



SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

INSTITUTO TECNOLÓGICO DE MINATITLÁN

INGENIERÍA EN SISTEMAS COMPUTACIONALES

“MANUAL DE PRÁCTICAS “

MATERIA

INGENIERIA DE SOFTWARE



MINATITLÁN, VER. JUNIO DEL 2023

3.2 ÍNDICE DEL MANUAL DE PRÁCTICAS

3.1 PORTADA DEL MANUAL DE PRACTICAS	1
3.2 ÍNDICE DEL MANUAL DE PRÁCTICAS.....	2
3.1 INTRODUCCIÓN	5
3.2 JUSTIFICACIÓN.....	6
3.3 OBJETIVO GENERAL DEL MANUAL DE PRÁCTICAS	7
3.4 DESARROLLO	8
3.4.1 Práctica 1 Elaboración y revisión del anteproyecto del sistema de desarrollo.....	8
3.4.1.1 Objetivo	8
3.4.1.2 Introducción	8
3.4.1.3 Correlación Los Temas Y Subtemas Del Programa De Estudio Vigente.	9
3.4.1.4 Material Y Equipo Necesario	9
3.4.1.5 Metodología	9
3.4.1.5 Sugerencias Didácticas	10
3.4.1.6 Reporte Del Alumno	11
3.4.2 Práctica 2 Con la ayuda de una herramienta CASE elabora el análisis del modelo de negocio seleccionado, considerando el modelo de requisitos, casos de uso, documentación de casos de uso y modelo de dominio.	12
3.4.2.1 Objetivo	12
3.4.2.2 Introducción	12
3.4.2.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.	12
3.4.2.4 Material Y Equipo Necesario	13
3.4.2.5 Metodología	13
3.4.2.6 Sugerencias Didácticas	14
3.4.2.7 Reporte Del Alumno	15
3.4.3 Práctica 3 Elaborar un estudio de factibilidad y el costo-beneficio aplicado a la organización	16
3.4.3.1 Objetivo	16
3.4.3.2 Introducción	16
3.4.3.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.	16
3.4.3.4 Material Y Equipo Necesario	16
3.4.3.5 Metodología	17
3.4.3.6 Sugerencias Didácticas	18
3.4.3.7 Reporte Del Alumno	18

3.4.4 Práctica 4 Establecer un diseño preliminar de las interfaces de usuario de acuerdo a los requisitos.....	20
3.4.4.1 Objetivo	20
3.4.4.2 Introducción	20
3.4.4.3 Especificar La Correlación Con El O Los Temas Y Subtemas Del Programa De Estudio Vigente.	20
3.4.4.4 Material Y Equipo Necesario	20
3.4.4.5 Metodología	21
3.4.4.6 Sugerencias Didácticas	21
3.4.4.7 Reporte Del Alumno	22
3.4.5 Práctica 5 Elaborar un diseño de bases de datos emanado del modelo entidad-relación	23
3.4.5.1 Objetivo	23
3.4.5.2 Introducción	23
3.4.5.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.	23
3.4.5.4 Material Y Equipo Necesario	23
3.4.5.5 Metodología	24
3.4.5.6 Sugerencias Didácticas	24
3.4.5.7 Reporte Del Alumno	25
3.4.5 Práctica 6 A partir del diccionario de datos y el diagrama E-R crear una base de datos 26	
3.4.6.1 Objetivo	26
3.4.6.2 Introducción	26
3.4.6.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.	26
3.4.6.4 Material Y Equipo Necesario	26
3.4.6.5 Metodología	27
3.4.6.6 Sugerencias Didácticas	27
3.4.6.7 Reporte Del Alumno	28
3.4.7 Práctica 7 Usando un lenguaje de programación establecer la conexión a una base de datos.....	29
3.4.7.1 Objetivo	29
3.4.7.2 Introducción	29
3.4.7.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.	29
3.4.7.4 Material Y Equipo Necesario	29
3.4.7.5 Metodología	30

3.4.7.6 Sugerencias Didácticas	30
3.4.7.7 Reporte Del Alumno	31
3.4.5 Práctica 8 Desarrollar los procesos identificados, asegurando las operaciones básicas de todo sistema: registro, actualización, consulta y estadística.....	33
3.4.8.1 Objetivo	33
3.4.8.2 Introducción	33
3.4.8.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.....	33
3.4.8.4 Material Y Equipo Necesario	33
3.4.8.5 Metodología	34
3.4.8.6 Sugerencias Didácticas	34
3.4.8.7 Reporte Del Alumno	35
3.4.9 Práctica 9 Probar el sistema con las técnicas existentes y validar que el modelo de requisitos esté atendido.....	36
3.4.9.1 Objetivo	36
3.4.9.2 Introducción	36
3.4.9.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.....	36
3.4.9.4 Material Y Equipo Necesario	36
3.4.9.5 Metodología	37
3.4.9.6 Sugerencias Didácticas	37
3.4.9.7 Reporte Del Alumno	38
3.4.10 Práctica 10 Implementar el sistema, capacitar a los usuarios y verificar la estabilidad del sistema para su liberación.....	39
3.4.10.1 Objetivo	39
3.4.10.2 Introducción	39
3.4.10.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.....	39
3.4.10.4 Material Y Equipo Necesario	39
3.4.10.5 Metodología	40
3.4.10.6 Sugerencias Didácticas	40
3.4.10.7 Reporte Del Alumno	41

3.1 INTRODUCCIÓN

La Ingeniería de Software, como disciplina fundamental en el desarrollo de sistemas y aplicaciones informáticas, desempeña un papel crucial en la resolución de desafíos tecnológicos y la satisfacción de las necesidades de la sociedad actual. En este contexto, la presente guía de prácticas se establece como un recurso esencial para estudiantes, profesionales y entusiastas del desarrollo de software que buscan adquirir y perfeccionar sus habilidades en esta disciplina fundamental.

La Ingeniería de Software es un campo en constante evolución que abarca desde la concepción y análisis de sistemas hasta la implementación y mantenimiento de soluciones software robustas y eficaces. Como resultado, el éxito en este campo exige una comprensión profunda de una amplia gama de conceptos y metodologías, así como la capacidad de aplicarlos de manera efectiva en proyectos prácticos.

Este manual de prácticas ha sido cuidadosamente diseñado para proporcionar una estructura educativa que abarca las principales áreas de la Ingeniería de Software. Cada sección de este manual se enfoca en aspectos específicos del proceso de desarrollo de software, desde la definición de requisitos hasta las pruebas y la implantación, con el objetivo de brindar a los estudiantes una comprensión sólida y práctica de los conceptos esenciales.

Cada práctica incluida en este manual se ha estructurado de manera clara y concisa, y se basa en principios reconocidos internacionalmente en la industria del software. Además, se han seleccionado ejemplos y ejercicios relevantes que reflejan situaciones comunes en el desarrollo de software, lo que permite a los estudiantes adquirir experiencia práctica aplicando las teorías y técnicas discutidas.

Esperamos que este manual de prácticas sirva como un recurso valioso para todos los interesados en la Ingeniería de Software, y que contribuya a su formación y crecimiento en esta disciplina en constante cambio.

A lo largo de esta guía, se presentarán prácticas detalladas, ejemplos y ejercicios diseñados para fomentar el pensamiento crítico, la resolución de problemas y el desarrollo de habilidades prácticas en el campo de la Ingeniería de Software. Le invitamos a aprovechar al máximo este recurso, y le deseamos éxito en su viaje hacia la comprensión y maestría de esta importante disciplina.

3.2 JUSTIFICACIÓN

La Ingeniería de Software es una disciplina crítica en la actualidad, que aborda el diseño, desarrollo y mantenimiento de sistemas de software en un mundo cada vez más digitalizado. La demanda de profesionales altamente capacitados en este campo es creciente, ya que las organizaciones buscan soluciones de software confiables y eficaces para satisfacer las necesidades de sus clientes y usuarios.

Este manual de prácticas tiene como objetivo proporcionar una guía estructurada y práctica para estudiantes y profesionales que deseen adquirir o mejorar sus habilidades en Ingeniería de Software. La justificación para la creación de este manual se fundamenta en los siguientes puntos:

1. **Formación Práctica Esencial:** La teoría es fundamental, pero la capacidad de aplicar ese conocimiento en la práctica es lo que distingue a un ingeniero de software competente. Este manual ofrece ejercicios y ejemplos prácticos que ayudarán a los estudiantes a desarrollar habilidades concretas y a comprender la aplicación real de los conceptos teóricos.
2. **Relevancia Profesional:** La Industria de Software está en constante evolución. Los profesionales deben estar preparados para afrontar los desafíos cambiantes y adoptar las mejores prácticas. Este manual se actualiza y se basa en estándares reconocidos a nivel internacional,

garantizando que los usuarios estén al tanto de las últimas tendencias y prácticas en Ingeniería de Software.

3. Adaptabilidad a Diferentes Niveles de Conocimiento: Este manual ha sido diseñado para ser útil tanto para principiantes como para estudiantes y profesionales más experimentados. Las prácticas abarcan desde conceptos fundamentales hasta técnicas avanzadas, permitiendo a los usuarios elegir su nivel de desafío.
4. Flexibilidad y Aprendizaje Autodirigido: Este manual de prácticas proporciona una estructura flexible que permite a los usuarios aprender a su propio ritmo y centrarse en las áreas que más les interesen o necesiten mejorar. Los ejercicios y ejemplos se presentan de manera clara y accesible.
5. Contribución a la Formación Académica y Profesional: Al ofrecer ejemplos y ejercicios específicos, este manual puede ser utilizado en cursos académicos de Ingeniería de Software, capacitaciones en la industria y programas de aprendizaje autodirigido. Sirve como una herramienta valiosa para complementar la enseñanza en el aula y el desarrollo profesional.

3.3 OBJETIVO GENERAL DEL MANUAL DE PRÁCTICAS

El objetivo de este manual de prácticas es facilitar el aprendizaje y la aplicación efectiva de los principios, técnicas y métodos de la Ingeniería de Software. A través de ejercicios, ejemplos y actividades prácticas, se busca que los usuarios adquieran habilidades esenciales en el análisis de requisitos, diseño, desarrollo, pruebas e implantación de software. Además, se promueve el pensamiento crítico y la resolución de problemas en el contexto de proyectos de desarrollo de software, preparando a los estudiantes y profesionales para enfrentar desafíos en el mundo de la tecnología de la información.

3.4 DESARROLLO

3.4.1 Práctica 1 Elaboración y revisión del anteproyecto del sistema de desarrollo

3.4.1.1 Objetivo

El objetivo de la Práctica 1 es capacitar a los participantes en la elaboración de un anteproyecto de sistema de desarrollo de software, siguiendo los estándares y mejores prácticas de la Ingeniería de Software. A través de esta actividad, se busca que los estudiantes y profesionales adquieran habilidades fundamentales en la definición de los objetivos del proyecto, la identificación de requisitos, y la elaboración de una propuesta inicial que sirva como base sólida para el desarrollo de soluciones de software efectivas.

3.4.1.2 Introducción

La Práctica 1 es el punto de partida en el proceso de desarrollo de software. Antes de adentrarnos en las fases de diseño, desarrollo, pruebas y despliegue, es esencial contar con una comprensión clara de lo que se busca lograr y cómo se abordarán los desafíos del proyecto. En esta práctica, exploraremos la elaboración y revisión de un anteproyecto, que servirá como el mapa inicial que guiará todo el proceso de desarrollo.

La elaboración de un anteproyecto eficaz requiere una comprensión detallada de los objetivos del sistema, los requisitos del cliente y las restricciones del proyecto. A lo largo de esta práctica, los participantes tendrán la oportunidad de aplicar conceptos relacionados con la revisión de especificación de requisitos (según la norma IEEE830), la trazabilidad de requisitos, el estudio de factibilidad y el análisis costo-beneficio.

3.4.1.3 Correlación Los Temas Y Subtemas Del Programa De Estudio Vigente.

Esta actividad corresponde a los subtemas 1.1 Revisión de especificación de requisitos, 1.1.2 Trazabilidad de requisitos, 1.4 Estudio de Factibilidad y 1.5 Análisis Costo-Beneficio.

3.4.1.4 Material Y Equipo Necesario

1. Papel y bolígrafos para tomar notas.
2. Acceso a una computadora con software de procesamiento de texto y hojas de cálculo.
3. Documentación relevante de la norma IEEE830 (disponible en línea o en la biblioteca).
4. Acceso a internet para investigación y consulta de recursos relacionados con la práctica.
5. Acceso a herramientas de colaboración si se realiza en equipo (como herramientas de gestión de proyectos o comunicación).

3.4.1.5 Metodología

1. Definición de Objetivos: Comience por definir los objetivos del sistema que se va a desarrollar. Esto implica una comprensión clara de las necesidades del cliente y los resultados esperados.
2. Revisión de Especificaciones de Requisitos: Analice detenidamente la documentación de los requisitos, siguiendo la norma IEEE830, y asegúrese de comprender completamente los detalles del proyecto.
3. Identificación de Requisitos Clave: Identifique los requisitos más críticos y aquellos que tienen un impacto significativo en el diseño y la implementación del sistema.

4. **Elaboración de Propuesta Inicial:** Utilice la información recopilada para elaborar una propuesta inicial que describa de manera resumida el sistema propuesto, los objetivos, los requisitos y las ventajas esperadas.
5. **Evaluación de Factibilidad:** Realice una evaluación de factibilidad técnica, económica y operativa del proyecto, teniendo en cuenta los recursos disponibles y las restricciones.
6. **Análisis Costo-Beneficio:** Realice un análisis costo-beneficio para evaluar la viabilidad financiera del proyecto y determinar si es una inversión adecuada.
7. **Revisión y Aprobación:** Revise y ajuste la propuesta según sea necesario y obtenga la aprobación del cliente o el supervisor antes de avanzar al siguiente paso del ciclo de desarrollo.

3.4.1.5 Sugerencias Didácticas

- **Fomentar la Colaboración:** Si es posible, realice esta práctica en equipos de trabajo. La colaboración en la elaboración del anteproyecto permite la discusión y el intercambio de ideas, enriqueciendo el proceso.
- **Investigación Activa:** Animen a los participantes a investigar ejemplos de anteproyectos de software reales. Esto les proporcionará ejemplos concretos y les ayudará a comprender cómo se aplican los conceptos en situaciones reales.
- **Enfoque en la Comunicación:** Hagan hincapié en la importancia de la comunicación efectiva con el cliente o el supervisor. La comprensión de las necesidades y expectativas del cliente es crucial en esta etapa.
- **Retroalimentación y Revisión:** Fomenten la revisión y la retroalimentación constante en la elaboración del anteproyecto. Esto ayudará a perfeccionar la propuesta y asegurar que se alinee con los requisitos del cliente.

3.4.1.6 Reporte Del Alumno

1. Portada: Datos personales, título de la práctica, fecha y nombre del instructor.
2. Índice: Lista de secciones y sus números de página.
3. Introducción: Breve explicación del propósito de la práctica.
4. Resultados de la Práctica:
 - A. Objetivos definidos.
 - B. Requisitos clave identificados.
 - C. Propuesta inicial del sistema.
 - D. Resultados de la evaluación de factibilidad.
 - E. Hallazgos del análisis costo-beneficio.
5. Conclusiones: Resumen de las lecciones aprendidas y observaciones significativas.
6. Recomendaciones: Sugerencias y recomendaciones basadas en los resultados de la práctica.
7. Anexos (si es necesario): Cualquier documentación adicional, como capturas de pantalla, tablas, gráficos u otros elementos de respaldo.

3.4.2 Práctica 2 Con la ayuda de una herramienta CASE elabora el análisis del modelo de negocio seleccionado, considerando el modelo de requisitos, casos de uso, documentación de casos de uso y modelo de dominio.

3.4.2.1 Objetivo

El objetivo de la Práctica 2 es proporcionar a los participantes la oportunidad de utilizar una herramienta CASE para realizar un análisis completo del modelo de negocio seleccionado. A través de esta actividad, se busca que los estudiantes adquieran experiencia en la creación de modelos de requisitos, casos de uso, documentación de casos de uso y modelos de dominio, aplicando metodologías reconocidas de la Ingeniería de Software.

3.4.2.2 Introducción

La Práctica 2 se centra en la fase de análisis del modelo de negocio en el ciclo de desarrollo de software. El análisis del modelo de negocio es esencial para comprender en profundidad los requisitos del sistema, los actores involucrados y cómo el sistema interactuará con su entorno.

En esta práctica, utilizaremos una herramienta CASE (Computer-Aided Software Engineering) para facilitar y optimizar el proceso de análisis. Una herramienta CASE proporciona una serie de funcionalidades que ayudan a capturar, documentar y visualizar los aspectos clave del sistema, lo que mejora la comunicación y la comprensión de los requerimientos por parte de los stakeholders.

3.4.2.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.

Esta actividad corresponde a los subtemas 1.1 Revisión de especificación de requisitos, 1.1.2 Trazabilidad de requisitos.

3.4.2.4 Material Y Equipo Necesario

1. Papel y bolígrafos para tomar notas.
2. Acceso a una computadora con la herramienta CASE seleccionada previamente.
3. Documentación relacionada con el modelo de negocio a analizar.
4. Conexión a internet para acceder a recursos adicionales.
5. Herramienta CASE seleccionada para la elaboración del análisis del modelo de negocio.
6. Acceso a herramientas de comunicación y colaboración si se realiza en equipo.

3.4.2.5 Metodología

1. Selección del Modelo de Negocio: Elija un modelo de negocio o sistema a analizar. Puede ser un sistema real o uno ficticio que se adapte a los objetivos de la práctica.
2. Familiarización con la Herramienta CASE: Familiarícese con la herramienta CASE seleccionada. Aprenda cómo utilizarla para crear modelos de requisitos, casos de uso, documentación de casos de uso y modelos de dominio.
3. Identificación de Actores y Requisitos: Identifique los actores clave y los requisitos del sistema en el modelo de negocio. Utilice la herramienta CASE para capturar esta información de manera organizada.
4. Creación de Casos de Uso: Cree casos de uso que describan las interacciones entre los actores y el sistema. Documente cada caso de uso, incluyendo su descripción, actores involucrados, precondiciones, flujos principales y flujos alternativos.
5. Documentación de Casos de Uso: Utilice la herramienta CASE para generar documentación detallada de los casos de uso, que incluya

descripciones más completas, diagramas de secuencia y cualquier otro detalle relevante.

6. Modelo de Dominio: Cree un modelo de dominio que represente las entidades clave y las relaciones en el modelo de negocio. Utilice la herramienta CASE para visualizar este modelo.
7. Revisión y Ajuste: Revise y ajuste los modelos y la documentación creada para asegurarse de que reflejen con precisión el modelo de negocio y los requisitos.
8. Presentación (Opcional): Si se requiere, presente los resultados a su instructor o grupo, utilizando los modelos y documentación generados.

3.4.2.6 Sugerencias Didácticas

- Selección del Modelo de Negocio: Anime a los participantes a elegir un modelo de negocio que les interese o que tenga relevancia en su área de estudio o carrera. Esto aumentará el compromiso y la motivación en la práctica.
- Exploración de la Herramienta CASE: Proporcione tiempo para que los estudiantes se familiaricen con la herramienta CASE antes de comenzar a trabajar en el análisis. Puede ser beneficioso realizar ejercicios de práctica para dominar la herramienta.
- Colaboración en Equipos (Opcional): Si es posible, realice la práctica en equipos, fomentando la colaboración en la creación de modelos y la revisión de resultados.
- Énfasis en la Comunicación: Destaque la importancia de la comunicación clara y efectiva en el proceso de análisis del modelo de negocio. Los participantes deben ser capaces de comunicar sus resultados de manera comprensible para los stakeholders.

3.4.2.7 Reporte Del Alumno

1. Portada: Datos personales, título de la práctica, fecha y nombre del instructor.
2. Índice: Lista de secciones y sus números de página.
3. Introducción: Breve explicación del propósito de la práctica.
4. Descripción de la Práctica: Breve resumen de los pasos y actividades realizadas durante la práctica.
5. Resultados de la Práctica: Detalles específicos sobre los resultados de la práctica, incluyendo la aplicación de la herramienta CASE y los modelos creados.
6. Conclusiones: Resumen de las lecciones aprendidas y observaciones significativas.
7. Recomendaciones: Sugerencias y recomendaciones basadas en los resultados de la práctica.
8. Anexos (si es necesario): Cualquier documentación adicional, como capturas de pantalla, diagramas, o informes generados durante el análisis del modelo de negocio.

3.4.3 Práctica 3 Elaborar un estudio de factibilidad y el costo-beneficio aplicado a la organización

3.4.3.1 Objetivo

El objetivo de la Práctica 3 es capacitar a los participantes en la elaboración de un estudio de factibilidad y análisis de costo-beneficio aplicado a un proyecto de software dentro de una organización. A través de esta actividad, se busca que los estudiantes adquieran habilidades en la evaluación de la viabilidad y rentabilidad de proyectos tecnológicos, lo que es esencial en la toma de decisiones y la planificación estratégica.

3.4.3.2 Introducción

La Práctica 3 se enfoca en una etapa crítica de la gestión de proyectos de software: la evaluación de la viabilidad y el análisis de costos y beneficios. Antes de comprometer recursos y esfuerzo en un proyecto, es crucial determinar si es factible y si proporcionará un retorno de inversión positivo.

En esta práctica, los participantes aprenderán a aplicar técnicas de evaluación de factibilidad y a realizar un análisis detallado de los costos y beneficios asociados con el proyecto de software. Esto les ayudará a tomar decisiones informadas y a justificar la inversión en el proyecto.

3.4.3.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.

Esta actividad corresponde a los subtemas 1.4 Estudio de Factibilidad y 1.5 Análisis Costo-Beneficio.

3.4.3.4 Material Y Equipo Necesario

1. Papel y bolígrafos para tomar notas.
2. Acceso a una computadora con software de hojas de cálculo y procesamiento de texto.
3. Documentación relacionada con el proyecto de software en evaluación.

4. Conexión a internet para acceder a recursos adicionales y herramientas de investigación.
5. Software de hojas de cálculo (como Microsoft Excel o Google Sheets) para realizar cálculos financieros.
6. Acceso a herramientas de comunicación y colaboración si se realiza en equipo.

3.4.3.5 Metodología

1. Selección del Proyecto de Software: Elija un proyecto de software para evaluar su factibilidad y realizar el análisis de costos y beneficios. Puede ser un proyecto real o hipotético, pero debe ser relevante para la organización.
2. Recopilación de Información: Reúna toda la información relevante sobre el proyecto, incluyendo requisitos, estimaciones de costos y plazos, y cualquier información necesaria para evaluar la factibilidad.
3. Estudio de Factibilidad: Realice un estudio de factibilidad que incluya la evaluación de aspectos técnicos, operativos y económicos del proyecto. Determine si el proyecto es técnicamente viable y si se ajusta a los objetivos estratégicos de la organización.
4. Análisis de Costo-Beneficio: Realice un análisis detallado de los costos y beneficios asociados con el proyecto. Calcule el retorno de inversión (ROI) y evalúe si el proyecto generará un valor positivo para la organización.
5. Presentación de Resultados: Si es necesario, presente los resultados del estudio de factibilidad y análisis de costo-beneficio a los stakeholders relevantes dentro de la organización.

3.4.3.6 Sugerencias Didácticas

- Selección del Proyecto: Anime a los participantes a elegir un proyecto que les interese o que tenga relevancia en su área de estudio o carrera. Esto aumentará el compromiso y la motivación en la práctica.
- Uso de Herramientas de Cálculo: Fomente el uso de herramientas de hojas de cálculo, como Microsoft Excel o Google Sheets, para realizar los cálculos financieros en el análisis de costos y beneficios. Proporcione orientación sobre cómo utilizar estas herramientas de manera efectiva.
- Colaboración en Equipos (Opcional): Si es posible, realice la práctica en equipos, fomentando la colaboración en la recopilación de información y el análisis de proyectos. Esto refleja situaciones de trabajo en la vida real.
- Énfasis en la Comunicación: Destaque la importancia de comunicar claramente los resultados del estudio de factibilidad y análisis de costos y beneficios. Los participantes deben ser capaces de presentar sus hallazgos de manera efectiva a los interesados y tomar decisiones informadas.

3.4.3.7 Reporte Del Alumno

1. Portada: Datos personales, título de la práctica, fecha y nombre del instructor.
2. Índice: Lista de secciones y sus números de página.
3. Introducción: Breve explicación del propósito de la práctica.
4. Descripción de la Práctica: Breve resumen de los pasos y actividades realizadas durante la práctica.
5. Resultados de la Práctica:
 - a. Detalles específicos sobre el estudio de factibilidad, incluyendo los aspectos técnicos, operativos y económicos evaluados.

- b. Análisis detallado de los costos y beneficios asociados con el proyecto de software.
- 6. Conclusiones: Resumen de las lecciones aprendidas y observaciones significativas relacionadas con la viabilidad del proyecto.
- 7. Recomendaciones: Sugerencias y recomendaciones basadas en los resultados del estudio de factibilidad y análisis costo-beneficio.
- 8. Anexos (si es necesario): Cualquier documentación adicional, como hojas de cálculo con detalles financieros, fuentes de información utilizadas y otros elementos de respaldo.

3.4.4 Práctica 4 Establecer un diseño preliminar de las interfaces de usuario de acuerdo a los requisitos.

3.4.4.1 Objetivo

El objetivo de la Práctica 4 es capacitar a los participantes en la creación de un diseño preliminar de las interfaces de usuario de un sistema de software, asegurando que se cumplan los requisitos del usuario y se proporcione una experiencia de usuario efectiva. A través de esta actividad, los estudiantes adquirirán habilidades en el diseño de interfaces centradas en el usuario.

3.4.4.2 Introducción

La Práctica 4 se enfoca en la fase de diseño de interfaces de usuario en el ciclo de desarrollo de software. El diseño de interfaces es esencial para garantizar que el software sea intuitivo, eficiente y satisfactorio para los usuarios.

En esta práctica, los participantes aprenderán a crear un diseño preliminar de las interfaces de usuario teniendo en cuenta los requisitos del usuario y los principios de usabilidad. Esto implica la creación de esquemas de diseño, wireframes o prototipos de baja fidelidad que servirán como base para el desarrollo posterior.

3.4.4.3 Especificar La Correlación Con El O Los Temas Y Subtemas Del Programa De Estudio Vigente.

Esta actividad corresponde a los subtemas 2.4 Diseño de interfaz de usuario y 1.1.1 Norma IEEE830.

3.4.4.4 Material Y Equipo Necesario

1. Papel y bolígrafos para tomar notas y bocetar ideas.
2. Acceso a una computadora o dispositivo para la creación de diseños preliminares de interfaces.
3. Especificación de requisitos del sistema o acceso a los requisitos del proyecto.

4. Software de diseño de interfaces de usuario (por ejemplo, Adobe XD, Sketch, Figma o herramientas similares).
5. Conexión a internet para acceder a recursos adicionales y herramientas de investigación.
6. Herramientas de colaboración si se realiza en equipo.

3.4.4.5 Metodología

1. Análisis de Requisitos de Usuario: Comience por revisar y comprender los requisitos del usuario del sistema. Identifique las necesidades, preferencias y expectativas de los usuarios finales.
2. Creación de Wireframes o Prototipos de Baja Fidelidad: Utilice una herramienta de diseño de interfaces de usuario para crear wireframes o prototipos de baja fidelidad que representen de manera esquemática la estructura y el diseño de las interfaces.
3. Iteración y Mejora: Colabore con otros participantes, si es necesario, y realice iteraciones en el diseño preliminar en función de la retroalimentación y las mejoras identificadas.
4. Pruebas de Usabilidad (Opcional): Realice pruebas de usabilidad con usuarios reales o compañeros para evaluar la efectividad del diseño preliminar y recopilar comentarios adicionales.
5. Documentación del Diseño Preliminar: Documente el diseño preliminar, explicando las decisiones de diseño, la estructura de la interfaz y las interacciones clave.

3.4.4.6 Sugerencias Didácticas

- Enfoque en la Usabilidad: Enfatice la importancia de diseñar interfaces que sean fáciles de usar y satisfagan las necesidades de los usuarios. Los estudiantes deben considerar la usabilidad en cada etapa del diseño.

- Ejemplos Prácticos: Proporcione ejemplos prácticos de interfaces de usuario reales o hipotéticas que los estudiantes puedan utilizar como base para sus diseños preliminares.
- Colaboración en Equipos (Opcional): Fomente la colaboración en equipos, ya que el diseño de interfaces puede ser un esfuerzo conjunto que involucre a diseñadores, desarrolladores y otros profesionales.
- Retroalimentación y Mejora Continua: Anime a los estudiantes a buscar retroalimentación de compañeros o usuarios y a utilizarla para mejorar sus diseños preliminares.

3.4.4.7 Reporte Del Alumno

1. Introducción: Breve explicación del propósito de la práctica.
2. Descripción del Diseño Preliminar: Muestre los wireframes o prototipos de baja fidelidad creados.
3. Explicación del Diseño: Detalle las decisiones de diseño y justificaciones basadas en requisitos de usuario y buenas prácticas.
4. Iteraciones y Mejoras: Describa cualquier mejora realizada en el diseño y las razones detrás de estas modificaciones.
5. Conclusión: Resumen de las lecciones aprendidas y observaciones clave sobre el diseño preliminar.
6. Recomendaciones: Sugerencias y recomendaciones basadas en los resultados del diseño preliminar.

3.4.5 Práctica 5 Elaborar un diseño de bases de datos emanado del modelo entidad-relación

3.4.5.1 Objetivo

El objetivo de la Práctica 5 es capacitar a los participantes en la creación de un diseño de bases de datos basado en un modelo entidad-relación previamente desarrollado. A través de esta actividad, los estudiantes adquirirán habilidades en el diseño de bases de datos eficientes y relacionales que respalden la gestión de datos en aplicaciones de software.

3.4.5.2 Introducción

La Práctica 5 se centra en la fase de diseño de bases de datos en el ciclo de desarrollo de software. Un diseño de bases de datos efectivo es crucial para garantizar que los datos se almacenen, recuperen y gestionen de manera eficiente y precisa.

En esta práctica, los participantes tomarán un modelo entidad-relación existente y traducirán sus elementos en tablas, atributos y relaciones en una base de datos relacional. Esto implica la identificación de claves primarias, claves foráneas y la normalización de las tablas.

3.4.5.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.

Esta actividad corresponde al subtema 2.3 Diseño de datos.

3.4.5.4 Material Y Equipo Necesario

1. Papel y bolígrafos para tomar notas y esquematizar diseños.
2. Acceso a una computadora con software de gestión de bases de datos, como MySQL, PostgreSQL o herramientas similares.
3. Modelo entidad-relación existente o acceso a las especificaciones del modelo.

4. Conexión a internet para acceder a recursos adicionales y herramientas de investigación.
5. Software de diseño y gestión de bases de datos para crear el diseño propuesto.
6. Acceso a herramientas de colaboración si se realiza en equipo.

3.4.5.5 Metodología

1. Revisión del Modelo Entidad-Relación: Comience revisando el modelo entidad-relación existente para comprender la estructura de datos y las relaciones entre las entidades.
2. Traducción a Diseño de Bases de Datos: Traduzca los elementos del modelo entidad-relación en tablas de base de datos, identificando las claves primarias y foráneas, y definiendo las relaciones.
3. Normalización: Aplique el proceso de normalización para garantizar la eficiencia y la integridad de la base de datos.
4. Creación del Esquema de Bases de Datos: Utilice software de gestión de bases de datos para crear un esquema que refleje el diseño propuesto.
5. Documentación del Diseño: Documente el diseño de la base de datos, explicando las tablas, campos, claves y relaciones. Proporcione una descripción general.

3.4.5.6 Sugerencias Didácticas

- Enfoque en la Estructura de Datos: Destaque la importancia de una estructura de datos sólida en la gestión de bases de datos. Los estudiantes deben comprender cómo organizar los datos de manera eficiente.
- Uso de Herramientas de Bases de Datos: Fomente el uso de software de gestión de bases de datos y proporcione orientación sobre cómo utilizarlo para crear el diseño propuesto.

- Colaboración en Equipos (Opcional): Si es posible, realice la práctica en equipos, fomentando la colaboración en la creación del diseño de bases de datos.
- Retroalimentación y Mejora Continua: Anime a los estudiantes a buscar retroalimentación de compañeros o el instructor y a utilizarla para mejorar sus diseños de bases de datos.

3.4.5.7 Reporte Del Alumno

1. Introducción: Breve explicación del propósito de la práctica.
2. Revisión del Modelo Entidad-Relación: Resumen de los elementos clave del modelo entidad-relación revisado.
3. Diseño de la Base de Datos: Presentación del diseño propuesto de la base de datos, incluyendo tablas, campos, claves primarias y foráneas, y relaciones.
4. Normalización (si aplicable): Si se realizó el proceso de normalización, descripción de las formas normales alcanzadas y su importancia.
5. Esquema de la Base de Datos: Capturas de pantalla o descripciones del esquema de la base de datos creado en el software de gestión.
6. Conclusión: Resumen de las lecciones aprendidas y observaciones clave sobre el diseño de bases de datos.
7. Recomendaciones: Sugerencias y recomendaciones basadas en el proceso de diseño y las lecciones aprendidas.

3.4.5 Práctica 6 A partir del diccionario de datos y el diagrama E-R crear una base de datos

3.4.6.1 Objetivo

El objetivo de la Práctica 6 es capacitar a los participantes en la creación de una base de datos a partir del diccionario de datos y el diagrama entidad-relación (E-R) de un sistema. A través de esta actividad, los estudiantes adquirirán habilidades en la implementación de un sistema de gestión de bases de datos que refleje la estructura y relaciones definidas en el diseño.

3.4.6.2 Introducción

La Práctica 6 se enfoca en la fase de implementación de bases de datos en el ciclo de desarrollo de software. Una implementación precisa y efectiva es crucial para garantizar que la base de datos funcione de manera eficiente y cumpla con los requisitos del sistema.

En esta práctica, los participantes tomarán el diccionario de datos y el diagrama entidad-relación (E-R) existente y crearán una base de datos funcional que refleje la estructura y las relaciones definidas en el diseño. Esto implica la creación de tablas, definición de claves primarias y foráneas, y la incorporación de datos de muestra.

3.4.6.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.

Esta actividad corresponde al subtema 2.4 Diseño de Datos.

3.4.6.4 Material Y Equipo Necesario

1. Papel y bolígrafos para tomar notas y esquematizar diseños.
2. Acceso a una computadora con software de gestión de bases de datos, como MySQL, PostgreSQL o herramientas similares.
3. Diccionario de datos y diagrama entidad-relación (E-R) del sistema.

4. Conexión a internet para acceder a recursos adicionales y herramientas de investigación.
5. Software de gestión de bases de datos para implementar el diseño.
6. Acceso a herramientas de colaboración si se realiza en equipo.

3.4.6.5 Metodología

1. Revisión de Diseño y Diccionario de Datos: Comience revisando detenidamente el diseño del sistema, el diagrama entidad-relación y el diccionario de datos para comprender la estructura y relaciones de la base de datos.
2. Creación de Tablas: Cree las tablas de la base de datos, definiendo los campos, tipos de datos, claves primarias y foráneas según lo indicado en el diseño.
3. Implementación de Restricciones: Agregue restricciones de integridad referencial, como claves foráneas y reglas de validación, para garantizar la consistencia de los datos.
4. Carga de Datos de Muestra: Inserte datos de muestra en las tablas para simular el funcionamiento de la base de datos.
5. Pruebas y Validación: Realice pruebas para asegurarse de que la base de datos funcione según lo esperado y valide la precisión de los datos cargados.

3.4.6.6 Sugerencias Didácticas

- Enfocarse en la Precisión: Destaque la importancia de implementar la base de datos de manera precisa, asegurando que se reflejen fielmente el diseño y las relaciones definidas en el E-R y el diccionario de datos.
- Uso de Herramientas de Bases de Datos: Fomente el uso de software de gestión de bases de datos y proporcione orientación sobre cómo crear las tablas y aplicar restricciones.

- Colaboración en Equipos (Opcional): Si es posible, realice la práctica en equipos, fomentando la colaboración en la implementación de la base de datos.
- Pruebas y Validación: Anime a los estudiantes a realizar pruebas exhaustivas para verificar que la base de datos funcione correctamente y que los datos de muestra sean precisos.

3.4.6.7 Reporte Del Alumno

1. Introducción: Breve explicación del propósito de la práctica.
2. Revisión del Diseño y Diccionario de Datos: Resumen de los elementos clave del diseño E-R y el diccionario de datos.
3. Creación de Tablas: Muestre las tablas creadas, incluyendo campos, tipos de datos, claves primarias y foráneas.
4. Implementación de Restricciones: Descripción de las restricciones de integridad referencial aplicadas en la base de datos.
5. Carga de Datos de Muestra: Indique cómo se insertaron los datos de muestra en las tablas para simular el funcionamiento del sistema.
6. Pruebas y Validación: Resumen de las pruebas realizadas para asegurar el funcionamiento adecuado de la base de datos y la precisión de los datos.
7. Conclusión: Resumen de las lecciones aprendidas y observaciones clave sobre la implementación de la base de datos.
8. Recomendaciones: Sugerencias y recomendaciones basadas en el proceso de implementación y las lecciones aprendidas.

3.4.7 Práctica 7 Usando un lenguaje de programación establecer la conexión a una base de datos.

3.4.7.1 Objetivo

El objetivo de la Práctica 7 es capacitar a los participantes en el proceso de establecer una conexión a una base de datos utilizando un lenguaje de programación. A través de esta actividad, los estudiantes adquirirán habilidades en la conexión efectiva entre una aplicación y una base de datos para realizar operaciones de lectura y escritura de datos.

3.4.7.2 Introducción

La Práctica 7 se enfoca en la interacción entre aplicaciones de software y bases de datos, una tarea fundamental en el desarrollo de sistemas. La habilidad de establecer una conexión a una base de datos es crucial para permitir que una aplicación almacene, recupere y gestione datos de manera eficiente.

En esta práctica, los participantes utilizarán un lenguaje de programación (como Python, Java o similar) para establecer una conexión a una base de datos, autenticarse en ella y ejecutar operaciones básicas, como consultas y actualizaciones de datos.

3.4.7.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.

Esta actividad corresponde al subtema 3.1 Lenguajes para el desarrollo de software.

3.4.7.4 Material Y Equipo Necesario

1. Papel y bolígrafos para tomar notas y esquematizar código.
2. Acceso a una computadora con un lenguaje de programación instalado (por ejemplo, Python, Java, C#) y acceso a internet.
3. Software de gestión de bases de datos (por ejemplo, MySQL, PostgreSQL).

4. Base de datos de muestra o acceso a una base de datos real para la práctica.
5. Documentación y recursos de referencia sobre el lenguaje de programación y la biblioteca de conexión a bases de datos.

3.4.7.5 Metodología

1. Selección del Lenguaje de Programación: Elija un lenguaje de programación adecuado para establecer la conexión a la base de datos. Asegúrese de que el lenguaje sea compatible con la biblioteca de conexión de la base de datos que se utilizará.
2. Configuración del Entorno: Configure el entorno de desarrollo, incluyendo la instalación de la biblioteca de conexión de la base de datos y la configuración de las credenciales de acceso a la base de datos.
3. Desarrollo del Código de Conexión: Escriba el código necesario para establecer una conexión a la base de datos. Esto incluye la especificación de la URL de conexión, la autenticación y la creación de una conexión.
4. Ejecución de Operaciones de Base de Datos: Desarrolle código para realizar operaciones de lectura y escritura en la base de datos, como consultas SQL y actualizaciones de datos.
5. Pruebas y Depuración: Realice pruebas exhaustivas para asegurarse de que la conexión y las operaciones de base de datos funcionen correctamente. Resuelva cualquier problema o error que surja.

3.4.7.6 Sugerencias Didácticas

- Enfoque en la Práctica: Anime a los estudiantes a practicar la escritura de código real para establecer una conexión a la base de datos y ejecutar operaciones.

- Utilización de Ejemplos: Proporcione ejemplos prácticos y guías de referencia sobre cómo usar la biblioteca de conexión en el lenguaje de programación seleccionado.
- Depuración de Errores: Enseñe a los estudiantes cómo depurar problemas comunes, como errores de conexión o errores de sintaxis SQL.
- Seguridad y Buenas Prácticas: Destaque la importancia de la seguridad al manejar credenciales de acceso a la base de datos y alentando a los estudiantes a seguir buenas prácticas de codificación.

3.4.7.7 Reporte Del Alumno

1. Introducción: Breve explicación del propósito de la práctica.
2. Selección del Lenguaje de Programación: Especifique el lenguaje de programación seleccionado para la práctica y las razones detrás de esta elección.
3. Configuración del Entorno: Describa la configuración del entorno de desarrollo, incluyendo la instalación de la biblioteca de conexión a la base de datos y la configuración de las credenciales.
4. Código de Conexión: Muestre y explique el código desarrollado para establecer una conexión a la base de datos. Esto incluye detalles como la URL de conexión, autenticación y creación de la conexión.
5. Operaciones de Base de Datos: Describa las operaciones de lectura y escritura que realizó en la base de datos utilizando el lenguaje de programación. Proporcione ejemplos de consultas SQL o actualizaciones de datos.
6. Pruebas y Depuración: Resuma las pruebas realizadas para asegurarse de que la conexión y las operaciones de base de datos funcionen correctamente. Si encontró problemas o errores, indique cómo los resolvió.

7. Conclusión: Resumen de las lecciones aprendidas y observaciones clave sobre la conexión a la base de datos con un lenguaje de programación.
8. Recomendaciones: Sugerencias y recomendaciones basadas en la experiencia adquirida durante la práctica.

3.4.5 Práctica 8 Desarrollar los procesos identificados, asegurando las operaciones básicas de todo sistema: registro, actualización, consulta y estadística.

3.4.8.1 Objetivo

El objetivo de la Práctica 8 es capacitar a los participantes en el desarrollo de los procesos esenciales de un sistema de software, incluyendo la implementación de operaciones básicas como registro, actualización, consulta y estadísticas. A través de esta actividad, los estudiantes adquirirán habilidades en la programación de funcionalidades clave para la gestión de datos en un sistema.

3.4.8.2 Introducción

La Práctica 8 se enfoca en la implementación de operaciones básicas que son fundamentales para el funcionamiento de cualquier sistema de software. Estas operaciones incluyen la capacidad de registrar nuevos datos, actualizar registros existentes, realizar consultas para recuperar información y generar estadísticas sobre los datos.

En esta práctica, los participantes desarrollarán código de programación para cada una de estas operaciones, lo que implica la interacción con una base de datos, la gestión de formularios de entrada y la presentación de resultados.

3.4.8.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.

Esta actividad corresponde al subtema 3.1 Lenguajes para el desarrollo de software.

3.4.8.4 Material Y Equipo Necesario

1. Papel y bolígrafos para tomar notas y esquematizar diseños de formularios y operaciones.
2. Acceso a una computadora con un entorno de desarrollo integrado (IDE) y el lenguaje de programación necesario instalado.

3. Acceso a una base de datos relacionada con el proyecto (puede ser una base de datos de muestra).
4. Conexión a internet para acceder a recursos adicionales y herramientas de investigación.
5. Software de gestión de bases de datos si es necesario para la práctica.
6. Documentación y recursos de referencia sobre el lenguaje de programación.

3.4.8.5 Metodología

1. Identificación de Procesos Esenciales: Comience identificando los procesos esenciales que deben ser implementados en el sistema, como registro, actualización, consulta y estadísticas.
2. Diseño de Interfaces de Usuario: Diseñe las interfaces de usuario necesarias para cada proceso, incluyendo formularios de entrada de datos y pantallas de consulta y visualización de estadísticas.
3. Desarrollo de Código: Escriba el código de programación para cada uno de los procesos identificados. Esto incluye la interacción con la base de datos, la validación de datos y la presentación de resultados.
4. Pruebas y Depuración: Realice pruebas exhaustivas para asegurarse de que todas las operaciones funcionen correctamente. Depure cualquier error o problema que surja.

3.4.8.6 Sugerencias Didácticas

- Enfoque en la Práctica: Fomente la práctica activa en el desarrollo de las operaciones básicas. Los estudiantes deben escribir código real para implementar cada operación.
- Uso de Ejemplos: Proporcione ejemplos prácticos y guías de referencia sobre cómo desarrollar operaciones de registro, actualización, consulta y estadísticas en el lenguaje de programación seleccionado.

- Validación de Datos: Enseñe a los estudiantes a validar datos de entrada para garantizar la integridad de los datos en el sistema.
- Seguridad y Buenas Prácticas: Destaque la importancia de la seguridad al manipular datos y alentando a los estudiantes a seguir buenas prácticas de codificación.

3.4.8.7 Reporte Del Alumno

1. Introducción: Breve descripción del propósito de la práctica.
2. Procesos Implementados: Enumere y describa los procesos implementados (registro, actualización, consulta y estadísticas).
3. Diseño de Interfaces: Proporcione una descripción de las interfaces de usuario diseñadas para cada proceso.
4. Código de Implementación: Incluya el código de programación desarrollado para cada proceso.
5. Pruebas y Resultados: Resuma las pruebas realizadas y los resultados obtenidos para cada operación.
6. Conclusión: Resumen de las lecciones aprendidas y observaciones clave sobre el desarrollo de operaciones básicas.
7. Recomendaciones: Sugerencias y recomendaciones basadas en la experiencia adquirida durante la práctica.

3.4.9 Práctica 9 Probar el sistema con las técnicas existentes y validar que el modelo de requisitos esté atendido.

3.4.9.1 Objetivo

El objetivo de la Práctica 9 es capacitar a los participantes en el proceso de prueba de un sistema de software utilizando técnicas existentes y validar que se cumplen los requisitos establecidos en el modelo de requisitos. A través de esta actividad, los estudiantes adquirirán habilidades en la evaluación y verificación de la funcionalidad y calidad del sistema.

3.4.9.2 Introducción

La Práctica 9 se centra en la fase de pruebas de software, que es fundamental para garantizar que el sistema cumple con los requisitos definidos. Las pruebas son una parte esencial del ciclo de desarrollo de software y permiten detectar posibles errores y evaluar la calidad del sistema.

En esta práctica, los participantes llevarán a cabo pruebas utilizando técnicas existentes, como pruebas de unidad, pruebas de integración, pruebas de sistema y pruebas de aceptación, para asegurarse de que el sistema cumple con los requisitos definidos en el modelo de requisitos.

3.4.9.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.

Esta actividad corresponde a los subtemas 4.1 Diseño de caso de prueba, 4.2 Pruebas de componentes y 4.3 Pruebas del sistema.

3.4.9.4 Material Y Equipo Necesario

1. Documentación de requisitos del sistema.
2. Documentación de diseño del sistema.
3. Herramientas de gestión de pruebas, como software de automatización de pruebas (si es aplicable).
4. Acceso al sistema o a un entorno de pruebas.

5. Computadora con acceso a herramientas de gestión de pruebas y software de registro de defectos (si se utiliza).
6. Acceso a una base de datos de pruebas o un entorno aislado para pruebas.
7. Documentación y recursos de referencia sobre técnicas de prueba.

3.4.9.5 Metodología

1. Revisión de Requisitos y Diseño: Comience revisando la documentación de requisitos y diseño del sistema para comprender los objetivos y las funcionalidades del sistema.
2. Planificación de Pruebas: Elabore un plan de pruebas que incluya los tipos de pruebas que se realizarán, los casos de prueba a ejecutar y los criterios de aceptación.
3. Diseño de Casos de Prueba: Diseñe casos de prueba detallados que cubran los escenarios clave del sistema y evalúen si cumple con los requisitos.
4. Ejecución de Pruebas: Ejecute los casos de prueba de acuerdo con el plan de pruebas. Registre los resultados y cualquier defecto encontrado.
5. Análisis de Resultados: Analice los resultados de las pruebas y compare los resultados con los criterios de aceptación.
6. Validación de Requisitos: Verifique que el sistema cumple con los requisitos definidos en la documentación de requisitos.

3.4.9.6 Sugerencias Didácticas

- Planificación de Pruebas Detallada: Enseñe a los estudiantes a elaborar un plan de pruebas exhaustivo que cubra todos los aspectos del sistema y los requisitos.
- Diseño de Casos de Prueba Efectivos: Destaque la importancia de diseñar casos de prueba efectivos que aborden los escenarios críticos y prueben la funcionalidad del sistema.

- Ejecución y Registro de Pruebas: Fomente la práctica de ejecutar pruebas y registrar los resultados de manera organizada, incluyendo la documentación de defectos si se encuentran.
- Validación de Requisitos: Haga hincapié en la importancia de la validación de requisitos y cómo las pruebas contribuyen a este proceso.

3.4.9.7 Reporte Del Alumno

1. Introducción: Breve descripción del propósito de la práctica.
2. Plan de Pruebas: Muestre el plan de pruebas elaborado para la práctica.
3. Casos de Prueba: Enumere y describa los casos de prueba diseñados.
4. Ejecución de Pruebas: Resuma la ejecución de pruebas, incluyendo resultados y defectos encontrados.
5. Análisis de Resultados: Analice los resultados de las pruebas y cómo se comparan con los criterios de aceptación.
6. Validación de Requisitos: Verifique si el sistema cumple con los requisitos definidos en la documentación de requisitos.

3.4.10 Práctica 10 Implementar el sistema, capacitar a los usuarios y verificar la estabilidad del sistema para su liberación.

3.4.10.1 Objetivo

El objetivo de la Práctica 10 es llevar a cabo la implementación final del sistema de software, capacitar a los usuarios finales y verificar la estabilidad del sistema antes de su liberación. El objetivo principal es garantizar una transición exitosa del sistema desde el desarrollo hasta su uso en producción.

3.4.10.2 Introducción

La Práctica 10 se centra en la fase de implementación y despliegue de un sistema de software. Es una etapa crítica que implica la instalación del sistema en un entorno de producción, la capacitación de los usuarios finales y la verificación de que el sistema funcione de manera estable antes de su liberación.

En esta práctica, los participantes llevarán a cabo actividades como la instalación del software en servidores, la configuración de bases de datos, la capacitación de usuarios y la realización de pruebas de estabilidad para asegurar que el sistema esté listo para su uso en producción.

3.4.10.3 Correlación Con Los Temas Y Subtemas Del Programa De Estudio Vigente.

Esta actividad corresponde a los subtemas 4.4 Documentación de resultados de las pruebas, 4.5 Entrega del sistema y Capacitación a usuarios y 4.6 Entrega de documentación técnica y de usuario del sistema.

3.4.10.4 Material Y Equipo Necesario

1. Versión final del software a implementar.
2. Documentación de usuario, manuales y guías de referencia.
3. Equipos de hardware y servidores necesarios para la implementación.
4. Herramientas de gestión de proyectos y comunicación.

5. Personal técnico encargado de la implementación y configuración del sistema.
6. Usuarios finales que recibirán capacitación.
7. Espacio para la capacitación de usuarios si es necesario.
8. Recursos de soporte técnico para abordar posibles problemas durante la implementación.

3.4.10.5 Metodología

1. Planificación de Implementación: Elabore un plan detallado que describa el proceso de implementación, incluyendo fechas, tareas y responsabilidades.
2. Instalación y Configuración: Instale el software en los servidores designados y configure la infraestructura necesaria, como bases de datos y servidores web.
3. Capacitación de Usuarios: Proporcione capacitación a los usuarios finales para que puedan utilizar el sistema de manera efectiva. Esto puede incluir sesiones presenciales o en línea, manuales y material de referencia.
4. Pruebas de Estabilidad: Realice pruebas de estabilidad para asegurarse de que el sistema funcione de manera confiable en un entorno de producción. Monitoree el rendimiento y la escalabilidad.
5. Soporte Técnico y Solución de Problemas: Establezca un proceso de soporte técnico y solución de problemas para abordar cualquier problema que surja durante la implementación.

3.4.10.6 Sugerencias Didácticas

- Planificación Detallada: Enseñe a los estudiantes a elaborar planes de implementación detallados que aborden todos los aspectos, desde la instalación hasta la capacitación de usuarios.

- **Enfoque en la Capacitación:** Destaque la importancia de la capacitación de usuarios y proporcione ejemplos de enfoques efectivos para la enseñanza del sistema.
- **Pruebas Rigurosas:** Anime a los estudiantes a realizar pruebas exhaustivas de estabilidad y rendimiento para garantizar que el sistema sea confiable en un entorno de producción.
- **Soporte Técnico Efectivo:** Enseñe la importancia de establecer un proceso eficaz de soporte técnico y cómo abordar problemas de manera eficiente.

3.4.10.7 Reporte Del Alumno

1. **Introducción:** Breve descripción del propósito de la práctica.
2. **Plan de Implementación:** Muestre el plan de implementación elaborado, incluyendo fechas y tareas.
3. **Instalación y Configuración:** Describa el proceso de instalación y configuración del sistema.
4. **Capacitación de Usuarios:** Detalle cómo se llevó a cabo la capacitación de usuarios y qué materiales se utilizaron.
5. **Pruebas de Estabilidad:** Resuma las pruebas de estabilidad y los resultados obtenidos.
6. **Soporte Técnico y Solución de Problemas:** Proporcione información sobre el proceso de soporte técnico y cómo se abordaron los problemas durante la implementación.

FUENTES DE INFORMACIÓN

- Booch G. (2006). El lenguaje Unificado de Modelado, UML 2.0, Guía de Usuario. 2ª. Edición. España: Pearson ADDISON-WESLEY.
- Braude, E. (2003). Ingeniería de Software una perspectiva orientada a objetos. México: ALFAOMEGA.
- Fontela, C. (2011)UML Modelado de Software para Profesionales. 1ª. Edición, Buenos Aires: Alfa-Omega Grupo Editor.
- Fowler M. (2000). UML Gota a Gota. 1ª. Edición. México: PEARSON.
- Hansen G.W.,Hansen J.V. 2002 Diseño y administración de Bases de datos:Prentice Hall.
- Kendall E. K. (2011). Análisis y Diseño de sistemas. 8ª. Edición. México: PEARSON..
- Pressman, R.S. (2008). Ingeniería del Software un enfoque práctico. 6ª. Edición México: MC GRAW HILL.
- Senn J.A. (1996). Análisis y Diseño de Sistemas. 2ª Edición. México: MC GRAW HILL.
- Silberschatz, A.; Korth, H. F., Sudarshan, S.2002 Fundamentos de Bases de Datos. 5ª ed. Ed. McGraw Hill.
- Sommerville, Ian (2001). Ingeniería de Software. México: PRENTICE HALL.
- Unam. (Sin fecha). Panorama de los diferentes modelos de ciclo de vida de los sistemas. Consultado en Octubre 2012. Disponible en: http://uxmcc1.iimas.unam.mx/~cursos/Objetos/clases3_4.html.