



**MANUAL DE PRÁCTICAS DE LA
MATERIA
Fundamentos de Ingeniería de
Software.**

**Carrera Ingeniería en Sistemas
Computacionales.**

**Elaborado por
MAESTRA MARIA DEL CARMEN ISLAS
SEPULVEDA.**

Dictamen SS-1-062-2018

**Adscrita al Departamento de Sistemas y
Computación**

Instituto Tecnológico de Hermosillo.

MANUAL DE PRÁCTICAS DE LA MATERIA ADMINISTRACION DE BASE DE DATOS

Carrera Ingeniería en Sistemas Computacionales.

Elaborado por MAESTRA MARIA DEL CARMEN ISLAS SEPULVEDA.

OBJETIVO DE LA MATERIA

Competencias específicas:

- Identificar la historia del software, así como las características de los paradigmas estructurado y orientado a objetos.
- Aplicar modelos, técnicas y herramientas para cada etapa del ciclo de vida de desarrollo de software.

Competencias previas:

- Utiliza algún lenguaje de programación orientado a objetos.
- Aplica razonamiento lógico.
- Identifica conceptos básicos de Bases de Datos
- Aplica técnicas de representación algorítmica

Introducción.

El presente escrito plantea el desarrollo de prácticas de la asignatura Fundamentos de Ingeniería de Software Clave de la asignatura: SCC-1007 2-2-4 (Créditos 2-2-4) SATCA1 Carrera: Ingeniería en Sistemas Computacionales.

Desarrollar un proyecto semestral que cubra las siguientes fases: Ingeniería de requisitos, Modelo de análisis, Modelo de diseño, Modelo de implementación. Brindar al estudiante de la asignatura; los conceptos básicos del papel evolutivo del software, las técnicas, (Modelo CRC, Modelo Objeto-Relación, atributos, operaciones y colaboradores) de manera que el estudiante comprenda el contexto del desarrollo del software orientado a objetos, desde la perspectiva de ingeniería de requisitos y permitiéndole elaborar un anteproyecto para la construcción del software.

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales los conceptos básicos relacionados con el desarrollo de sistemas, los tipos de modelos para el desarrollo y gestión de software considerando la calidad, lo que permite integrar soluciones computacionales con diferentes tecnologías en diversas áreas.

Es una introducción para la planeación y el proceso de desarrollo de proyectos de software que involucra la comprensión de términos, herramientas, métodos, para crear soluciones informáticas eficientes.

De acuerdo a sus competencias específicas está orientada a conocer y aplicar los conceptos básicos de ingeniería de software.

Se relaciona previamente con las asignaturas de programación orientada a objetos y sirve de base para las asignaturas de ingeniería de software y gestión de proyectos.

Es una introducción para la planeación y el proceso de desarrollo de proyectos de software que involucra la comprensión de términos, herramientas, métodos, utilizados para crear soluciones informáticas eficientes.

INDICE.

	Fundamentos de la Ingeniería de Software	
Unidad Uno		
Práctica 1	Generará un glosario de los conceptos básicos de la Ingeniería del software	5
Práctica 2	Elaborar un ensayo de su visión comprensiva respecto las etapas del desarrollo del Software , puede incluir un cuadro descriptivo del ciclo de vida del desarrollo de software.	8
Práctica 3	Elaborar un ensayo sobre las diversas etapas de evolución del desarrollo de productos de software.	10
Práctica 4	Elaborar un cuadro sintético de los diferentes métodos de modelado del software con el fin de contrastar las características de cada modelo, conocer sus ventajas y desventajas, así como el desarrollo de aplicaciones para las cuales son más adecuados, permitiendo hacer una conclusión comparada de los modelos.	12
Unidad Dos	Ingeniería de requisitos	
Práctica 5	Elaborar un mapa mental del proceso de modelado de requerimientos , después de investigar las diferentes tareas que corresponden a la ingeniería de requerimientos de un producto de Software.	15
Práctica 6	Desarrollo de modelo de requerimientos de un caso sencillo propuesto en clase.	18
Práctica 7	Uso de Herramientas CASE para la Ingeniería de requisitos. Usando la herramienta. StarUML documentará un ejercicio propuesto en clase.	20
Práctica 8	Integrados en equipos (tres estudiantes) propondrá un caso basado en un problema real con el fin de considerarlo como el proyecto final para el curso, desarrollando el modelo de requisitos para la propuesta de solución a un problema.	22
Unidad Tres	Modelo de Análisis	
Práctica 9	Basándose en el modelo de requisitos de un producto Software, desarrollara el modelo de análisis de este.	24
Práctica 10	Basándose en el modelo estático de un producto de software clasificará las clases según Estereotipos	26
Práctica 11	Participará en un ejercicio de revisión cruzada (revisión entre pares) de los modelos de análisis presentados por los equipos para su proyecto final. Clases, Diagramas de secuencias, Diccionario de clases según Módulos.	28
Unidad Cuatro	Modelo de Diseño	
Práctica 12	Estrategias de diseño Y Diseño del Sistema.	30
Práctica 13	Diagramas de secuencias del Diseño	32
Práctica 14	Revisión de Diseño	34
Unidad Cinco	Modelo de Implementación.	
Práctica 15	Modelado de la Implementación: Diagrama de componentes, Diagrama de despliegue, Modelos de pruebas.	36

UNIDAD 1.- Introducción a la Ingeniería de Software.

Competencia específica a desarrollar: Identificar la historia del software, así como las características de los paradigmas estructurado y orientado a objetos. Conocer e identificar las herramientas CASE.

Subtemas.

- 1.1 ¿Arte o ingeniería?
- 1.2 Objetivos
- 1.3 Introducción
- 1.4 ¿Qué es la ingeniería?
- 1.5 Ingeniería y ciencias de la ingeniería
- 1.6 El software como artefacto tecnológico
- 1.7 Sistemática, disciplina y cuantificación
- 1.8 La Ingeniería del Software como disciplina profesional
- 1.9 Conceptos básicos de la Ingeniería de Software

Sugerencias Didácticas.

- Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes.
- Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración entre los estudiantes.
- Propiciar el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, en el estudiante, las cuales lo encaminan hacia la investigación, la aplicación de conocimientos y la solución de problemas.
- Observar y analizar fenómenos y problemáticas propias del campo ocupacional.
- Relacionar los contenidos de esta asignatura con las demás del plan de estudios para desarrollar una visión interdisciplinaria en el estudiante.
- Presentar los resultados de la investigación en forma oral y escrita haciendo énfasis en las conclusiones.
- Llevar a cabo actividades prácticas que promuevan el desarrollo de habilidades para la experimentación, tales como observación, identificación, datos relevantes, manejo y control de variables, planteamiento de hipótesis y trabajo en equipo.
- Propiciar el uso adecuado de conceptos y terminología científico tecnológica. Proponer problemas que permitan al estudiante la integración de contenidos de la asignatura y entre distintas asignaturas, para su análisis y solución.

PRACTICAS PROPUESTAS.

Práctica Numero 1. Conceptos básicos de Ingeniería de Software.

Objetivo: El alumno investigará los conceptos básicos de Ingeniería de Software con el propósito construir un glosario de términos, que le permita una vez integrado acudir a él para revisar los conceptos si así lo requiere al desarrollar los temas posteriores de la materia.

Introducción

La ingeniería es según la Real Academia de la Lengua es el conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía. El software es una parte principal del entorno humano. Actualmente sería complicado imaginarse la vida sin toda la tecnología que utilizamos a diario.

El software está presente no sólo en los sistemas informáticos que realizan tareas de tratamiento de información sino en un sinnúmero de sistemas de la más diversa complejidad.

En esta asignatura se introduce el conjunto de técnicas y procedimientos que se han ido desarrollando a lo largo de las últimas décadas, para poder elaborar de una forma ordenada y eficiente el diseño de un software antes de su codificación, es decir la visualización de lo que se quiere construir antes de generar líneas de código que componen el software. Todas estas técnicas y procedimientos forman parte de la ingeniería de software.

El profesional en este ámbito recibe el nombre de ingeniero de software, su actividad supone la concreción de una idea en la realidad, en el ámbito del desarrollo de Sistemas de Software es aquel profesional que aplica un enfoque interdisciplinario que le permite estudiar y comprender la realidad, con el propósito de implementar u optimizar sistemas informáticos complejos.

La aplicación del método científico y la inventiva para llevar a cabo los planes de solución creación o innovación. Siendo que estas actividades incluyen la investigación, inclusive para la búsqueda de nuevas técnicas, el diseño, el desarrollo, la producción, la construcción y la operación.

Es por ello importante que maneje la “jerga” conceptual de esta disciplina.

Correlación con los temas y subtemas del programa de estudio. El glosario de términos, le permitirá una vez integrado acudir a él para dominar la terminología y aclarar sus dudas relacionada con términos y conceptos si así lo requiere al desarrollar los temas posteriores, realizar planteamientos durante sus prácticas en esta materia, así como en las materias para las que esta es un requisito previo.

Material y equipo necesario. Bibliografía, equipo de cómputo para editar el documento, enlace al servidor de la plataforma Moodle.

Metodología.

En esta etapa El estudiante realizará consultas bibliográficas relacionadas con los conceptos planteados en la introducción de la práctica, para integrar un

glosario con los términos básicos de Fundamentos de Ingeniería de Software, deberá incluir la bibliografía que utilizo como fuente para su investigación.

Integrará un documento que enviará al vínculo de la actividad disponible para el curso en la plataforma de Moodle.

Participando en la dinámica de revisión/discusión en grupo de acuerdo a la fecha propuesta por el maestro.

El maestro determinara si es requerido que se realicen más aportaciones al glosario o el trabajo realizado individualmente y en el grupo es suficiente.

Preguntas de análisis en el aula: (Para trabajo individual, equipo y plenaria).

¿Cuáles son las condiciones para que consideremos a la práctica del desarrollo de Software una actividad ingenieril?, ¿Consideras que la programación de sistemas es un arte?, ¿**De qué maneras** hacemos aplicación práctica del conocimiento científico en el diseño y construcción de productos de software?, ¿Por qué es requerida la documentación al desarrollar el análisis de un problema al cual buscamos darle solución a través de Software?, ¿De qué forma podemos garantizar que al desarrollar un software operará de acuerdo a las necesidades del cliente? ¿Dar mantenimiento a un producto de software es malo o es bueno? ¿Solo un producto software mal desarrollado requiere mantenimiento?

Reporte del alumno.

Presentación del trabajo.

- a) **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- b) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- c) **Desarrollará la definición sintética de conceptos y actividades implicados en cada fase del desarrollo de software.**
- d) **Redactara conclusiones personales respecto al tema desarrollado.**
- e) **Bibliografía**

Práctica Numero 2.- Ciclo de vida del desarrollo de productos de software o ciclo de desarrollo de sistemas.

Objetivo.

Que el estudiante plantee a través de un ensayo, su visión comprensiva respecto las etapas del desarrollo del Software, del Ciclo de Vida del Desarrollo del Software. Y se plantee los aspectos que considera cada etapa y la forma en que contribuye a gestionar el desarrollo de un producto software funcionalmente correcto y con un desempeño de calidad.

Introducción.

El ciclo de vida de un sistema es un enfoque por fases, del análisis y diseño que sostiene que los sistemas sean desarrollados de mejor manera mediante el uso de una metodología específica de actividades del analista y del usuario.

A medida que se intenta incrementar la complejidad que abarcaran las aplicaciones de software se incrementa el riesgo de errores costosos por fallas en el software.

Por lo cual es necesario aportar condiciones para evitar los errores en el proceso de desarrollo de software. Permitiendo elevar la tasa de proyectos sean entregados a tiempo, puedan ser usados, trabajar después de ajustes mínimos y algunas correcciones, y generen una relación costo beneficio más predecible, como cualquier otro producto diseñado en base a la ingeniería.

Correlación con los temas y subtemas del programa de estudio. Su visión comprensiva respecto las etapas del desarrollo del Software, el Ciclo de Vida del Desarrollo del Software, permitirá al estudiante desarrollar los temas posteriores y realizar sus prácticas y planteamientos en esta materia, así como todas las materias de su plan de estudios y ejercicio profesional.

Material y equipo necesario. Bibliografía, equipo de cómputo para editar el documento, enlace al servidor de la plataforma Moodle.

Metodología.

El estudiante integrado en equipos de tres, realizará una investigación documental para conocer los aspectos que considera cada etapa del CVDS y la forma en que contribuye cada etapa del desarrollo de un producto software para lograr que sea funcionalmente correcto y con un desempeño de calidad.

Integrando un documento que enviara al curso en la plataforma de Moodle. Además de presentar en exposición/discusión en el aula de acuerdo a la fecha propuesta por el maestro, en base a lo que corresponda el maestro determinara si

es requerido que se realicen más aportaciones o el trabajo realizado en el documento y su presentación en el grupo es suficiente.

Preguntas de análisis en el aula: (Para trabajo individual, equipo y plenaria).
¿Por qué se nos sugiere un enfoque sistémico para el desarrollo de software?, ¿A que nos referimos cuando mencionamos el termino enfoque de sistema o enfoque sistémico? ¿Cuáles son las condiciones para que consideremos a la práctica del desarrollo de Software una actividad sistémica?, ¿Por qué debemos progresar con el análisis, diseño, codificación, prueba y mantenimiento?, ¿De qué maneras realizamos la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora?, ¿Por qué es requerida la documentación al desarrollar el análisis de un problema al cual buscamos darle solución a través de Software?, ¿De qué forma podemos garantizar que al desarrollar un software opere de acuerdo a las necesidades del cliente?
¿Dar mantenimiento a un producto de software es malo o es bueno? ¿Solo un producto software mal desarrollado requiere mantenimiento? ...

Reporte del alumno.

Presentación del trabajo.

- a) **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- b) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- c) **Desarrollará la definición sintética de conceptos y actividades implicados en cada fase del desarrollo de software.**
- d) **Redactara conclusiones personales respecto al tema desarrollado.**
- e) **Bibliografía**

Práctica Numero 3. Elaborar un Ensayo que revise la razón de ser de las diversas etapas de evolución de la Ingeniería de software.

Objetivo: Que el estudiante se plantee a través de un ensayo, una visión los eventos del proceso de desarrollo de software que han impulsado avances y la evolución del proceso de desarrollo de Software.

Correlación con los temas y subtemas del programa de estudio. Su visión comprensiva respecto a la evolución de la Ingeniería de software permitirá al estudiante desarrollar los temas posteriores y realizar sus prácticas y planteamientos en esta materia, así como todas las materias de su plan de estudios y ejercicio profesional.

Introducción.

Durante los primeros años de la informática, el software se consideraba como un añadido. La programación era un "arte", para el que no existían metodologías, era un proceso que se realizaba sin ninguna planificación.

En esa época toda la programación se desarrollaba a la medida para cada aplicación, y en consecuencia tenía muy poca difusión, habitualmente quien lo escribía era porque lo necesitaba, y era quien lo mantenía. La primera época termina con la aparición de los primeros sistemas operativos. El software se establece como producto y aparecieron las empresas dedicadas al desarrollo y distribución masiva del mismo.

El origen del término Ingeniería del Software, se atribuye a dos conferencias organizadas por la OTAN en 1967 y 1968. A mediados de la década de 1970, surge una época en la que los sistemas informáticos aumentaron mucho en su complejidad, Esto supuso mucha presión para los desarrolladores.

Correlación con los temas y subtemas del programa de estudio.

.

Material y equipo necesario. Bibliografía, equipo de cómputo para editar el documento, enlace al servidor de la plataforma Moodle. Proyector para usarlo al exponer su trabajo en clase.

Metodología.

El estudiante partirá de elaborar un cuadro descriptivo de las diversas etapas del Ciclo de vida del desarrollo de software, para adquirir claridad respecto las etapas del desarrollo del Software, para que desarrolle un ensayo donde demuestre conocer el sentido del proceso de desarrollo de software, en el definirá la importancia de considerar al software desde el punto de vista de

producto y desde el punto de vista de artefacto tecnológico, a partir de los procesos que la Ingeniería de Software.

Debe considerar los siguientes puntos en su contenido.

Introducción (Debe describir en forma general el contenido del tema indicando el orden en que este se desarrollará)

Antecedentes (Inicie por describir la "crisis del software" indique con ejemplos los daños ocasionados y termine por definir el concepto de ingeniería del software)

Desarrollo (describa las propuestas de la ingeniería del software, las etapas que propone para el desarrollo del software y describa la importancia de considerar al software desde el punto de vista de producto y desde el punto de vista de artefacto tecnológico).

Conclusiones

Bibliografía

Enviara el documento final al vínculo de la actividad disponible en la plataforma de Moodle en el curso.

Participando en la exposición/discusión en grupo de acuerdo a la fecha propuesta por el maestro, en base a lo que corresponda el maestro determinará si es requerido que se realicen más aportaciones o el trabajo realizado es satisfactorio.

Puede hacerlo a través de una rúbrica.

Preguntas de análisis en el aula: (Para trabajo individual, equipo y plenaria).

¿Por qué comenzaron a definirse herramientas y técnicas para la profesionalización de la actividad del desarrollo de Software? ¿Crees que podríamos hablar de una ingeniería propia para el desarrollo de productos software sin haberse desarrollado metodologías especiales para hacerlo?

¿El software habría evolucionado igual sin el avance de la técnica para desarrollar productos de software?

¿Es posible hablar de metodologías desarrolladas sin congruencia con los entornos o tecnologías donde se desplegarán y ejecutarán las soluciones de software?

¿Qué opinas acerca de que los modelos y técnicas permiten el desarrollo de productos SW de calidad?

¿Pueden las técnicas diseñadas para el desarrollo de SW hacer que la actividad sea más ágil y cercana a las necesidades del usuario?

Reporte del alumno.

Presentación del trabajo.

- a. **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- b. **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- c. **Desarrollará la definición sintética de conceptos y actividades implicados en cada fase del desarrollo de software.**
- d. **Redactara conclusiones personales respecto al tema desarrollado.**
- e. **Bibliografía**

Práctica Numero 4. Cuadro comparativo de los diferentes modelos de desarrollo de software.

Objetivo: Con el fin de poder tener una visión clara de sus diferencias específicas, sus ventajas comparativas en cuanto a los requerimientos de modelado por las características del proyecto y su gestión, el estudiante elaborara un cuadro comparativo de los diferentes modelos de desarrollo de software. Enlistando sus características básicas, sus ventajas y desventajas y el tipo de sistemas a los que puede aplicarse.

Introducción

La evolución de los sistemas informáticos, comienza hacia 1990 y se dirige al impacto colectivo de las computadoras y el software, en todos los entornos. La industria del software tiene un gran peso en la economía mundial.

A medida que se intenta incrementa la complejidad que abarcan las aplicaciones de software se incrementa el riesgo de errores costosos por fallas en el software, lo que se conoce como Crisis del software; haciendo más relevante el uso de procesos de modelado que permitan definir mejores modelos y procesos de desarrollo que promuevan la calidad de los productos de software, mejoren la productividad, eviten los riesgos, reduzcan costos, facilitar el abordar sistemas más grandes y complejos.

Correlación con los temas y subtemas del programa de estudio.

Relacionar los contenidos de esta asignatura con las demás del plan de estudios para desarrollar una visión interdisciplinaria en el estudiante.

Material y equipo necesario. Bibliografía, equipo de cómputo para editar el documento, enlace al servidor de la plataforma Moodle. Proyector para usarlo al exponer su trabajo en clase.

Metodología. En equipos de tres personas Investigaran, en al menos dos fuentes bibliográficas cuales son los modelos de desarrollo de software.

- Los estudiantes investigarán en equipos de máximo tres personas, designado en clase por el profesor.
- Presentarán de manera breve resultado de su trabajo y lo intercambiarán con otro estudiante o equipo de estudiantes para que la aplique y valore la corrección y claridad del trabajo recibido Logrando una evaluación colaborativa entre los equipos que intercambiaron trabajos. Toda la clase valorará el ejercicio realizado con el fin de tener una visión de los diferentes Modelos.

Preguntas de análisis en el aula: (trabajo individual, equipo y plenaria).

¿Consideras que el diseño de procesos de modelado coherentes a las necesidades de análisis y diseño de aplicaciones de software tiene impacto en la evolución de software?

¿Qué es relevante en el desarrollo de procesos de modelado de aplicaciones de SW?

¿Qué ha permitido definir mejores procesos de modelado y desarrollo de los productos de software?

¿Qué hace que los procesos de modelado de software promuevan la calidad del producto?

Sugerencias Didácticas.

- Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes.
- Observar y analizar fenómenos y problemáticas propias del campo ocupacional.
- Presentar los resultados de la investigación en forma oral y escrita haciendo énfasis en las conclusiones.

Reporte del alumno. Los estudiantes elaboraran un cuadro sintético de los resultados de la investigación bibliográfica realizada en equipo. Elaborando como parte del reporte una conclusión personal.

Presentación del trabajo.

- a) **En caso de presentarse en línea deberá ajustarse a la hora de entrega requerida por el profesor.**
- b) **Reporte escrito.** – Definirá por escrito de manera personal una conclusión a cerca de los resultados obtenidos y revisados en el grupo.
- c) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- d) **Se participará en la integración de conceptos implicados en el tema y se elaborará una matriz comparativa en el caso de ser requerida.**
- e) **Bibliografía y/o Cibergrafia.**

UNIDAD 2.- Ingeniería de Requerimientos de Software.

Competencia específica a desarrollar: Desarrollar las habilidades para identificar las diferentes técnicas que se aplican para la obtención de un modelo de requerimientos de para un determinado producto software.

Subtemas.

- 2.1. Tareas de la Ingeniería de Requisitos.
- 2.2. Técnicas de la Ingeniería de Requisitos.
- 2.3. Modelado de requisitos.
- 2.4. Herramientas CASE para la Ingeniería de requisitos.

Sugerencias Didácticas.

- Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes.
- Propiciar el uso de las nuevas tecnologías en el desarrollo de los contenidos de la asignatura.
- Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes.
- Propiciar, en el estudiante, el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, las cuales lo encaminan hacia la investigación, la aplicación de conocimientos y la solución de problemas.
- Observar y analizar fenómenos y problemáticas propias del campo ocupacional.
- Relacionar los contenidos de esta asignatura con las demás del plan de estudios para desarrollar una visión interdisciplinaria en el estudiante.
- Presentar los resultados de la investigación en forma oral y escrita haciendo énfasis en las conclusiones.

Práctica Numero 5. El estudiante elaborara un mapa mental de la ingeniería de requerimientos.

Objetivo: El estudiante desarrollara un mapa mental que defina los elementos de la ingeniería de requerimientos de software. Revisada la bibliografía de al menos dos fuentes distintas.

Introducción.

Para tener una visión inicial de lo que es el software y cómo se produce el estudiante de ingeniería de software debe profundizar en el concepto de ingeniería del software y de los conceptos que definen actualmente al SW como un producto o incluso como un artefacto y el porqué de su definición de software en función de requisitos de calidad exigibles.

El software, según su aplicación y modo de desarrollarlo puede ser muy diverso. Por tanto, se hayan definido distintos modelos de desarrollo de Software a lo largo del tiempo. La industria se ha propuesto abordar la solución a problemas diversos en tamaño complejidad y objetivos usando técnicas de modelado basada en definición de requisitos de la solución de software.

Correlación con los temas y subtemas del programa de estudio.

Tener la visión de Ingeniería de requerimientos para desarrollar modelos de solución fortalecerá las competencias del estudiante para abordar la definición de los requerimientos de software a proyectos de desarrollo como parte del aprendizaje de esta y las materias posteriores a esta, así como de su ejercicio profesional.

Material y equipo necesario. Bibliografía, equipo de cómputo para editar el documento, enlace al servidor de la plataforma Moodle. Proyector para usarlo al exponer su trabajo en clase.

Metodología.

- Los estudiantes investigarán en equipos de máximo tres personas, y diseñarán cada uno un mapa conceptual de la Ingeniería de requerimientos.
- Lo intercambiarán con otro estudiante de su equipo o equipo de estudiantes para que la aplique y valore la corrección y claridad del trabajo recibido Logrando una evaluación colaborativa entre los estudiantes y/o equipos que intercambiaron trabajos.
- Toda la clase valorará el ejercicio realizado con el fin de tener una visión del proceso elaboración de un modelo de requerimientos del producto software.

Preguntas de análisis en el aula: (Para trabajo individual, equipo y plenaria).
Analizaremos qué es lo que significa el análisis de requerimientos dentro de procesos de desarrollo de software

Particularmente queremos identificar:

¿Cuál es el punto de inicio en el desarrollo de un producto Software?,

¿Cómo traduzco las necesidades de los usuarios en los requerimientos del software?

¿Cómo documentamos el comportamiento requerido del software?

¿Cuál es la diferencia entre los requerimientos funcionales y los no funcionales?

Si ya he podido establecer los requerimientos funcionales del software, ¿Cómo defino la calidad que se necesita que tenga el Software?

Si ya he podido establecer los requerimientos de los datos que tienen que entrar al sistema, las operaciones que se realizan con nuestros datos, el flujo de trabajo que siguen esas operaciones

¿Cuál sería el impacto en la calidad del software si es uno de ellos no se cumple o simplemente no obtenemos un sistema dentro de esta caracterización?

¿Afecta la calidad? también ¿Qué es lo que significa identificar los atributos de calidad del problema?

Y finalmente ¿Cómo se lleva a cabo un proceso de modelado para poder recabar los requerimientos en un proceso de desarrollo de software?

Reporte del alumno.

Presentación del trabajo.

1. **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito. **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
2. **Diseño de un mapa mental del tema**
3. **bibliografía Preliminar.**

Práctica Número 6.- Desarrollo el de modelo de requerimientos de un caso sencillo propuesto en clase.

Objetivo: Que el estudiante considerando las diferentes tareas que corresponden a la ingeniería de requerimientos de Software las aplique a un caso sencillo como entrenamiento para aplicarlo a un caso que propondrá como proyecto final del curso. Logrando como resultado modelar los requerimientos para un producto software que proveerá la solución a un problema específico.

Deberá elegir las técnicas adecuadas al caso y documentar los resultados con Diagrama de Casos de Uso, Diagramas Lógicos y especificación de Requisitos.

Sugerencias Didácticas.

- Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes.
- Propiciar el uso de las nuevas tecnologías en el desarrollo de los contenidos de la asignatura.
- Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes.
- Propiciar, en el estudiante, el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, las cuales lo encaminan hacia la investigación, la aplicación de conocimientos y la solución de problemas.
- Observar y analizar fenómenos y problemáticas propias del campo ocupacional.
- Relacionar los contenidos de esta asignatura con las demás del plan de estudios para desarrollar una visión interdisciplinaria en el estudiante.
- Presentar los resultados de la investigación en forma oral y escrita haciendo énfasis en las conclusiones.

Material y equipo necesario.

Computadora 1 o más como equipo de realización de pruebas, Servicio de Internet, impresora, bibliografía, Proyector.

Metodología.

El estudiante se integrará en equipos (tres personas) y considerando las diferentes tareas que corresponden a la ingeniería de requerimientos de Software las aplicará a un caso sencillo como entrenamiento para aplicarlo después usando una herramienta case y posteriormente a un caso que propondrá como proyecto final para el curso. Siendo el trabajo a entregar un modelo requerimientos que cumpla con el estándar, para un producto software que proveerá la solución a un problema específico.

Los casos serán propuestos en clase por el profesor. Para solucionar este problema en específico recuerda que de todo lo visto lo más importante es que:

primero que nada, describes el problema en lenguaje natural antes de tratar de hacerlo de una forma técnica. Una vez hecho usaras las técnicas llenar los huecos para identificar el dominio o el alcance del problema, después resaltaras los hallazgos preferiblemente de forma gráfica y después validar el trabajo hecho de diseño de requisitos, directamente con el cliente.

Es recomendable ANTES DE QUE SE REALICE LA PRACTICA.

Sugerir al estudiante profundizar más en el tema, encontrará algo más de información en la bibliografía referida para el curso, y también aprender y entender primero que es un requerimiento y sobre las características de los requerimientos no funcionales y los requerimientos funcionales del sistema.

Discutir con ellos el hecho de que, mucha gente, bueno pues a los programadores siempre les da mucha, mucha, pereza ver lo que es eso de los requerimientos, hablar con el cliente saber qué es lo que pretende el cliente en un sistema, en un programa, en una página web, siempre o casi siempre nos cuesta mucho saber qué es un requerimiento ya sea funcional o no funcional si no tenemos disposición y apertura mental.

Desarrollar varios casos para que se valla viendo poco a poco estos conceptos y vayamos entendiendo qué la única forma para que un programador, un ingeniero sepa al trabajar un caso que es un requerimiento, es con la práctica.

Reporte del alumno.

Los estudiantes realizaran una plenaria en base a los trabajos realizados por cada equipo o estudiante si se trabajó de forma individual.

Presentación del trabajo.

- a) **Reporte escrito.** - Debe elaborarse un informe escrito de la práctica, que documente el modelo de requisitos del caso de estudio, el cual debe incluir una conclusión personal del trabajo presentado por escrito.
- b) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- c) **Conceptos implicados en el tema en el caso de ser requerida.**
- d) **Bibliografía**
- e) Exposición en clase de la solución del caso de estudio.

Práctica Número 7.- Uso de Herramientas CASE, para desarrollar el modelo de requisitos para un caso de estudio usando la herramienta StarUML.

Objetivo: Que el estudiante aplique una herramienta case, en este caso StarUML, considerando las diferentes tareas que corresponden a la ingeniería de requerimientos de Software, las aplique a un caso sencillo como entrenamiento para aplicarlo a un caso que propondrá como proyecto final del curso. Logrando como resultado modelar los requerimientos para un producto software que proveerá la solución a un problema específico. Deberá elegir las técnicas adecuadas al caso y documentar los resultados con Diagrama de Casos de Uso, Diagramas Lógicos y especificación de Requisitos. Documentando además los requerimientos funcionales y no funcionales de la solución que proveerá el Software que se modelará.

El estudiante pueda desarrollar el modelo de requisitos para todos sus casos usando la herramienta StarUML después de realizar esta práctica.

Material y equipo necesario.

Computadora 1 o más como equipo de realización de pruebas, Servicio de Internet, impresora, bibliografía, Proyector.

Introducción: El primer objetivo consiste principalmente en facilitar el proceso de identificación de requerimientos del Software documentándolo mediante un proceso automático asistido por un software.

Después de describir de forma natural el problema junto a los actores involucrados los cuales serán importantes más adelante cuando necesitemos fuentes de información en la indagación y queramos validar que estamos entendiendo de verdad el problema. Determinar hasta qué punto tenemos que ocuparnos del problema con nuestro sistema. Es decir, delimitarlo.

Material y equipo necesario.

Bibliografía, SW CASE StarUML, Computadora, impresora, Proyector.

Metodología. Basándose en un ejercicio resuelto en clase donde se cumplió con la especificación de requerimientos, el estudiante deberá explorar el uso de una herramienta case para la documentación de requisitos de software el cual presentará impreso como documentación y enviará a la liga correspondiente al curso en la plataforma Moodle a un foro abierto para que todos puedan visitarlo y conocer los resultados logrados por todos los estudiantes.

Reforzando la idea de desarrollar una cultura de trabajo en equipo y colaboración abierta. De forma que después pueda aplicar lo aprendido a un caso propuesto por su equipo como proyecto final.

Reporte del alumno.

Los estudiantes realizarán una plenaria en base a los trabajos realizados por cada equipo o estudiante si se trabajó de forma individual.

Presentación del trabajo.

- f) Debe elaborarse un informe escrito de la práctica, en caso de haberse usado más de una herramienta case por los diferentes estudiantes o equipos de trabajo presentarse una tabla que incluya el contraste **de las características más relevantes del uso de la herramienta usada y las presentadas en cada investigación por el grupo**, exponiendo los resultados y **contrastar las características de cada herramienta.**
- g) **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- h) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- i) **Conceptos implicados en el tema y matriz comparativa en el caso de ser requerida.**
- j) **Preguntas de análisis en el aula.** Redacción de Preguntas y Respuestas planteadas en el trabajo desarrollado en equipo o durante la presentación en el aula.
- k) **Bibliografía**
- l) **Debe hacerse una presentación breve ajustada a un tiempo designado, 10 min. por ejemplo, con el fin de cumplir con los requisitos para la presentación del trabajo realizado en la práctica en el reporte escrito.**

UNIDAD TRES. Modelo de Análisis.

Objetivo específico de la Unidad. Identificar a través de un modelo de requisitos la arquitectura de clases que participarán en el diseño del producto.

Subtemas.

- 3.1. Arquitectura de clases
- 3.2. Identificación de clases según Estereotipos.
- 3.3. Clases
- 3.4. Diagramas de secuencias
- 3.5. Diccionario de clases según Módulos
- 3.6. Herramientas CASE para el análisis

Actividades de Aprendizaje

. • • Aplicar al menos una herramienta CASE para el análisis. • Parte 1 del proyecto:
A. Identificación y delimitación del problema B. Propuesta de solución

Sugerencias Didácticas.

Práctica Número 8.- Modelo de Análisis de un producto de software.

Objetivo: Basándose en la especificación de requerimientos para un producto de software preferentemente de un ejercicio resuelto en clase aplique el modelo objeto-relación comportamiento para indicar como responderá el sistema de Software OO mediante el uso de diagramas tales como: Modelo CRC, Modelo Objeto-Relación, atributos, operaciones y colaboradores. De forma que después pueda extrapolar lo aprendido a un caso real propuesto por su equipo para el proyecto final.

Material y equipo necesario. Bibliografía, SW CASE StarUML, Computadora, impresora, Proyector.

Metodología.

Que el estudiante pueda al aplicar el modelo objeto-relación comportamiento indicar como responderá el sistema de Software OO a partir de una especificación técnica de los requisitos y representarlos a mediante el uso de diagramas tales como: Modelo CRC, Modelo Objeto-Relación, atributos, operaciones y colaboradores. De forma que después pueda aplicar lo aprendido a un caso real propuesto por su equipo para el proyecto final.

el estudiante deberá desarrollar el modelo de análisis de análisis correspondiente, el cual presentará como especificación técnica mediante el uso de diagramas tales como: **Modelo CRC, Modelo Objeto-Relación, atributos, operaciones y colaboradores.**

El trabajo impreso y documentado le enviará a la liga correspondiente en la plataforma Moodle para el curso de Fundamentos de Ingeniería de Software. compartiendo los resultados en el grupo mediante su exposición o en un foro de la plataforma digital del curso para propiciar el aprendizaje por los trabajos hechos por el resto del grupo.

Preguntas de análisis en el aula: (trabajo individual, equipo y plenaria).

Documentar las preguntas auto formuladas que fueron de utilidad para desarrollar el modelo de arquitectura del producto software.

Reporte del alumno.

Los estudiantes realizaran una plenaria en base a los trabajos realizados por cada equipo o estudiante si se trabajó de forma individual.

Presentación del trabajo.

- a) Debe elaborarse un informe escrito de la práctica, en caso de haberse usado más de una herramienta case por los diferentes estudiantes o equipos de trabajo presentarse una tabla que incluya el contraste **de las características más relevantes del uso de la herramienta usada y las presentadas en cada investigación por el grupo**, exponiendo los resultados **y contrastar las características de cada herramienta.**
- b) **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- c) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- d) **Conceptos implicados en el tema y matriz comparativa en el caso de ser requerida.**
- e) **Preguntas de análisis en el aula.** Redacción de Preguntas y Respuestas planteadas en el trabajo desarrollado en equipo o durante la presentación en el aula.
- f) **Bibliografía**
- g) **Debe hacerse una presentación breve ajustada a un tiempo designado, 10 min. por ejemplo, con el fin de cumplir con los requisitos para la presentación del trabajo realizado en la práctica en el reporte escrito.**

Práctica Número 9.-Identificación de clases según estereotipos.

Objetivo: Que el estudiante pueda clasificar las clases del modelo conceptual dentro de una categoría o estereotipo (funcionalidad y borde) identificando cuales cambios se presentan en objetos borde y cuales en objetos que gestionan la funcionalidad, con el fin de realizar la transición del modelo de requisitos al modelo de análisis al identificar todos los objetos necesarios para implementar todos los casos de uso

Material y equipo necesario. Bibliografía, SW CASE StarUML, Computadora, impresora, Proyector.

Metodología. Basándose en un ejercicio resuelto en clase que cumplió con la especificación de requerimientos el estudiante deberá desarrollar el modelo de análisis para el desarrollo de un producto de software,

El estudiante clasificará las clases del modelo conceptual dentro de una categoría o estereotipo (funcionalidad y borde) identificando cuales cambios se presentan en el sistema y que objetos intervienen en dichos cambios, categorizándolos como **objeto borde** cuando tengan que ver con cambios en la interfaz de usuario. Además de identificar cuales objetos que gestionan la funcionalidad, y los categorizara como **objetos de funcionalidad** (modelo de negocio), para todos y cada uno de los casos de uso del modelo de requisitos. Con el fin de realizar la transición del modelo de requisitos al modelo de análisis.

El resultado se presentará impreso como documentación y enviará a la liga correspondiente al curso en la plataforma Moodle.

De forma que después pueda aplicar lo aprendido al caso propuesto por su equipo para el proyecto final, compartiendo en el grupo los resultados para aprender de los trabajos hechos por el resto del grupo.

Preguntas de análisis en el aula: (trabajo individual, equipo y plenaria).

Documentar las preguntas auto formuladas que fueron de utilidad para desarrollar el modelo de arquitectura del producto software.

Sugerencias Didácticas.

Reporte del alumno.

Los estudiantes realizaran una plenaria en base a los trabajos realizados por cada equipo o estudiante si se trabajó de forma individual.

Presentación del trabajo.

- a) Debe elaborarse un informe escrito de la práctica, en caso de haberse usado más de una herramienta case por los diferentes estudiantes o equipos de trabajo presentarse una tabla que incluya el contraste **de las características más relevantes del uso de la herramienta usada y las presentadas en cada investigación por el grupo**, exponiendo los resultados **y contrastar las características de cada herramienta.**
- b) **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- c) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- d) **Conceptos implicados en el tema y matriz comparativa en el caso de ser requerida.**
- e) **Preguntas de análisis en el aula.** Redacción de Preguntas y Respuestas planteadas en el trabajo desarrollado en equipo o durante la presentación en el aula.
- f) **Bibliografía**
- g) **Debe hacerse una presentación breve ajustada a un tiempo designado, 20 min. por ejemplo, con el fin de cumplir con los requisitos para la presentación del trabajo realizado en la práctica en el reporte escrito.**

Práctica Número 10.- Clases, Diagramas de secuencias, Diccionario de clases según Módulos.

Objetivo: Basándose en el ejercicio resuelto de las dos prácticas anteriores, el estudiante deberá desarrollar los Diagramas de secuencias, Diccionario de clases según Módulos para concretar el modelo de análisis para el desarrollo de un producto de software.

Introducción. El diccionario de clases según módulos, es un catálogo, un depósito de los elementos en un sistema. Dichos elementos se centran alrededor de los datos y la forma en que están estructurados para satisfacer los requerimientos de usuarios y la forma de satisfacer las necesidades de la organización.

Material y equipo necesario. Bibliografía, SW CASE StarUML, Computadora, impresora, Proyector.

Metodología.

El estudiante clasificará las clases del modelo conceptual dentro de una categoría o estereotipo (funcionalidad y borde) identificando cuales cambios se presentan en el sistema y que objetos intervienen en dichos cambios, categorizándolos como **objeto borde** cuando tengan que ver con cambios en la interfaz de usuario. Además de identificar cuales objetos que gestionan la funcionalidad, y los categorizara como **objetos de funcionalidad** (modelo de negocio), para todos y cada uno de los casos de uso del modelo de requisitos. Con el fin de realizar la transición del modelo de requisitos al modelo de análisis.

El resultado se presentará impreso como documentación y enviará a la liga correspondiente al curso en la plataforma Moodle.

De forma que después pueda aplicar lo aprendido al caso propuesto por su equipo para el proyecto final, compartiendo en el grupo los resultados para aprender de los trabajos hechos por el resto del grupo.

Preguntas de análisis en el aula: (trabajo individual, equipo y plenaria).

Documentar las preguntas auto formuladas que fueron de utilidad para desarrollar el modelo de arquitectura del producto software.

Sugerencias Didácticas.

Reporte del alumno.

Los estudiantes realizaran una plenaria en base a los trabajos realizados por cada equipo o estudiante si se trabajó de forma individual.

Presentación del trabajo.

- a) Debe elaborarse un informe escrito de la práctica, en caso de haberse usado más de una herramienta case por los diferentes estudiantes o equipos de trabajo presentarse una tabla que incluya el contraste **de las características más relevantes del uso de la herramienta usada y las presentadas en cada investigación por el grupo**, exponiendo los resultados **y contrastar las características de cada herramienta.**
- b) **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- c) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- d) **Conceptos implicados en el tema y matriz comparativa en el caso de ser requerida.**
- e) **Preguntas de análisis en el aula.** Redacción de Preguntas y Respuestas planteadas en el trabajo desarrollado en equipo o durante la presentación en el aula.
- f) **Bibliografía**
- g) **Debe hacerse una presentación breve ajustada a un tiempo designado, 10 min. por ejemplo, con el fin de cumplir con los requisitos para la presentación del trabajo realizado en la práctica en el reporte escrito.**

Práctica Número 11.-Ejercicio de Modelado de una Revisión Estructurada.

Objetivo: El estudiante participará en un ejercicio de revisión cruzada (revisión entre pares) de los modelos de análisis presentados por los equipos para su proyecto final. Clases, Diagramas de secuencias, Diccionario de clases según Módulos.

que el estudiante pueda modelar la actividad de una revisión estructurada entre colegas; donde la forma de revisar el "producto de software". Tal como es indicado por la definición de la IEEE, normalmente se refiere a un tipo de documento técnico esto puede ser un documento de diseño de software o código fuente de un programa, pero también casos de uso, definiciones del proceso de negocios, especificaciones de casos de prueba y una variedad de otra documentación técnica también puede ser revisada.

NOTA: de acuerdo a la definición de esta herramienta de revisión, el maestro podrá elegir realizarla en cualquier otra fase previa, para que los estudiantes la hagan o enriquezcan la forma intuitiva en que ya la hacen.

Material y equipo necesario. Bibliografía, SW CASE StarUML, Computadora, impresora, Proyector.

Sugerencias didácticas. La práctica de la disciplina.

Metodología. Basándose en el ejercicio de diseño resuelto, sea un caso propuesto o el caso del proyecto final, se trabajara de acuerdo a los equipos de trabajo para modelar y practicar una revisión estructurada de software El equipo de trabajo Identificará un diseñador o programador que lidera a los miembros de un equipo de desarrollo y otra de las partes involucradas en el desarrollo de un producto de software(usuarios, clientes, etc.), para lo cual puede integrarse a un cliente o usuario "real" o pedirle a los miembros de otro equipo que jueguen ese papel, además de integrar otro equipo que observe(registre) el proceso. y los participantes hacen preguntas y comentarios acerca de posibles errores, violación de estándares de desarrollo, y otros problemas.

Redactando un documento que registre los hallazgos para llevarlos a la adecuación del producto software, es decir el documento técnico. (este documento es independiente del registro del equipo observador, el cual propiamente registra la dinámica de la integración y la forma en que se asumen los roles para el ejercicio).

El equipo observador deberá investigar la forma en que se hace una revisión técnica de software que difiere de una revisión de software estructurada en la forma abierta de ser estructura y su objetivo de familiarización.

De forma que después pueda aplicar el ejercicio compartiendo en el grupo los resultados para aprender de los trabajos hechos por el resto del grupo.

Preguntas de análisis en el aula: (trabajo individual, equipo y plenaria).

Documentar las preguntas auto formuladas que fueron de utilidad para desarrollar la práctica.

Reporte del alumno.

Los estudiantes realizarán una plenaria en base a los trabajos realizados por cada equipo o estudiante si se trabajó de forma individual.

Presentación del trabajo.

- a) Debe elaborarse un informe escrito de la práctica, de acuerdo con lo planteado en el apartado Metodología, el equipo que hace revisión a su producto y el equipo observador entregan reporte, si otro equipo participa para aportar el desempeño de roles, solo puede agregar opiniones en el informe del equipo observador.
- b) **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- c) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- d) **Conceptos implicados en el tema y matriz comparativa en el caso de ser requerida.**
- e) **Preguntas de análisis en el aula.** Redacción de Preguntas y Respuestas planteadas en el trabajo desarrollado en equipo o durante la presentación en el aula.
- f) **Bibliografía**
- g) **Debe hacerse una presentación breve ajustada a un tiempo designado.**

UNIDAD CUATRO. Modelo de Diseño.

Objetivo específico de la unidad. Aplicar modelos, técnicas y herramientas para la etapa de diseño del software.

Subtemas.

- 4.1. Estrategias de diseño
- 4.2. Diseño de objetos
- 4.3. Diseño de sistema
- 4.4. Revisión del diseño
- 4.5. Diagramas de Secuencias del Diseño
- 4.6. Herramientas CASE para el diseño

Sugerencias Didácticas. Disciplinarias.

Práctica Número 12.- Estrategias de diseño. Diseño del Sistema y Revisión de Diseño.

Objetivo: Que el estudiante a partir del modelo de análisis del proyecto propuesto aplicando el modelo objeto-relación-comportamiento que indica cómo responderá el sistema OO a eventos. Realice las siguientes actividades:

Refinamiento a clases.

Refinamiento a subsistemas.

Refinamiento a diagramas de colaboración.

Refinamiento a diagramas de componentes.

Refinamiento a diagramas de actividades.

Refinamiento a diagrama de secuencia.

Realizará una tabla comparativa que muestre las inconsistencias detectadas.

Para terminar, realizando el diseño de la estructura del sistema

después de haber realizado el modelo de diseño en el caso de estudio.

- Utilizando StarUML para redactar el Reporte técnico del diseño cumpliendo el requisito de aplicar al menos una herramienta CASE para el diseño.

Introducción. La fase de diseño (y los modelos UML resultantes) expande y detalla los modelos de análisis tomando en cuenta todas las implicaciones y restricciones técnicas. El propósito del diseño es especificar una solución que trabaje y pueda ser fácilmente convertida en código fuente y construir una arquitectura simple y fácilmente extensible.

Material y equipo necesario. Bibliografía, SW CASE StarUML, Computadora, impresora, Proyector.

Metodología.

Basándose en un ejercicio resuelto en clase que cumplió con la especificación de requerimientos el estudiante deberá desarrollar el modelo de análisis para el desarrollo de un producto de software, el cual presentará impreso como documentación y enviará a la liga correspondiente al curso en la plataforma Moodle.

De forma que después pueda aplicar lo aprendido al caso propuesto por su equipo para el proyecto final, compartiendo en el grupo los resultados para aprender de los trabajos hechos por el resto del grupo.

Preguntas de análisis en el aula: (trabajo individual, equipo y plenaria).

Documentar las preguntas auto formuladas que fueron de utilidad para desarrollar el modelo de arquitectura del producto software.

Sugerencias Didácticas. Práctica de la disciplina.

Reporte del alumno.

Los estudiantes realizaran una plenaria en base a los trabajos realizados por cada equipo o estudiante si se trabajó de forma individual.

Presentación del trabajo.

- a) Debe elaborarse un informe escrito de la práctica, en caso de haberse usado más de una herramienta case por los diferentes estudiantes o equipos de trabajo presentarse una tabla que incluya el contraste **de las características más relevantes del uso de la herramienta usada y las presentadas en cada investigación por el grupo**, exponiendo los resultados **y contrastar las características de cada herramienta.**
- b) **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- c) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- d) **Conceptos implicados en el tema y matriz comparativa en el caso de ser requerida.**
- e) **Preguntas de análisis en el aula.** Redacción de Preguntas y Respuestas planteadas en el trabajo desarrollado en equipo o durante la presentación en el aula.
- f) **Bibliografía**
- g) **Debe hacerse una presentación breve ajustada a un tiempo designado, 10 min. por ejemplo, con el fin de cumplir con los requisitos para la presentación del trabajo realizado en la práctica en el reporte escrito.**

Práctica Número 13.- Diagrama de secuencias del Diseño.

Objetivo: El estudiante desarrollara la documentación técnica del prototipo con el fin de facilitar la realización de este y el desarrollo conjunto de la aplicación.

Material y equipo necesario. Bibliografía, SW CASE StarUML, Computadora, impresora, Proyector.

Sugerencias didácticas. La práctica de la disciplina.

Metodología. Para mejora y estandarizar la documentación usara la herramienta case StarUML, aumentando la portabilidad de la aplicación y el reusó de componentes y permitir un refinamiento visual de las aplicaciones, la utilización de gráficos

Diseñara el diagrama de mediante secuencia para modelar la interacción entre objetos en un sistema para cada caso de uso. Refinando las vistas de los escenarios propuestos por los casos de uso para diseñar su implementación, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

En el diagrama de secuencias existen dos tipos de mensajes: sincrónicos y asincrónicos. De acuerdo con cuyo tipo se define la forma de su ejecución y su implicación a nivel de instanciación de los objetos.

Basándose en un ejercicio resuelto para el proyecto final presentará impreso como documentación y enviará a la liga correspondiente al curso en la plataforma Moodle.

Preguntas de análisis en el aula: (trabajo individual, equipo y plenaria).

Documentar las preguntas auto formuladas que fueron de utilidad para desarrollar el modelo de arquitectura del producto software.

Reporte del alumno.

Los estudiantes realizaran una plenaria en base a los trabajos realizados por cada equipo o estudiante si se trabajó de forma individual.

Presentación del trabajo.

- a) Debe elaborarse un informe escrito de la práctica que se integrara al modelo de Diseño del caso de estudio.
- b) **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- c) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- d) **Conceptos implicados en el tema y matriz comparativa en el caso de ser requerida.**

- e) **Preguntas de análisis en el aula.** Redacción de Preguntas y Respuestas planteadas en el trabajo desarrollado en equipo o durante la presentación en el aula.
- f) **Bibliografía**
- g) **Debe hacerse una presentación breve ajustada a un tiempo designado, 10 min. por ejemplo, con el fin de cumplir con los requisitos para la presentación del trabajo realizado en la práctica en el reporte escrito.**

Práctica Número 14.- Revisión del Diseño.

Objetivo: que el estudiante pueda modelar la actividad de una revisión técnica; distinguiéndola de una revisión estructurada que es abierta y exploratoria.

NOTA: de acuerdo a la definición de esta herramienta de revisión, el maestro podrá elegir realizarla en cualquier otra fase previa, para que los estudiantes la hagan o enriquezcan la forma intuitiva en que ya la hacen.

Material y equipo necesario. Bibliografía, SW CASE StarUML, Computadora, impresora, Proyector.

Sugerencias didácticas. La práctica de la disciplina.

Metodología.

que el estudiante pueda modelar la actividad de una revisión técnica; distinguiéndola de una revisión estructurada que es abierta y exploratoria.

Basándose en el ejercicio de diseño resuelto, sea un caso propuesto o el caso del proyecto final, se trabajará de acuerdo a los equipos de trabajo para modelar y practicar una revisión técnica la cual se enfocará en la verificación de cumplimiento de las especificaciones del producto, donde la forma de revisar el "producto de software". Tal como es indicado por la definición de la IEEE, normalmente se refiere a un tipo de documento técnico esto puede ser un documento de diseño de software o código fuente de un programa, pero también casos de uso, definiciones del proceso de negocios, especificaciones de casos de prueba y una variedad de otra documentación técnica también puede ser revisada.

El grupo completo modelara la participación de los equipos para desarrollar una revisión técnica a cada uno de los proyectos finales desarrollados por los estudiantes a lo largo del curso.

El equipo de trabajo Identificará un diseñador o programador que lidera a los miembros de un equipo de desarrollo y otra de las partes involucradas en el desarrollo de un producto de software (usuarios, clientes, etc.), para lo cual puede integrarse a un cliente o usuario "real" o pedirles a los miembros de otro equipo que jueguen ese papel, además de integrar otro equipo que observe(registre) el proceso. y los participantes hacen preguntas y comentarios acerca de posibles errores, violación de estándares de desarrollo, y otros problemas.

Redactando un documento que registre los hallazgos para llevarlos a la adecuación del producto software, es decir el documento técnico. (este documento

es independiente del registro del equipo observador, el cual propiamente registra la dinámica de la integración y la forma en que se asumen los roles para el ejercicio).

El equipo observador deberá investigar la forma en que se hace una revisión técnica de software que difiere de una revisión de software estructurada en la forma abierta de ser estructura y su objetivo de familiarización.

De forma que después pueda aplicar el ejercicio compartiendo en el grupo los resultados para aprender de los trabajos hechos por el resto del grupo.

Preguntas de análisis en el aula: (trabajo individual, equipo y plenaria).

Documentar las preguntas auto formuladas que fueron de utilidad para desarrollar la práctica.

Reporte del alumno.

Los estudiantes realizaran una plenaria en base a los trabajos realizados por cada equipo o estudiante si se trabajó de forma individual.

Presentación del trabajo.

- h) Debe elaborarse un informe escrito de la práctica, de acuerdo con lo planteado en el apartado Metodología, el equipo que hace revisión a su producto y el equipo observador entregan reporte, si otro equipo participa para aportar el desempeño de roles, solo puede agregar opiniones en el informe del equipo observador.
- i) **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- j) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- k) **Conceptos implicados en el tema y matriz comparativa en el caso de ser requerida.**
- l) **Preguntas de análisis en el aula.** Redacción de Preguntas y Respuestas planteadas en el trabajo desarrollado en equipo o durante la presentación en el aula.
- m) **Bibliografía**
- n) **Debe hacerse una presentación breve ajustada a un tiempo designado.**

UNIDAD CINCO. Modelo de Implementación.

Objetivo específico de la unidad. Identificar procesos de la fase de implementación.

Subtemas.

5.1. Diagrama de componentes

5.2. Diagrama de despliegue

5.3. Modelos de pruebas

Practica Numero 15.- Identificar procesos de la fase de implementación para un producto software diseñado.

Objetivo: Que el estudiante pueda identificar el proceso de implementación de un producto de software. Específicamente deberá hacerlo para el diseño elaborado con el caso propuesto como proyecto final. Realizando las especificaciones técnicas se harán mediante; Diagrama de componentes Diagrama de despliegue. Modelos de pruebas.

Material y equipo necesario. Bibliografía, SW CASE StarUML, Computadora, impresora, Proyector.

Sugerencias didácticas. La práctica de la disciplina.

Metodología. Basándose en un ejercicio resuelto en clase que cumplió con la especificación de requerimientos, análisis y diseño; el estudiante deberá identificar procesos de la fase de implementación como parte del desarrollo de un producto de software, el cual presentará impreso como documentación y enviará a la liga correspondiente al curso en la plataforma Moodle.

Este ejercicio puede hacerse con un ejercicio propuesto en clase de forma que después pueda aplicar lo aprendido al caso propuesto por su equipo para el proyecto final y/o hacerse directamente con el proyecto final y utilizar una herramienta case, aquí se propone StarUML.

El resultado de la práctica se integra a la documentación del desarrollo del proyecto concretándose la documentación del proceso completo de desarrollo de un producto de software.

Deberá presentarse mediante exposición en el aula los resultados de los trabajos hechos por todo del grupo, con el fin de **enriquecer la experiencia de aprendizaje de todos los estudiantes del curso.**

Preguntas de análisis en el aula: (trabajo individual, equipo y plenaria).

Documentar las preguntas formuladas que fueron de utilidad para reflexionar sobre el trabajo realizado a la hora de la exposición de forma que permita elaborar una conclusión del trabajo final realizado y el producto software obtenido.

Reporte del alumno.

Los estudiantes realizaran una plenaria en base a los trabajos realizados por cada equipo o estudiante si se trabajó de forma individual.

Presentación del trabajo.

- a) Debe elaborarse un informe escrito de la práctica, en caso de haberse usado más de una herramienta case por los diferentes estudiantes o equipos de trabajo presentarse una tabla que incluya el contraste **de las características más relevantes del uso de la herramienta usada y las presentadas en cada investigación por el grupo**, exponiendo los resultados **y contrastar las características de cada herramienta.**
- b) **Reporte escrito.** - debe incluir una conclusión personal del trabajo presentado por escrito.
- c) **Portada:** Nombre del instituto, nombre del alumno, carrera, materia y horario, Nombre del maestro y fecha de entrega del trabajo).
- d) **Conceptos implicados en el tema y matriz comparativa en el caso de ser requerida.**
- e) **Preguntas de análisis en el aula.** Redacción de Preguntas y Respuestas planteadas en el trabajo desarrollado en equipo o durante la presentación en el aula.
- f) **Bibliografía**

Debe hacerse una presentación del producto software realizado a lo largo del curso, con el fin de cumplir con los requisitos para la presentación del trabajo realizado en la práctica en el reporte escrito.

Bibliografía del Curso.

Booch G. El lenguaje Unificado de Modelado, UML 2.0, Guía de Usuario. 1ª. Edición. Ed. Pearson ADDISON-WESLEY. España. 2006.

Cota, A. Ingeniería de Software: Soluciones Avanzadas. 2ª. Edición. Ed. Oxford. México. 2000.

Fowler M. UML Gota a Gota. 1ª. Edición. Ed. Pearson. México. 2000.

Jacobson, I., Booch, G., Rumbaugh, J. El proceso unificado de desarrollo de software. Addison Wesley. España. 2003.

Kendall E. K., Análisis y Diseño de sistemas. 1ª. Edición. Prentice Hall. México. 2005.

Pressman, R.S. Ingeniería del Software un enfoque práctico. México. Mc GrawHill. Madrid, España. 2008.

Senn J.A. Análisis y Diseño de sistemas. 2ª Edición, Mc Graw Hill, México, 1996.