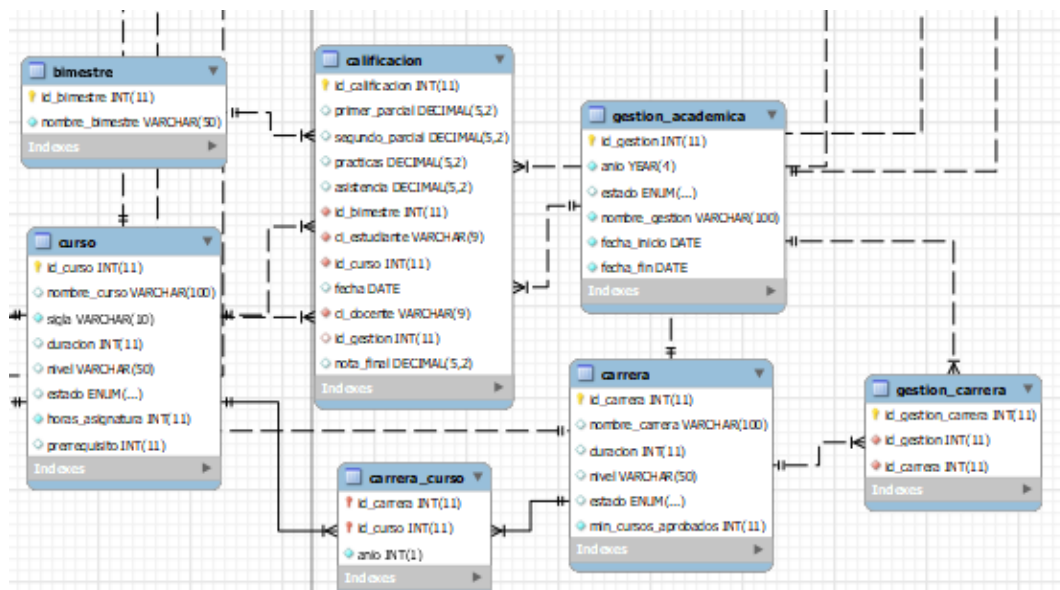
Cabe mencionar que algunos datos del estudiante pueden ser editados en el perfil del estudiante.

V. HISTORIA DE USUARIO 5: ADMINISTRACIÓN DE CALIFICACIONES

- TAREA 5.1. DISEÑAR LA BASE DE DATOS PARA LA GESTIÓN DE CALIFICACIONES DEL ESTUDIANTADO DE ACUERDO A LOS REQUERIMIENTOS DEL ESTUDIANTE Y DEL ADMINISTRADOR

Figura 3.15

Diagrama de Base de Datos para la gestión de calificaciones



Nota. Diagrama de la base de datos diseñado para la administración de calificaciones de la institución, considerando todas las gestiones, carreras y cursos en los que el estudiante puede estar involucrado. Cabe mencionar que las calificaciones se editan en función de los 4 bimestres disponibles de la gestión. Además, se incluye la nota del segundo turno (2T) para el estudiantado, complementando así los requerimientos establecidos por el Ministerio de Gobierno.

- **TAREA 5.2. IMPLEMENTAR UN APARTADO WEB PARA LA GESTIÓN DE LAS CALIFICACIONES DE PARTE DEL ADMINISTRADOR**

Figura 3.16

Interfaz web para la gestión de calificaciones

CI Estudiante	Nombre Completo	Matrícula	Carrera	Fecha de Inscripción	Acciones
83492715	Carlos Andres Mamani Lopez	CAML83492715	SISTEMAS INFORMÁTICOS	2024-09-25	Editar Nota
58192734	Ana Maria Fernandez Vargas	AMFV58192734	SISTEMAS INFORMÁTICOS	2024-09-25	Editar Nota
40728193	Miguel Angel Torres Morales	MATM40728193	SISTEMAS INFORMÁTICOS	2024-09-25	Editar Nota
72983415	Lucia Carolina Vargas Ortiz	LCVO72983415	SISTEMAS INFORMÁTICOS	2024-09-25	Editar Nota
36287495	Juan Pablo Ramirez Torres	JPR36287495	SISTEMAS INFORMÁTICOS	2024-09-25	Editar Nota
56198274	Elena Gabriela Castro Fernandez	EGCF56198274	SISTEMAS INFORMÁTICOS	2024-09-25	Editar Nota
11111111	pepito debora torres	PDT11111111	SISTEMAS INFORMÁTICOS	2024-09-26	Editar Nota

Nota. La gráfica muestra la búsqueda y el registro de un estudiante según su gestión, carrera y curso. El administrador puede editar la nota del estudiante en cada uno de los 4 bimestres de la gestión. Las notas están ponderadas sobre 100 puntos y el estudiante deberá obtener al menos 61 puntos para aprobar; de lo contrario, reprobará la materia. Estos datos se reflejan al momento de consultar las notas, ya sea para el usuario administrador, docente o estudiante.

- **TAREA 5.3. DESARROLLAR UN MÓDULO WEB PARA LA VISUALIZACIÓN DE NOTAS DE MANERA GRÁFICA Y DESCRIPTIVA TANTO PARA EL ESTUDIANTE COMO PARA EL ADMINISTRADOR**

Figura 3.17

Interfaz web para la visualización de notas de parte del estudiante

Instituto Tecnológico - Estudiante								
Historial Académico del Estudiante								
Datos Personales								
Nombres: Lucía Carolina Vargas Ortiz			Carnet de Identidad: 72983415			Celular: 77281233		
Calificaciones								
Gestión	Código	Curso	1er Bimestre	2do Bimestre	3er Bimestre	4to Bimestre	Nota Promedio	Estado
GESTIÓN 2026	PRG-102	PROGRAMACIÓN I	70	50	41	15	44	Aprobado
GESTIÓN 2025	PRG-102	PROGRAMACIÓN I	70	50	41	15	44	Aprobado
GESTIÓN 2024	PRG-102	PROGRAMACIÓN I	70	50	41	15	44	Aprobado

Nota. Al ingresar con la cuenta de usuario estudiante, se podrá revisar todo el rendimiento académico y ver las notas de todas las materias. Además, se podrá visualizar si se aprobó o no una materia en una gestión determinada o en todas las materias, permitiendo al estudiante evaluar su rendimiento académico a lo largo de su carrera. Por último, cabe mencionar que estos datos se pueden exportar en formato PDF para una mejor comodidad del estudiante.

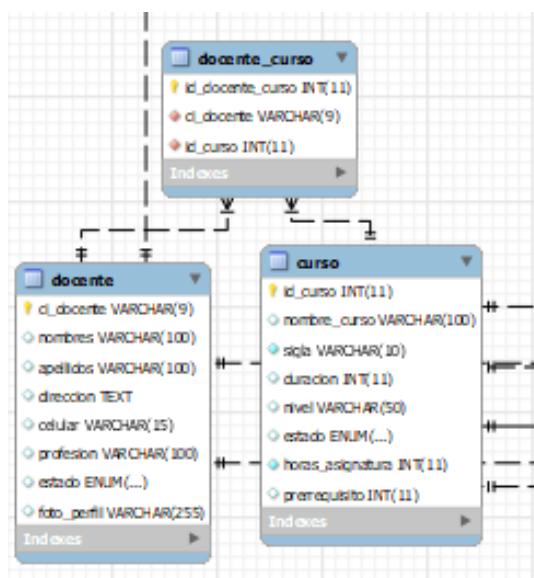
3.4.3. TERCERA ITERACIÓN

VI. HISTORIA DE USUARIO 6: REGISTRO DE DOCENTES

- TAREA 6.1. DISEÑAR LA BASE DE DATOS PARA EL REGISTRO DE DOCENTES

Figura 3.18

Diseño de Base de Datos para el Registro de Docentes



Nota. Diagrama de la base de datos creado para gestionar a los docentes de la institución, considerando los cursos que estos pueden impartir a los estudiantes. Es importante señalar que los cursos de los docentes están vinculados directamente con la gestión actual, lo que facilita una mejor organización entre docentes y cursos.

- **TAREA 6.2. IMPLEMENTAR UN APARTADO WEB PARA EL REGISTRO DE DOCENTES**

Figura 3.19

Interfaz web para el registro de docentes

Nota. Interfaz web para el registro de docentes considerando todas las materias que se pueden impartir en la institución, los datos requeridos por el docente son los más esenciales sin tomar en cuenta aspectos económicos como ser su salario.


• TAREA 6.2. IMPLEMENTAR UN MÓDULO WEB PARA LA ADMINISTRACIÓN DE DOCENTES

Figura 3.20

Interfaz web para la administración de docentes

Listar Docentes

+ Registrar Nuevo Docente Exportar PDF

Docentes Registrados							
Foto de Perfil	Carnet de Identidad	Nombre Completo	Celular	Profesión	Estado	Cursos que Imparte	Acciones
	612312	Camila Mamani Choque	77291202	Lic. en Gastronomía	Activo	INOCUIDAD ALIMENTARIA, SEGURIDAD OCUPACIONAL Y MEDIO AMBIENTE, INGLÉS, TÉCNICAS CULINARIAS E HISTORIA DE LA GASTRONOMÍA, PROCEDIMIENTOS BÁSICOS DE REPOSTERÍA, PASTERÍA Y PANADERÍA, INFORMÁTICA APLICADA A LA GASTRONOMÍA, NUTRICIÓN Y DIETÉTICA, ORGANIZACIÓN Y GESTIÓN DE ALIMENTOS Y BEBIDAS, SERVICIO DE COMEDOR, ETIQUETA Y PROTOCOLO, FRANCES TÉCNICO, GASTRONOMÍA NACIONAL, PANADERÍA, EMPRENDIMIENTO PRODUCTIVO - GASTRONOMÍA, COCINA CREATIVA, BAR Y COCTELERÍA, ENOLOGÍA Y MARIDAJE, EVENTOS, BANQUETES Y GARNISH, MARKETING GASTRONÓMICO, INDUSTRIA Y ESPECIALIDADES GASTRONÓMICAS, TALLER DE MODALIDAD DE GRADUACIÓN, REPOSTERÍA Y PASTERÍA, GASTRONOMÍA INTERNACIONAL	Editar Eliminar
	9128821	María Loza Fernandez	77722312	Lic. en Ingeniería de Sistemas	Activo	INGLÉS TÉCNICO, MATEMÁTICA PARA LA INFORMÁTICA, PROGRAMACIÓN I, HARDWARE DE COMPUTADORAS, TALLER DE SISTEMAS OPERATIVOS, OFIMÁTICA Y TECNOLOGÍA MULTIMEDIA	Editar Eliminar
	9144385	Alejandro Magne Loza	71293120	Lic. en Informática	Activo	INGLÉS TÉCNICO, MATEMÁTICA PARA LA INFORMÁTICA, PROGRAMACIÓN I, HARDWARE DE COMPUTADORAS, TALLER DE SISTEMAS OPERATIVOS, OFIMÁTICA Y TECNOLOGÍA MULTIMEDIA, DISEÑO Y PROGRAMACIÓN WEB I, ESTADÍSTICA, PROGRAMACIÓN II, ESTRUCTURA DE DATOS, REDES DE COMPUTADORAS I, PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES, DISEÑO Y PROGRAMACIÓN WEB II, BASE DE DATOS I, ANÁLISIS Y DISEÑO DE SISTEMAS I, EMPRENDIMIENTO PRODUCTIVO - SISTEMAS INFORMÁTICOS, DISEÑO Y PROGRAMACIÓN WEB III, GESTIÓN Y MEJORAMIENTO DE LA CALIDAD DE SOFTWARE, REDES DE COMPUTADORAS II, TALLER DE MODALIDAD DE GRADUACIÓN, ANÁLISIS Y DISEÑO DE SISTEMAS II, PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES II, BASE DE DATOS II	Editar Eliminar

Nota. Listado de todos los docentes registrados en el sistema, se puede apreciar todas las carreras que imparte el docente y sus datos personales mas relevantes. Estos datos se pueden editar siendo usuario administrador.

3.4.4. CUARTA ITERACIÓN

VII. HISTORIA DE USUARIO 7: GENERACIÓN DE GRÁFICAS ADMINISTRATIVAS

• TAREA 7.1. IDENTIFICAR LOS REPORTES Y GRÁFICAS MÁS RELEVANTES A MOSTRAR PARA EL ADMINISTRADOR

La identificación de los reportes y gráficas más relevantes para el administrador implica un análisis detallado orientado a comprender y satisfacer las necesidades informativas clave que faciliten una gestión académica eficiente.

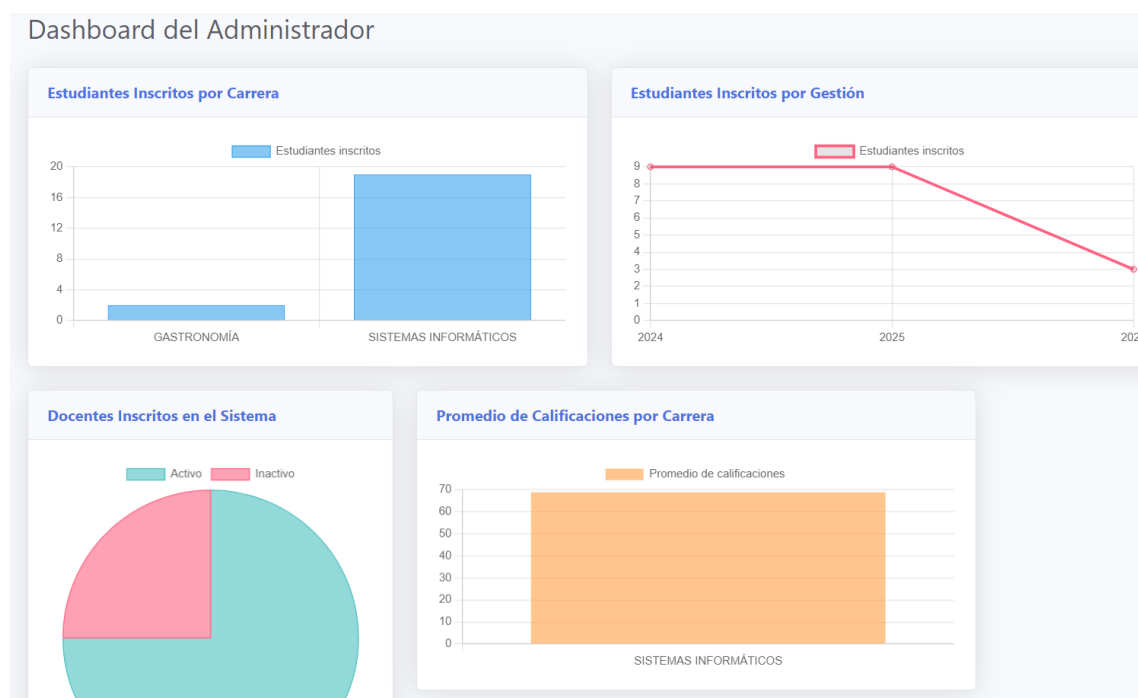
Este proceso requiere una consulta con los administradores para captar las métricas y datos esenciales que son críticos en su labor diaria. Por ende, datos como los estudiantes inscritos por carrera, por gestión, como también los docentes que están

registrados en el sistema, por último, el promedio de calificaciones por carreras son algunos de los más relevantes a mostrar.

- **TAREA 7.2. IMPLEMENTAR DIFERENTES REPORTES Y GRÁFICAS ACADÉMICAS QUE PERMITAN AL ADMINISTRADOR VISUALIZAR EL RENDIMIENTO DE LOS ESTUDIANTES A TRAVÉS DE LOS CURSOS**

Figura 3.21

Diferentes gráficas administrativas



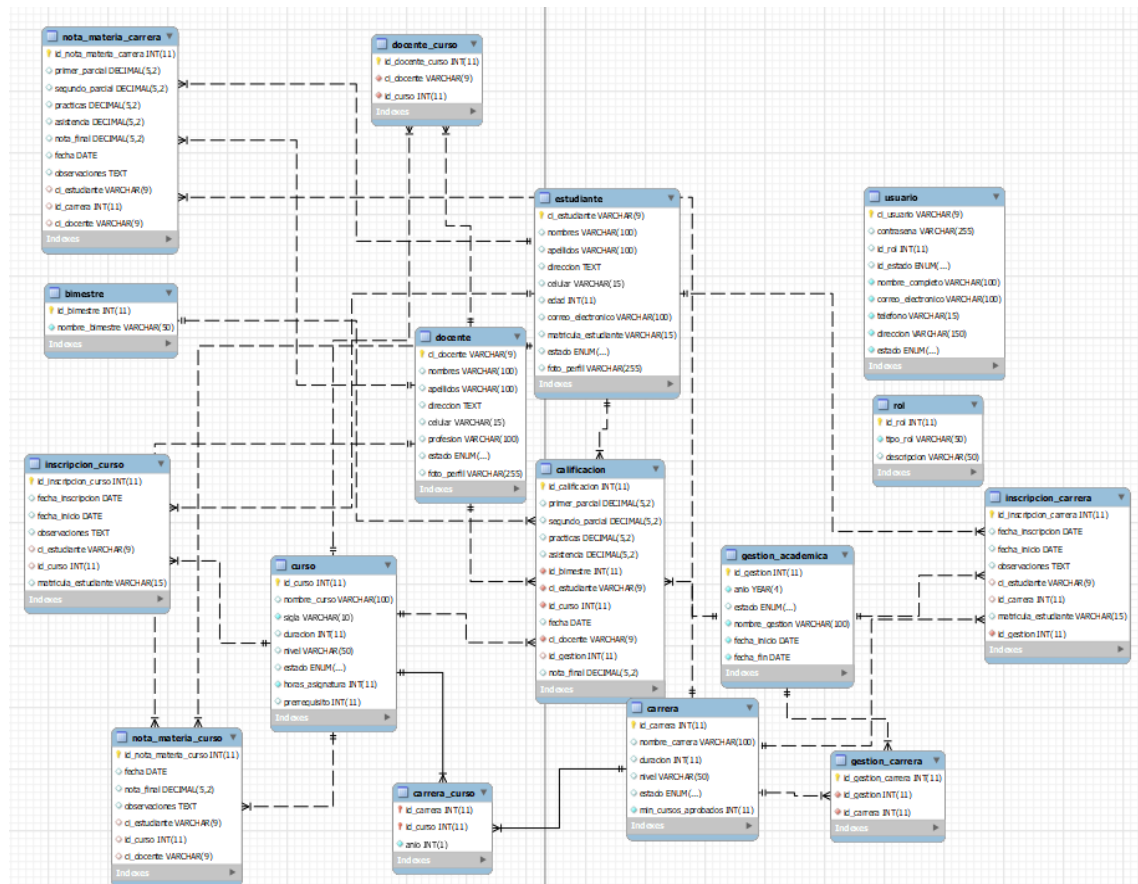
Nota. En el panel principal se muestran diferentes gráficas que permiten visualizar el rendimiento académico y administrativo de los estudiantes. De este modo, se puede analizar el desempeño individual y colectivo de los estudiantes en cada materia y gestionar los recursos académicos de manera más eficiente. Además, estas gráficas facilitan la identificación de áreas donde los estudiantes presentan dificultades, lo cual permite a los administradores y docentes tomar decisiones informadas para mejorar la calidad educativa. También se pueden comparar los resultados a lo largo de diferentes gestiones para evaluar tendencias y planificar intervenciones específicas.

3.5. FASE III - CODIFICACIÓN

En esta fase de desarrollo, se llevará a cabo la codificación de cada iteración, considerando las agrupaciones correspondientes de historias de usuario y aplicando los criterios de aceptación. Para facilitar el proceso y mejorar la comprensión durante el desarrollo del sistema, se utilizarán diferentes diagramas.

Figura 3.22

Diagrama General de Base de Datos



Nota. Diagrama general de base de datos del proyecto que involucra todas las funcionalidades para los usuarios.

3.6. FASE IV - PRUEBAS

3.6.1. CRITERIOS DE ACEPTACIÓN

Tabla 3.20

Criterio de Aceptación: HU-01

CRITERIO DE ACEPTACIÓN – HU--01	
NOMBRE DE HISTORIA: CONTROL DE ACCESO DE USUARIOS	
NUMERO: 01	USUARIO: ADMINISTRADOR
DESCRIPCIÓN: <ul style="list-style-type: none"> ❖ Registro de Usuarios: El administrador debe poder registrar nuevos usuarios (estudiantes, docentes, administradores) con sus respectivos roles y permisos. ❖ Validación de Datos: El sistema debe validar la entrada de datos, asegurando que no se registren usuarios con datos incompletos o duplicados. ❖ Autenticación Segura: La interfaz de autenticación permite que solo usuarios registrados puedan acceder al sistema. ❖ Actualización y Eliminación: El administrador debe poder actualizar y eliminar perfiles de usuarios existentes. 	
EVALUACIÓN DE LA PRUEBA: La prueba se concluyó satisfactoriamente.	

Nota. Primer criterio de aceptación para el control de acceso de usuarios

Tabla 3.21

Criterio de Aceptación: HU-02

CRITERIO DE ACEPTACIÓN – HU-02	
NOMBRE DE HISTORIA: ADMINISTRACIÓN DE GESTIONES Y CARRERAS	
NUMERO: 02	USUARIO: ADMINISTRADOR
DESCRIPCIÓN:	

-
- ❖ **Administración de Gestiones:** El administrador crea, modifica y elimina gestiones de acuerdo a la disponibilidad.
 - ❖ **Administración de Carreras:** El administrador crea, modifica y elimina carreras de acuerdo a la gestión actual.
 - ❖ **Interfaz Intuitiva:** La interfaz para administrar las gestiones y carreras es intuitiva y permite realizar estas operaciones de manera eficiente.
-

EVALUACIÓN DE LA PRUEBA: La prueba se concluyó satisfactoriamente.

Nota. Segundo criterio de aceptación para la administración de gestiones y carreras

Tabla 3.22

Criterio de Aceptación: HU-03

CRITERIO DE ACEPTACIÓN – HU-03	
NOMBRE DE HISTORIA: ADMINISTRACIÓN DE CURSOS	
NUMERO: 03	USUARIO: ADMINISTRADOR
DESCRIPCIÓN:	
<ul style="list-style-type: none"> ❖ Gestión de Cursos: El administrador debe poder crear, modificar y eliminar cursos y asignar docentes a cada curso. ❖ Interfaz Intuitiva: La interfaz para administrar cursos debe ser intuitiva y permitir realizar estas operaciones de manera eficiente. ❖ Base de Datos: Los cursos respecto a la carrera y asignaciones deben almacenarse correctamente en la base de datos. 	
EVALUACIÓN DE LA PRUEBA: La prueba se concluyó satisfactoriamente.	

Nota. Tercer criterio de aceptación para la administración de cursos

Tabla 3.23*Criterio de Aceptación: HU-04*

CRITERIO DE ACEPTACIÓN – HU-04	
NOMBRE DE HISTORIA: ADMINISTRACIÓN DE ESTUDIANTES	
NUMERO: 04	NUMERO: 04
DESCRIPCIÓN: <ul style="list-style-type: none"> ❖ Gestión de Estudiantes: El administrador debe poder registrar, modificar y eliminar información de estudiantes. ❖ Asociación a Cursos: Los estudiantes deben poder ser inscritos en los cursos correspondientes a través de la interfaz. ❖ Actualización de Información: La información del perfil del estudiante, como datos personales y académicos, debe poder actualizarse y mantenerse coherente. ❖ Historial Académico: El sistema debe permitir al administrador revisar y gestionar el historial académico de los estudiantes. 	
EVALUACIÓN DE LA PRUEBA: La prueba se concluyó satisfactoriamente.	

Nota. Cuarto criterio de aceptación para la administración de estudiantes y sus requerimientos

Tabla 3.24*Criterio de Aceptación: HU-05*

CRITERIO DE ACEPTACIÓN – HU-05	
NOMBRE DE HISTORIA: ADMINISTRACIÓN DE CALIFICACIONES	
NUMERO: 05	NUMERO: 05
DESCRIPCIÓN: <ul style="list-style-type: none"> ❖ Visualización de Calificaciones: El estudiante debe poder acceder a su perfil y visualizar sus calificaciones e historial académico. 	

-
- ❖ **Gestión de Calificaciones:** El administrador o docente debe poder ingresar y modificar las calificaciones de los estudiantes.
 - ❖ **Visualización Gráfica:** El sistema debe permitir la visualización de calificaciones en formatos gráficos y descriptivos.
 - ❖ **Base de Datos:** Las calificaciones deben ser almacenadas de manera segura y confiable en la base de datos.
-

EVALUACIÓN DE LA PRUEBA: La prueba se concluyó satisfactoriamente.

Nota. Quinto criterio de aceptación para la administración de calificaciones de los estudiantes

Tabla 3.25

Criterio de Aceptación: HU-06

CRITERIO DE ACEPTACIÓN – HU06	
NOMBRE DE HISTORIA: ADMINISTRACIÓN DE DOCENTES	
NUMERO: 06	USUARIO: ADMINISTRADOR
DESCRIPCIÓN:	
<ul style="list-style-type: none"> ❖ Registro de calificaciones: El docente puede registrar una nueva calificación de acuerdo al curso que dicta. ❖ Asignación de Cursos: Los docentes deben poder ser asignados a los cursos correspondientes mediante la interfaz. ❖ Validación de Datos: El sistema debe validar la entrada de datos para evitar duplicados o errores en el registro de docentes. ❖ Base de Datos: La información de los docentes debe almacenarse correctamente en la base de datos. 	
EVALUACIÓN DE LA PRUEBA: La prueba se concluyó satisfactoriamente.	

Nota. Sexto criterio de aceptación para la administración de docentes

Tabla 3.26*Criterio de Aceptación: HU-07*

CRITERIO DE ACEPTACIÓN – HU-07	
NOMBRE DE HISTORIA: GENERACIÓN DE GRÁFICAS ADMINISTRATIVAS	
NUMERO: 07	USUARIO: ADMINISTRADOR
DESCRIPCIÓN: <ul style="list-style-type: none"> ❖ Visualización de Gráficas: El sistema debe permitir la visualización de gráficos y estadísticas relevantes para facilitar la toma de decisiones. ❖ Interfaz de Reportes: La interfaz debe ser fácil de usar y permitir la selección de diferentes tipos de reportes y gráficas. ❖ Exportación de Datos: Los reportes generados deben poder ser exportados en formatos como PDF o Excel para su análisis externo. 	
EVALUACIÓN DE LA PRUEBA: La prueba se concluyó satisfactoriamente.	

Nota. Séptimo criterio de aceptación para la generación de gráficas administrativas

4. CUARTO CAPÍTULO

MARCO EVALUATIVO

En este capítulo se llevará a cabo la verificación de la calidad del proyecto, evaluando las características del software conforme a la norma ISO 9126. Asimismo, se analizarán los costos asociados al desarrollo del sistema, empleando el modelo COCOMO II para su estimación.

4.1. Métricas de Calidad

El principal propósito de la ingeniería de software es desarrollar un sistema, aplicación o producto de gran calidad. Para alcanzar este fin, los ingenieros de software deben utilizar métodos eficientes junto con herramientas contemporáneas, todo ello dentro de un proceso de desarrollo de software bien establecido. A continuación, se presentarán una serie de métricas de software que permiten evaluar cuantitativamente la calidad del software.

4.1.1. Usabilidad

La usabilidad se refiere a un conjunto de atributos que permiten evaluar el esfuerzo que el usuario debe invertir al interactuar con el sistema. Esto implica realizar una serie de preguntas que determinan lo sencillo que es para los usuarios aprender y manejar el sistema.

En la tabla siguiente se reflejan estos criterios en niveles porcentuales alcanzados por el sistema en términos de su comprensibilidad para el usuario, y posteriormente se presenta el porcentaje final de usabilidad. Se llevaron a cabo encuestas a 12 diferentes usuarios finales para medir la facilidad de uso, comprensión y aprendizaje del sistema, tal como se muestra en la tabla adjunta.

Tabla 4.1*Encuesta realizada a diferentes usuarios*

PREGUNTAS	RESPUESTAS		PORCENTAJES
	SI	NO	
1. ¿El acceso al sistema es complicado?	1	9	90%
2. ¿Son comprensibles las respuestas del sistema?	2	8	80%
3. ¿Son complicados los procesos que realiza el sistema?	2	8	80%
4. ¿El sistema tiene interfaces entendibles?	1	9	90%
5. ¿La interfaz del sistema es agradable a la vista?	0	10	100%
6. ¿Son satisfactorias las respuestas que el sistema devuelve?	2	8	80%
7. ¿El sistema reduce su tiempo de trabajo?	2	8	80%
8. ¿Es difícil aprender a manejar el sistema?	3	7	70%
9. ¿El sistema satisface las necesidades que usted requiere?	2	8	80%
10. ¿Utiliza el sistema con facilidad?	3	7	70%
TOTAL			82%

Nota. Se llevó a cabo una encuesta a 12 usuarios diferentes, entre los que se incluyen administradores, docentes y estudiantes del instituto. Las encuestas fueron realizadas de manera presencial, utilizando formatos en papel y de entrevista.

La tabla anterior indica que el sistema posee una usabilidad del 82%, lo que implica que, de cada 100 personas, 82 consideran que el sistema es fácil de usar y comprensible.

$$Usabilidad = 82\%$$

4.1.2. Funcionalidad

La funcionalidad es la capacidad del software de proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizando en ocasiones específicas, este atributo del sistema no puede medirse de forma directa, por esa razón para el cálculo de la adecuación funcional utilizaremos la métrica de punto función, para esto se debe determinar cinco características de dominios de información.

La ecuación de Punto Función se define bajo la siguiente formula:

$$PF = PF_{No\ Ajustado} \times (0.65 + 0.01 \times \sum F_i) \quad (4.1)$$

Donde:

- PF es la Medida de adecuación funcional
- $PF_{No\ Ajustado}$ es la Sumatoria de diferentes datos
- 0.65 es el número de confiabilidad del proyecto
- 0.01 es el error mínimo aceptable de complejidad
- $\sum F_i$ son los valores de ajuste de complejidad

Primeramente, para calcular el Punto Función se debe trabajar en sus variables que conforma, para ello se empieza con el punto función no ajustado.

$$PF_{No\ Ajustado} = U + I + E + Q + L \quad (4.2)$$

Donde:

- U es el Número de Entradas Externas (User Inputs)
- I es el Número de Salidas Externas (User Outputs)
- E es el Número de Consultas (User Inquiries)
- Q es el Número de Archivos Lógicos Internos (Internal Logical Files)
- L es el Número de Archivos de Interfaz Externa (External Interface Files)

La anterior formula permitirá conocer la cuenta total, o bien el punto función no ajustado, para obtener el dato de punto función.

Tabla 4.2

Conteo Total para Punto Función

PARÁMETROS DE MEDIDA	CANTIDAD	FACTORES DE PONDERACIÓN	TOTAL
N° de entradas de usuario	24	4	96
N° de salidas de usuario	21	7	147
N° de peticiones de usuario	17	4	68
N° de archivos	22	10	220
N° de interfaces externas	1	7	7
Cuenta Total			538

Nota. Se determinaron los factores de ponderación en función de la relevancia asignada a cada parámetro de medida del proyecto. El total acumulado fue de 538, el cual se utilizará para calcular el valor del Punto Función.

Los datos obtenidos en la anterior tabla representan el conteo total de los parámetros de entradas, salidas, peticiones, archivos e interfaces externas. Esta información permitirá ajustar los valores según las características ambientales del sistema, para reflejar con mayor precisión el nivel de complejidad asociado a los requerimientos específicos del proyecto, tales como la necesidad de copias de seguridad fiables, la comunicación de datos especializada y el rendimiento crítico.

Por otro lado, para obtener los valores de ajuste de complejidad, las respuestas a las preguntas de la siguiente tabla asignan un nivel de importancia que contribuye al valor total de ajuste, afectando el cálculo del punto de función

Tabla 4.3*Valores de ajuste de complejidad*

IMPORTANCIA		0	1	2	3	4	5	
ESCALA		SIN IMPORTANCIA	INCREMENTAL	MODERADO	MEDIO	SIGNIFICATIVO	ESENCIAL	SUBTOTAL
1.	¿Requiere el sistema copias de seguridad y de recuperación fiable?						X	5
2.	¿Se requiere comunicación de datos especializadas para transferir información a la aplicación u obtenerlas de ellas?					X		4
3.	¿Es crítico el rendimiento?					X		4
4.	¿El sistema web será ejecutado en el S.O. actual?					X		4
5.	¿Se requiere una entrada interactiva para el sistema?						X	5
6.	¿Se actualizan los archivos de forma interactiva?						X	5
7.	¿Son complejas las entradas, salidas, los archivos o las peticiones?					X		4
8.	¿Es complejo el procesamiento interno del sistema?					X		4
9.	¿Se ha diseñado el código para ser reutilizado?						X	5
10.	¿Están incluidas en el diseño la conversión y la instalación?				X			3

11. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	X	3
12. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	X	4
TOTAL		50

Nota. Se formularon 12 preguntas que permiten evaluar si el proyecto satisface las expectativas requeridas, con el objetivo de obtener los valores de ajuste de complejidad correspondientes.

Una vez obtenido todos los datos se puede obtener el valor de Punto Función

$$PF_{obtenida} = PF \text{ No Ajustado} \times (0.65 + 0.01 \times \sum F_i)$$

$$PF_{obtenida} = 538 \times (0.65 + 0.01 \times 50)$$

$$PF_{obtenida} = 618.7$$

Ahora se realizará el cálculo de Punto Función de manera ideal, considerando que los valores de ajuste de complejidad están en un puntaje ideal.

$$PF_{ideal} = PF \text{ No Ajustado} \times (0.65 + 0.01 \times \sum F_i)$$

$$PF_{ideal} = 538 \times (0.65 + 0.01 \times 60)$$

$$PF_{ideal} = 672.5$$

Entonces se obtiene el Punto Función de manera es:

$$Funcionalidad = \frac{PF_{obtenida}}{PF_{ideal}} \times 100\%$$

$$Funcionalidad = \frac{618.7}{672.5} \times 100\%$$

$$Funcionalidad = 92\%$$

Por lo tanto, quiere decir que el proyecto satisface las necesidades explícitas con un valor del 92% de aprobación.

4.1.3. Confiabilidad

La confiabilidad permite analizar la relación entre el nivel de funcionalidad del software y los recursos utilizados. En otras palabras, refleja el tiempo durante el cual el software está disponible para su uso, y se calcula considerando la probabilidad de que el sistema experimente fallas.

Por lo tanto, la siguiente función muestra el nivel de confiabilidad del sistema, considerando las probabilidades de fallas conforme al tiempo:

Confiabilidad del Proyecto:

$$F_t = \text{Funcionalidad} \times e^{-\lambda t} \quad (4.3)$$

Donde:

- Funcionalidad es el Dato Obtenido sobre la Funcionalidad del Proyecto
- λ es la Tasa de Falla del Sistema
- t es el Tiempo de Operación

Probabilidad de no hallar una falla en un tiempo determinado “t”:

$$P(T > t) = 1 - F(t) \quad (4.4)$$

Posteriormente, se determina la confiabilidad del sistema reemplazando datos, donde la tasa de falta del sistema es de 2 errores por cada hora, por ende, lambda tiene un valor de 0.02. Por otro lado, si se requiere que se realizará la confiabilidad para 12 meses de operación, entonces el tiempo de operación será ese valor.

$$F_t = \text{Funcionalidad} \times e^{-\lambda t}$$

$$F_t = 0.92 \times e^{-(0.01)(12)}$$

$$F_t = 0.92 \times 0.887$$

$$F_t = 0.816$$

Ahora se definen las fórmulas para obtener si existirá alguna falla a través del tiempo:

$$P(T > t) = 1 - F(t)$$

$$P(T > t) = 1 - 0.816 = 0.18$$

Como resultado se determina que la probabilidad que el sistema presente alguna falla es del 18%, dándonos así una confiabilidad del proyecto del 81%.

$$\textbf{Confiabilidad} = 81\%$$

4.1.4. Mantenibilidad

La mantenibilidad se refiere a los atributos que permiten evaluar el esfuerzo requerido para efectuar modificaciones en el software, ya sea para corregir errores o mejorar sus funcionalidades. Para medir la mantenibilidad del sistema, se emplea el Índice de Madurez del Software (IMS), el cual ofrece una indicación de la estabilidad de un producto de software.

Función IMS:

$$IMS = \frac{M_t - (F_C + F_A + F_E)}{M_t} \quad (4.5)$$

Donde:

- M_t es el Numero de módulos total de la versión actual
- F_C es el Número de módulos de la versión actual que se cambiaron
- F_A es el Número de módulos de la versión actual que se añadieron
- F_E es el Número de módulos de la versión anterior que se eliminaron en la versión

$$IMS = \frac{4 - (0 + 0 + 0)}{4} = 1$$

Por ende, el porcentaje de 100% significa que no requiere de mantenimiento inmediatamente.

$$\textbf{Mantenibilidad} = 100\%$$

4.1.5. Portabilidad

El sistema fue desarrollado utilizando PHP con una base de datos MySQL y se ejecuta en cualquier servidor web, sin necesidad de una instalación adicional. En cuanto a la plataforma, el sistema de información está diseñado para sistemas operativos de la familia Microsoft Windows. Además, se verificó su desempeño en los navegadores más utilizados, como Google Chrome, Firefox, Microsoft Edge, entre otros, obteniendo una calificación del 100% en portabilidad. Este resultado confirma que el sistema funciona de manera óptima en los distintos navegadores.

$$\textit{Portabilidad} = 100\%$$

4.1.6. Calidad del Proyecto

Finalmente se recolectan los datos de usabilidad, funcionalidad, confiabilidad, mantenibilidad y portabilidad para obtener la calidad del proyecto.

Tabla 4.4

Calidad del Proyecto

CARACTERÍSTICA	RESULTADO
Usabilidad	82%
Funcionalidad	92%
Confiabilidad	81%
Mantenibilidad	100%
Portabilidad	100%
Total	91%

Nota. De acuerdo a la tabla, se obtiene un total del 91 por ciento de calidad del proyecto lo que representa un alto valor evaluativo.

4.2. Costo de Desarrollo

4.2.1. Costo del Software Desarrollado

Para llevar a cabo el análisis de costos, se utilizó COCOMO II, un modelo matemático basado en datos empíricos que se emplea para estimar los costos del desarrollo de software tal como se vio en el marco teórico. En este modelo de estimación se utilizan las siguientes ecuaciones.

El proyecto al tener un tamaño medio y con complejidad moderada, entonces se le considera semi-acoplado, por ende, los valores para este proyecto son $a = 3$ y $b = 1.12$. Por otro lado, para obtener KLDC se convierte los puntos función a miles de líneas de código, teniendo en cuenta que el factor LDC por PF para el lenguaje de programación PHP es de 12 unidades.

$$KLDC = \frac{(PF \times \text{Lineas de código por cada PF})}{1000} \quad (4.6)$$

$$KLDC = \frac{(618.7 \times 12)}{1000}$$

$$KLDC = 7.42$$

Para obtener el factor multiplicativo $M(x)$ según COCOMO II, es necesario evaluar aspectos como el riesgo, la complejidad, la calidad requerida, el nivel de experiencia del equipo y las herramientas utilizadas. Los valores asignados deben reflejar la importancia y evaluación del proyecto. Para este proyecto, se establecieron los siguientes datos:

$$M(x) = RELY \times CPLX \times TIME \times ACAP \times PCAP \times TOOL \times MODP \quad (4.7)$$

$$M(x) = 1.15 \times 1.00 \times 1.00 \times 0.80 \times 0.70 \times 1.00 \times 1.00$$

$$M(x) = 0.644$$

Entonces, teniendo todos los datos del proyecto, se obtiene la estimación del esfuerzo de acuerdo con la Ecuación 2.1:

$$E = a \times (KLDC)^b \times m(X)$$

$$E = 3.0 \times (7.42)^{1.12} \times 0.644$$

$$E = 3.0 \times (7.42)^{1.12} \times 0.644$$

$$E = 18.23 \text{ [personas/mes]}$$

Calculando el tiempo de duración de desarrollo D, se trabajará con los parámetros de semi-acoplado con $c=2.5$ y $d=0.35$:

$$D = c \times E \times d$$

$$D = 2.5 \times 18.23 \times 0.35$$

$$D = 15.95 \text{ [meses]} \cong 16 \text{ [meses]}$$

Ahora para calcular el personal requerido se tiene:

$$P = \frac{E}{D} = \frac{18.23}{15.95} = 1.14$$

El salario promedio de un programador varía entre los 2000 a 4000 Bolivianos, para nuestro proyecto se tomará un valor promedio de 3000 Bolivianos, con el fin de calcular el costo del software desarrollado.

Costo del Software por Persona:

$$Costo_{software-persona} = Nro.de Programadores \times Salario Programador$$

$$Costo_{software-persona} = 1 \times 2250 \text{ [Bs]}$$

$$Costo_{software-persona} = 2250 \text{ [Bs]}$$

Costo Total de Desarrollo:

$$Costo_{Total} = Costo_{software-persona} \times Numero de Meses$$

$$Costo_{Total} = 2250 \times 16 \text{ [Bs]}$$

$$Costo_{Total} = 2250 \times 16 \text{ [Bs]}$$

$$Costo_{Total} = 36000 \text{ [Bs]} \cong 5194.8 \text{ [USD]}$$

5. QUINTO CAPÍTULO

MARCO CONCLUSIVO

5.1. CONCLUSIONES

Una vez finalizado el proyecto de grado, se ha logrado alcanzar el objetivo principal planteado, satisfaciendo las necesidades de la institución, así como cumpliendo con los siguientes puntos:

- El desarrollo del sistema web de gestión académica para el Instituto Tecnológico del Occidente ha mejorado significativamente la eficiencia operativa y la gestión académica.
- Este sistema ha sustituido procesos manuales y administrativos, permitiendo a usuarios (administradores, docentes y estudiantes) acceder a información de manera más ágil y precisa.
- Los usuarios han expresado satisfacción debido a la reducción de tareas redundantes y el acceso fácil a información relevante como calificaciones y reportes.
- Se utilizó el estándar ISO/IEC 25010 para garantizar la calidad del software en funcionalidad, fiabilidad, usabilidad, mantenibilidad, portabilidad y eficiencia, desde el diseño hasta la implementación.
- La implementación de Extreme Programming (XP) facilitó un desarrollo iterativo y colaborativo, promoviendo ajustes rápidos según las necesidades de los usuarios.
- Usando el modelo COCOMO II, el costo total del desarrollo se estimó en \$5195 USD, con un esfuerzo de 18 meses por persona.
- El sistema permite la centralización de datos académicos en una base de datos relacional, mejorando la toma de decisiones y reduciendo errores.
- El diseño responsivo del sistema garantiza acceso desde diversos dispositivos, mejorando la experiencia del usuario y adaptándose a un número creciente de usuarios y funciones.
- Las diferentes gráficas y reportes acerca del rendimiento estudiantil permiten a los administradores visualizar de manera más organizada y concisa la

información académica, facilitando la toma de decisiones y la implementación de estrategias de mejora para el desempeño de los estudiantes.

5.2. RECOMENDACIONES

- Es fundamental ofrecer capacitación constante al personal sobre el uso del sistema, complementada con un manual detallado.
- Se recomienda realizar evaluaciones de eficiencia y usabilidad mediante encuestas y análisis de rendimiento.
- Considerar la expansión del sistema para incluir la gestión de asistencia, planificación de horarios y registro de actividades extracurriculares.
- Añadir funcionalidades para la gestión financiera y asistencial, como pagos de matrículas y registro de asistencia.
- Implementar mecanismos avanzados de seguridad, incluyendo autenticación de dos factores (2FA) y cifrado de datos.
- Considerar la migración a una infraestructura en la nube para mejorar la escalabilidad y disponibilidad del sistema a largo plazo.
- Después de un año de implementación, realizar una evaluación del impacto en el rendimiento académico y la eficiencia del personal administrativo para identificar áreas de mejora.

6. SEXTO CAPÍTULO

MARCO BIBLIOGRÁFICO

6.1. BIBLIOGRAFÍA

Aspe Alanoca, C. (2020). Sistema Web de Gestión Académica. Caso: Unidad Educativa "Inicua" (Tesis de pregrado). Universidad Pública de El Alto.

Beck, K., et al. (2001). Manifiesto ágil para el desarrollo de software. Recuperado de <https://agilemanifesto.org/>

Bennett, S., McRobb, S., & Farmer, R. (2010). Análisis y diseño de sistemas orientados a objetos. (4.a ed.). McGraw-Hill.

Berners-Lee, T., & Fischetti, M. (2019). Tejiendo la red: El diseño original y el destino final de la World Wide Web según su creador.

Boehm, B. W. (2000). Economía de la ingeniería de software. Prentice Hall.

Booch, G. (2007). Análisis y diseño orientado a objetos con aplicaciones (3.a ed.). Addison-Wesley.

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). Guía del usuario del lenguaje de modelado unificado (2.a ed.). Addison-Wesley.

Codd, E. F. (1970). A relational model of data for large shared data banks. ACM Computing Surveys, 2(1), 377-387.

Date, C. J. (2004). Introducción a los sistemas de bases de datos (8.a ed.). Addison-Wesley.

Elmasri, R., & Navathe, S. B. (2015). Fundamentos de sistemas de bases de datos. (7.a ed.). Pearson.

Fowler, M. (2004). UML Gota a Gota: Guía breve del lenguaje de modelado de objetos estándar (3.a ed.). Addison-Wesley.

Fowler, M., & Scott, K. (1997). UML: El lenguaje de modelado unificado (1.a ed.). Addison-Wesley.

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). El lenguaje de modelado unificado: Guía del usuario. Addison-Wesley.

Kendall, K. E., & Kendall, J. E. (2019). Análisis y Diseño de Sistemas (9.a ed.). Pearson Educación.

Kroenke, D. M., & Auer, D. J. (2021). Conceptos de Base de Datos. Pearson.

Marcotte, E. (2011). Diseño web adaptable. Smashing Magazine.

McLaughlin, B. (2019). Microservicios: Diseñando sistemas de grano fino. O'Reilly Media.

O'Reilly Media. (2020). Web Security for Developers: Real Threats, Practical Defense. O'Reilly Media.

Pressman, R. S., & Maxim, B. R. (2019). Ingeniería del software: Un enfoque práctico. McGraw-Hill Education.

Rubin, K. S. (2017). Scrum esencial: Una guía práctica para el proceso ágil más popular. Addison-Wesley.

Salazar Aviles, J. J. (2020). Análisis, diseño e implementación de un sistema académico para el control de notas, tareas y proceso de matriculación en la Unidad Educativa Guayas y Quil (Tesis de pregrado). Universidad de Guayaquil.

Schwaber, K., & Sutherland, J. (2020). Guía de Scrum. Recuperado de <https://scrumguides.org>

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). Conceptos de sistemas de bases de datos (6.a ed.). McGraw-Hill.

Sommerville, I. (2020). Ingeniería de Software. Pearson.

Stallings, W. (2011). Criptografía y seguridad en redes: Principios y práctica (5ta ed.). Pearson Educación.

Schmid, K. (2017). Análisis costo-beneficio en ingeniería de software: Una guía. Wiley.