

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura: Fundamentos de desarrollo de sistemas
Carrera: Ingeniería en Sistemas Computacionales
Clave de la asignatura: SCM - 0413
Horas teoría-horas práctica-créditos 3-2-8

2.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Toluca del 18 al 22 agosto 2003.	Representantes de la academia de sistemas y computación de los Institutos Tecnológicos.	Reunión nacional de evaluación curricular de la carrera de Ingeniería en Sistemas Computacionales.
Institutos Tecnológicos de: Hermosillo, Tehuacan, Toluca 23 agosto al 7 noviembre del 2003	Academia de sistemas y computación.	Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación.
Instituto Tecnológico de León 1 – 5 marzo 2004	Comité de consolidación de la carrera de Ingeniería en Sistemas Computacionales.	Definición de los programas de estudio de la carrera de Ingeniería en Sistemas Computacionales.

3.- UBICACIÓN DE LA ASIGNATURA

a). Relación con otras asignaturas del plan de estudio

Anteriores		Posteriores	
Asignaturas	Temas	Asignaturas	Temas
Administración.	La organización.	Planificación y modelado.	Planificación del sistema, análisis del sistema.
Taller de bases de datos.	Manejo del modelo relacional y OO.		
Desarrollo sustentable.	Todos por ser una materia formadora.		
Ética.			

b). Aportación de la asignatura al perfil del egresado

- Conoce los elementos básicos para el análisis, diseño, implantación y gestión de software.
- Identifica un panorama general de las normas, estándares, métodos y herramientas correspondientes a las tecnologías de información.

4.- OBJETIVO(S) GENERAL(ES) DEL CURSO

Proporcionará los conocimientos y el desarrollo de habilidades que le permitan proponer y analizar los elementos básicos del desarrollo de un producto de software.

5.- TEMARIO

Unidad	Temas	Subtemas
1	Conceptos Introdutorios.	<ul style="list-style-type: none">1.1 Introducción a los sistemas.<ul style="list-style-type: none">1.1.1 Descripción general.1.1.2 Tipos.1.1.3 Clasificación.1.2 Ciclo de vida de un proyecto de software.<ul style="list-style-type: none">1.2.1 Planificación y gestión del proyecto.1.2.2 Determinación de requerimientos.1.2.3 Análisis y diseño.1.2.4 Programación.1.2.5 Pruebas e Implementación.
2	Introducción a la ingeniería de software.	<ul style="list-style-type: none">2.1 Definición de ingeniería de software.2.2 Historia de la ingeniería de software.2.3 Características del software.2.4 Mitos del software.2.5 Capas de la ingeniería de software.2.6 El proceso del software.2.7 Software de alta calidad.2.8 Factores de calidad y productividad.
3	Paradigmas de la ingeniería de software.	<ul style="list-style-type: none">3.1 El enfoque estructurado.<ul style="list-style-type: none">3.1.1 Diagramas de flujos de datos.3.1.2 Diccionarios de datos.3.1.3 Diseño de módulos.3.1.4 Descomposición en procesos.3.2 El enfoque orientado a objetos.<ul style="list-style-type: none">3.2.1 Análisis.3.2.2 Diseño.
4	Modelos de proceso de software.	<ul style="list-style-type: none">4.1 Modelo de cascada.4.2 Modelo de espiral.4.3 Modelo incremental.4.4 Proceso de desarrollo unificado.4.5 Proceso software personal.

5.- TEMARIO (Continuación)

5	Técnicas , herramientas y estudios previos.	5.1 Técnicas de recopilación de información. 5.1.1 Entrevista. 5.1.2 Cuestionario. 5.1.3 Recopilación y análisis de documentos. 5.1.4 Observación y técnica "STROBE". 5.2 Herramientas CASE. 5.2.1 Estructuradas. 5.2.2 Orientadas a Objetos. 5.3 Desarrollo de prototipos.
6	Diseño y arquitectura de productos de software.	6.1 Descomposición modular. 6.2 Arquitecturas de dominio específico. 6.2.1 Diseño de software de arquitectura multiprocesador. 6.2.2 Diseño de software de Arquitectura Cliente/Servidor 6.2.3 Diseño de software distribuido 6.2.4 Diseño de software de tiempo real.

6.- APRENDIZAJES REQUERIDOS

- Conocer los tipos de organización así como sus estructuras y funciones correspondientes.
- Aplicar elementos de Bases de datos.

7.- SUGERENCIAS DIDÁCTICAS

- Recopilar información en una organización para desarrollar un producto de software.
- Aplicar técnicas de adquisición de datos (entrevistas, cuestionarios, sondeos, entre otros).
- Exponer en el aula el análisis de la información recopilada.
- Integrar equipos de desarrollo motivando el aprendizaje en equipo.
- Investigar sobre Herramientas CASE de ultima generación.
- Uso de simuladores pertinentes.

8.- SUGERENCIAS DE EVALUACIÓN

- Valoración de conocimientos previos.
- Dar seguimiento al desempeño en el desarrollo del programa.
 - Elaboración y validación de técnicas de adquisición de datos.
 - Aplicación de los instrumentos.
 - Obtención y análisis de la información.
- Presentación y defensa de la congruencia de la información obtenida mediante la documentación de:
 - Diagramas de flujos de datos.
 - Dicionarios de datos.
 - Diseño de módulos.
- Participación en dinámicas grupales (mesas redondas, conferencias, lluvia de ideas, debate, entre otras).
- Actividades de auto evaluación.
- Exámenes prácticos.
- Exámenes departamentales.

9.- UNIDADES DE APRENDIZAJE

UNIDAD 1.- Conceptos introductorios.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Estudiante identificará los diferentes tipos de sistemas de software que existen y comprenderá las fases del ciclo de vida de un proyecto de software.	1.1 Realizar una síntesis sobre los elementos, tipos y clasificación de sistemas. 1.2 Discutir, en equipo, las características que diferencian a cada uno de las fases del ciclo de vida de un proyecto de software.	1

UNIDAD 2.- Introducción a la ingeniería de software.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Comprenderá los elementos que integran la Ingeniería de Software y el aseguramiento de la calidad.	2.1 2.5 Realizar una síntesis sobre los características, mitos, elementos y capas de la ingeniería de software. 2.6 2.8 Discutir, en equipo, la importancia de usar normas para el aseguramiento de la calidad durante el proceso de software.	1, 2, 3, 4,5,6,7

UNIDAD 3.- Paradigmas de la ingeniería de software.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Comprenderá la diferencia de aplicar un enfoque estructurado vs. orientado a objetos en el desarrollo de un proyecto de software.	3.1 3.2 Discutir, en equipo, las características que diferencian a cada uno de los paradigmas <ul style="list-style-type: none">• Estructurado.• Orientado a Objetos. Realizar un ejercicio que muestre la aplicación de cada uno de los paradigmas.	1, 2, 3, 4

UNIDAD 4.- Modelos de proceso de software.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Identificará los diferentes modelos de proceso que se aplican en el desarrollo de software.	<p>4.1 Discutir, en equipo, las características que diferencian a cada uno de los modelos de desarrollo de proceso de software:</p> <ul style="list-style-type: none"> • Modelo de cascada. • Modelo de espiral. • Modelo incremental. • Proceso de desarrollo unificado. <p>4.2 Proceso software personal.</p>	2,3,4,5,8

UNIDAD 5.- Técnicas , herramientas y estudios previos.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Diseñará esquemas relacionales de base de datos.	<p>5.1 5.2 Discutir en equipo, las características de:</p> <ul style="list-style-type: none"> • Técnicas de recopilación de información. • Herramientas CASE. <p>5.1 5.2 Realizar un ejercicio aplicando las técnicas de recopilación de información haciendo uso de una herramienta CASE.</p> <p>5.3 Discutir, en equipo, la importancia de desarrollar prototipos para involucrar al usuario en el desarrollo del producto de software.</p>	1, 2, 3, 4, 5,12

UNIDAD 6.- Diseño y arquitectura de productos de software.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Comprenderá las arquitecturas en el diseño de software dependiendo del tipo de dominio de la aplicación.	<p>6.1 6.2 Discutir en equipo, las características de cada una de las arquitecturas.</p> <p>6.2.1 Arquitectura multiprocesador.</p> <p>6.2.2 Arquitectura Cliente/Servidor.</p> <p>6.2.3 Sistemas distribuidos.</p> <p>6.2.4 Sistemas de Tiempo Real</p>	1, 2, 3, 4, 5,12

10. FUENTES DE INFORMACIÓN

1. Kendall, Kenneth E. (2001).
Análisis y Diseño de Sistemas.
Ed. Prentice-Hall.
2. Laudon & Laudon 8/E (2003).
Management Information Systems.
Ed. Prentice-Hall.
3. Pressman Roger S (2001).
Ingeniería del software.
Ed. McGraw-Hill.
4. Sommerville, Ian (2001).
Ingeniería de software.
Ed. Prentice-Hall.
5. Yourdan, Edward (1999).
Análisis Estructurado Moderno.
Ed. Prentice-Hall.
6. Jacobson, Ivar. (2000).
El Proceso unificado de desarrollo de software.
Ed. Addison Wesley.
7. Fowler, Martin, (1999).
UML Gota a Gota.
Ed. Addison Wesley.
8. Larman, Craig (1999).
UML y patrones.
Pearson.
9. Humphrey, Watts S. (2000).
Introducción al Proceso Software Personal.
Ed. Addison Wesley.
10. Pfleeger, Shari Lawrence (2002).
Ingeniería de Software Teoría y práctica.
Ed. Prentice-Hall.
11. Bruegge Bernd (2001).
Ingeniería de Software Orientada a Objetos.
Ed. Prentice-Hall.

12. Braude, Eric (2003).
Ingeniería de Software Una perspectiva Orientada a Objetos.
Ed. Alfaomega.
13. Meyer, Bertrand (1999).
Construcción de Software Orientada a Objetos.
Ed. Prentice Hall.

11. PRÁCTICAS

Unidad Práctica

- 1 Síntesis sobre los elementos, tipos y clasificación de sistemas.

Características de los tipos de ciclos de vida.

Síntesis: características, mitos, elementos y capas de la Ingeniería de Software.

Normas para el aseguramiento de la calidad durante el proceso de software.

Enfoque estructurado vs. Orientado a Objetos.

Modelos de desarrollo de proceso de software:

- Modelo de Cascada
- Modelo de Espiral
- Modelo Incremental
- Proceso de Desarrollo Unificado
- Proceso Software Personal

Herramientas CASE.

Arquitecturas.

- Arquitectura multiprocesador.
- Arquitectura cliente/servidor.
- Sistemas distribuidos.
- Sistemas de tiempo real.

Documentación del Proyecto (1ª. Parte).