

1 – FUNDAMENTACION DE LA MATERIA

Esta es una materia fundamental de la carrera. Se verán en ella las bases de la Ingeniería de Software, Análisis de Sistemas y Diseño de Sistemas.

La Ingeniería de Software es una disciplina cuya meta es el desarrollo y mantenimiento de sistemas de software. El software es abstracto e intangible. No está restringido por materiales o gobernado por leyes físicas o por procesos de manufactura. De alguna forma esto simplifica la ingeniería de software ya que no existen limitaciones físicas del potencial del software. Sin embargo, esta falta de restricciones naturales significa que el software puede llegar a ser extremadamente complejo y, por lo tanto, muy difícil de entender y “materializar”.

De ahí se puede deducir la importancia de la materia. La dificultad humana de desarrollar y trabajar con elementos abstractos hace que esta materia sea particularmente complicada de dictar y requiere por parte del alumno una concentración y dedicación especial.

2 – OBJETIVOS

Lograr que los alumnos:

- a) Conozcan y comprendan la importancia del análisis de sistemas y del diseño de sistemas como etapas imprescindibles en el ciclo del desarrollo de software.
- b) Que desarrollen un método propio de razonamiento que requiere un alto grado de abstracción.
- c) Dotar a los alumnos de un lenguaje preciso y concreto que forma parte de la disciplina informática.

3 – PLAN DIDACTICO

- a) La regularidad se obtendrá cumpliendo con la asistencia mínima prevista por la Institución y aprobando los parciales que fije la cátedra.
- b) Se tomará un examen parcial escrito en fecha a determinar y habrá un examen recuperatorio. La participación del alumno en clase será continua y estará fomentada desde la cátedra.
- c) Habrá clases teóricas y prácticas. En las primeras se explicarán los diferentes temas que hacen a la materia dando en todos los casos ejemplos reales de situaciones y casos para la mejor comprensión de cada tema. En las clase prácticas se plantearán ejercicios basados en sistemas reales para que el alumno pueda aplicar en cada caso los conceptos vistos.
- d) Habrá dos tipos de trabajos prácticos: individuales y grupales.
- e) Paralelamente a las clases teóricas y a los trabajos prácticos puntuales, se desarrollará un trabajo práctico grupal en el que cada grupo de alumnos deberá construir un sistema completo aplicando todas las etapas. Cada grupo de alumnos tendrá un tema diferente.

4 – PROGRAMA SINTETICO

Unidad 1: Introducción a la Ingeniería de Software
Unidad 2: Ingeniería de sistemas y el ciclo de vida de sistemas
Unidad 3: Procesos de software
Unidad 4: Administración de proyectos
Unidad 5: Requerimientos del software
Unidad 6: Procesos de la ingeniería de requerimientos
Unidad 7: Modelos de sistemas
Unidad 8: Construcción de prototipos de software
Unidad 9: Especificación formal. Modelado de Objetos

3 – PROGRAMA ANALITICO

Unidad 1: Introducción a la Ingeniería de Software.

Objetivos específicos del aprendizaje: El objetivo de esta unidad es introducir el tema de la ingeniería de software. Al final el alumno estará capacitado para:

- a) Comprender qué es la ingeniería de software y por qué es tan importante
- b) Conocer las respuestas clave a las preguntas que proporcionan una introducción a la ingeniería de software
- c) Comprender los aspectos profesionales y de ética que son importantes para los ingenieros de software.

Contenidos:

- a) Qué es el software. Qué es la Ingeniería de Software.
- b) Qué es un proyecto de software. Qué es un modelo de procesos de software.
- c) Costos de la Ingeniería de Software
- d) Herramienta CASE
- e) Atributos del software
- f) Retos fundamentales de la ingeniería de software
- g) Responsabilidad profesional y ética.

Unidad 2: Ingeniería de sistemas basados en computadoras

Objetivos específicos del aprendizaje: El objetivo de esta unidad es introducir el concepto de ingeniería de sistemas basados en computadora. Al final el alumno estará capacitado para:

- a) Saber por qué el software de un sistema se ve afectado por la ingeniería de sistemas
- b) Conocer el concepto de propiedades emergentes de los sistemas, como la fiabilidad, el funcionamiento, la protección y la seguridad.
- c) Conocer por qué el entorno del sistema debe considerarse durante el proceso de diseño del sistema
- d) Conocer la ingeniería de sistemas y los procesos de creación del sistema.

Contenidos:

- a) Propiedades de los sistemas
- b) Los sistemas y su entorno
- c) Modelado de sistemas
- d) Componentes funcionales de los sistemas
- e) El proceso de la Ingeniería de Sistemas
- f) Definición de requerimientos del sistema
- g) Diseño del sistema. Desarrollo de los subsistemas. Integración, instalación, operación, evolución y desmantelamiento.
- h) Adquisición del sistema.

Unidad 3: Procesos del software.

Objetivos específicos del aprendizaje: El objetivo de esta unidad es introducir el concepto de proceso del software como un conjunto coherente de actividades para la producción de software. Al final, el alumno estará capacitado para:

- a) Conocer qué es el proceso de software y el modelo de este proceso
- b) Conocer los diversos modelos del proceso de software y cuándo deben utilizarse
- c) Conocer, someramente, los modelos de proceso de ingeniería de requerimientos de software, desarrollo de software, pruebas y evolución

-
- d) Introducir la tecnología CASE para ayudar al proceso de software

Contenidos:

- a) Modelos del proceso de software
- b) El modelo de “cascada”.
- c) Desarrollo evolutivo
- d) Desarrollo formal del sistema
- e) Desarrollo orientado a la reutilización
- f) Iteración de procesos
- g) Desarrollo incremental, en espiral
- h) Especificación de software
- i) Diseño e implementación
- j) Validación
- k) Ayuda automatizada al proceso
- l) Herramientas CASE. Clasificación.

Unidad 4: Administración de Proyectos

Objetivos específicos del aprendizaje: El objetivo de esta unidad es dar un panorama de la administración de proyectos de software. Al final, el alumno estará capacitado para:

- a) Comprender las diferencias entre la administración de proyectos de software y otros tipos de administración de ingeniería de proyectos
- b) Conocer las tareas principales de los administradores de proyectos de software
- c) Comprender por qué la planificación es esencial en todos los proyectos de software
- d) Conocer la forma en que las representaciones gráficas son utilizadas por los administradores de proyectos para representar la evolución del proyecto en el tiempo
- e) Conocer el proceso de administración de riesgos y algunos de los riesgos que surgen en los proyectos de software.

Contenidos:

- a) Actividades de la administración
- b) Planificación del proyecto
- c) El plan del proyecto. Hitos y productos a entregar.
- d) “Calendarización” del proyecto
- e) Gráficos de barras (Gantt) y redes de actividades (CPM – PERT)
- f) Administración de riesgos.
- g) Identificación, análisis, planificación y supervisión de riesgos

Unidad 5 – Requerimientos del software

Objetivos específicos del aprendizaje: El objetivo de esta unidad es dar una introducción a los requerimientos de sistemas de software y explicar las diferentes formas de expresar estos requerimientos. Al final, el alumno estará capacitado para:

- a) Entender los conceptos relacionados a los requerimientos del usuario y del sistema y por qué estos requerimientos se pueden expresar utilizando diferentes notaciones
- b) Conocer las diferencias entre los requerimientos funcionales y no funcionales
- c) Aplicar dos técnicas para describir los requerimientos del sistema: la descripción de lenguaje natural estructurado y la descripción basada en un lenguaje de programación
- d) Aprender la manera en que los requerimientos se organizan en un documento de requerimientos

Contenidos:

- a) Qué son los requerimientos. Diferencia entre requerimiento y necesidad.
- b) Requerimientos funcionales y no funcionales.
- c) Los requerimientos del dominio
- d) Requerimientos del usuario
- e) Requerimientos del sistema
- f) Especificaciones en lenguaje estructurado
- g) Especificación de requerimientos utilizando un lenguaje de descripción de programas
- h) La especificación de interfaces
- i) El documento de requerimientos del software

Unidad 6 – Procesos de la ingeniería de requerimientos

Objetivos específicos del aprendizaje: El objetivo de esta unidad es describir diferentes procesos genéricos de la ingeniería de requerimientos. Al final, el alumno estará capacitado para:

- a) Conocer las principales actividades de la ingeniería de requerimientos y sus relaciones
- b) Reconocer diferentes técnicas para la obtención y análisis de requerimientos
- c) Conocer la importancia de la validación de requerimientos y cómo se utilizan las revisiones de éstos en este proceso
- d) Entender por qué es necesaria la administración de requerimientos y cómo ayuda a otras actividades de la ingeniería de requerimientos

Contenidos:

- a) Estudios de factibilidad
- b) Obtención y análisis de requerimientos.
- c) Obtención orientada a puntos de vista
- d) Escenarios. Escenarios de eventos. Casos de uso.
- e) Etnografía
- f) Validación, revisión y administración de requerimientos
- g) Requerimientos duraderos y volátiles

- h) Planificación de la administración de requerimientos.
- i) Administración del cambio de los requerimientos.

Unidad 7 – Modelos de Sistemas

Objetivos específicos del aprendizaje: El objetivo de esta unidad es introducir varios tipos de modelos de sistemas que se pueden desarrollar durante el proceso de ingeniería de requerimientos. Al final, el alumno estará capacitado para:

- a) Saber por qué es importante modelar el contexto de un sistema
- b) Conocer los conceptos de modelado de comportamiento, de datos y de objetos
- c) Conocer algunas notaciones definidas de UML y la manera en que estas notaciones se utilizan para desarrollar diferentes tipos de modelos de sistemas
- d) Conocer la manera en que los bancos de trabajo CASE ayudan al modelado de sistemas

Contenidos:

- a) Definiciones de modelos
- b) Modelos de contexto
- c) Modelos de comportamiento
- d) Modelos de flujo de datos
- e) Modelos de máquinas de estado
- f) Modelos de datos
- g) Modelos de objetos
- h) Modelos de herencia
- i) Agregación de objetos
- j) Modelado del comportamiento de objetos
- k) Bancos de trabajo CASE

Unidad 8 – Construcción de prototipos de software

Objetivos específicos del aprendizaje: El objetivo de esta unidad es explicar cómo se utilizan los prototipos de software en el proceso y describir los diferentes enfoques para desarrollar prototipos. Al final, el alumno estará capacitado para:

- a) Conocer el papel de la construcción de prototipos de software en los diferentes tipos de desarrollo de proyectos
- b) Conocer la diferencia entre la construcción de prototipos evolutivos y la de prototipos desechables
- c) Conocer tres técnicas diferentes para el desarrollo de prototipos: desarrollo mediante lenguajes de alto nivel, mediante programación de bases de datos y mediante la aplicación y reutilización de componentes
- d) Saber por qué la construcción de prototipos es la única técnica viable para el diseño y desarrollo de la interfaz del usuario.

Contenidos:

- a) Definición de prototipos. Diferentes tipos.
- b) Construcción de prototipos en el proceso de software
- c) Construcción de prototipos evolutivos
- d) Construcción de prototipos desechables
- e) Técnica de construcción rápida de prototipos
- f) Desarrollo con lenguajes dinámicos de alto nivel
- g) Programación de bases de datos
- h) Ensamblaje de componentes y aplicaciones
- i) Construcción de prototipos de la interfaz del usuario

Unidad 9 – Especificación formal

Objetivos específicos del aprendizaje: El objetivo de esta unidad es introducir las técnicas de especificación formal que se pueden utilizar para agregar detalle a la especificación de requerimientos del sistema. Al final, el alumno estará capacitado para:

- a) Comprender por qué las técnicas de especificación formal ayudan a descubrir problemas en los requerimientos del sistema
- b) Conocer el uso de técnicas algebraicas de especificación formal para definir las especificaciones de la interfaz
- c) Conocer la manera en que las técnicas basadas en modelos formales se utilizan para la especificación del comportamiento

Contenidos:

- a) Qué son los métodos formales
- b) Especificación formal en el proceso de software
- c) Especificación de la interfaz
- d) Especificación del comportamiento

BIBLIOGRAFIA

- **LAUDON y LAUDON.** Administración de los Sistemas de Información. Organización y Tecnología. Editorial Prentice Hall
- **PFLEEGER, Shari Lawrence.** Ingeniería de Software. Teoría y Práctica. Editorial Prentice Hall
- **PRESSMAN, Roger.** Ingeniería de Software. Un enfoque práctico. Editorial McGraw Hill
- **SENN, James.** Análisis y Diseño de Sistemas de Información. Editorial McGraw Hill
- **SOMMERVILLE, Ian.** Ingeniería de Software. Editorial Addison Wesley
- **WHITTEN, BENTLEY y HARLOW.** Análisis y Diseño de Sistemas de Información. Editorial McGraw Hill
- **YOURDON, Edward.** Análisis Estructurado Moderno. Editorial Prentice Hall.
- **CABRERA, Jaime.** Profesión Sistemas. Edición propia