

UNIVERSIDAD SIMÓN BOLÍVAR
Vicerrectorado Académico



1. Departamento: *COMPUTACIÓN Y TECNOLOGÍA DE LA INFORMACIÓN*

2. Asignatura: INGENIERÍA DE SOFTWARE

3. Código de la asignatura: CI-3715

No. de unidades-crédito: 5

No. de horas semanales: Teoría 3 Práctica 1 Laboratorio 3

4. Fecha de entrada en vigencia de este programa: 2011

5.- OBJETIVO GENERAL:

Introducir al estudiante a la problemática y a aquellas técnicas de desarrollo de software de mediana envergadura representativas del enfoque disciplinado asociado con la Ingeniería de Software. Se enfatizarán las técnicas de diseño y validación de sistemas de programas que:

- Incrementen la productividad del desarrollador de software;
- Permitan controlar la complejidad inherente a sistemas de mediana envergadura;
- Introducen al estudiante al desarrollo en equipos (*teams*).

6.- CONTENIDO DETALLADO TEORIA:

Introducción a la ingeniería de software: El desarrollo de software de mediana envergadura, necesidad del trabajo disciplinado en equipo, etapas en el ciclo de vida del software, factores críticos, definición e importancia de la IS. Software orientado a objetos.

Modelado orientado a objetos: Necesidad de múltiples modelos. Introducción a la notación UML. Historia de UML. Diagramas UML.

Planteamiento del problema, requerimientos funcionales y atributos de un software. El modelo de casos de uso: actores, casos de usos, escenarios, descripción de casos de uso, relaciones entre casos de usos, diagrama de casos de uso.

Diagramas de Clases: objetos, clases, atributos, calificadores, métodos, agregación, composición, herencia y asociaciones. Traducción de Clases a Esquema Relacional. Traducción de Clases a Código.

Diagramas de interacción: Diagramas de secuencias de eventos del sistema y diagramas de Comunicación. Contratos. Traducción de Diagramas de Comunicación a Código.

Arquitectura de capas: presentación, aplicación, almacenamiento. Principios de diseño: experticia, acoplamiento y cohesión. El uso de patrones de diseño como observador (model-view-controller, publish-subscribe), iterador, estrategia, apoderado (proxy), decorador, fabrica de abstacta y compuesto.

Pruebas informales (testing): Definición. Barreras psicológicas. Planificación y seguimiento del proceso de pruebas basado en el registro de fallas y defectos. Pruebas unitarias. Pruebas de integración. Pruebas de sistemas.

7.- CONTENIDO DETALLADO PRÁCTICA Y/O LABORATORIO:

Planificación de un proyecto de software. Control de proyectos: Nociones básicas de trabajo en equipo, roles (por ejemplo: miembro, coordinador de proyecto, editor técnico, webmaster, coordinador de base de datos, coordinador de arquitectura), hojas de registro de trabajo, el uso del web site del equipo, planificación, control, seguimiento, coordinación de reuniones, manejo de agenda, distribución explícita de tareas, estimación informal de tiempo, caminos críticos y análisis post-hoc (postmortem).

Desarrollo de un software de mediana envergadura. Prototipado de pantallas. Diseño y Programación Orientados a Objetos.

Documentación: Elaboración de manuales. El manual de instalación. El manual del usuario. El manual del desarrollador.

8.- BIBLIOGRAFIA:

“Object-Oriented Software Engineering: Using UML, Patterns and Java”. Bernd Bruegge, and Allen H. Dutoit. Prentice Hall. 3ra Edición

“Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development”. Craig Larman. Prentice Hall. 3ra Edición.