

CARTA AL ESTUDIANTE

CONOCIMIENTOS PREVIOS:

- Dominio completo de por lo menos un lenguaje de programación de alto nivel.
- Conocimiento de técnicas de especificación de tipos de datos y de encapsulamiento de procedimientos.
- Conocimientos de Bases de Datos y Análisis de Sistemas.

OBJETIVO GENERAL:

- Que el estudiante se familiarize con las técnicas para el diseño, construcción e implementación de productos de "software".

OBJETIVOS ESPECIFICOS:

- Aplicar en un sentido eminentemente práctico los conocimientos adquiridos.
- Dominar técnicas para modularización de sistemas.
- Habituarse a los ambientes de programación en los que se necesiten labores de planificación, control y documentación.
- Desarrollar un sistema de mediana complejidad, mediante la aplicación de herramientas sistemáticas para la construcción de "software".

CONTENIDO:

1. LA PROBLEMÁTICA PROPIA DEL DESARROLLO DE SOFTWARE.
 - Costos y tiempos: las dos variables críticas.
 - Mitos y realidades. Un análisis breve de las posibilidades reales de los proyectos de software y de los factores críticos de éxito.
2. PLANIFICACIÓN DE PRODUCTOS DE SOFTWARE.
 - Herramientas básicas de planificación: Pert, Gantt, etc. Análisis y uso de algún producto típico para esto, por ejemplo MANAGER, HTPM, Mc Project, etc.
 - Planificación a Nivel de Sistema.
 - Planificación a Nivel de Programa.
 - Estudio de un modelo de estimación de costos.
3. TÉCNICAS DE DISEÑO.
 - Conceptos fundamentales de diseño. Aquí se usa fuertemente todo el contenido de programación 2. Incluye técnicas para la derivación de la

estructura jerárquica de un producto de software. Se deben abarcar los conceptos tales como acople, cohesión, rendimiento, flexibilidad, etc.

- Herramientas y metodologías de diseño.
- Uso de planos estructurados.
- Diseño por prototipos.
- Técnicas de desarrollo con orientación a objetos.

4. TECNICAS DE CONSTRUCCION.

- Organización y conducción de equipos de programación.
- Técnicas de documentación de software.
- Construcción en ambientes asistidos por computador.
- Uso de herramientas C.A.S.E.

5. TECNICAS DE IMPLEMENTACION:

- Verificación y validación a nivel de producto.
- Técnicas para demostración de productos.
- Conducción de pruebas de aceptación.
- Técnicas para pruebas y depuración: a nivel de módulo y a nivel de programa (Top-down y bottom-up).

EVALUACION DEL CURSO:

- Exámen parcial 1:	25%
- Exámen parcial 2:	25%
- Proyecto	25%
- Tareas cortas y Quices:	10%
- Participación	15%

NOTAS:

- La presentación del proyecto es requisito obligatorio par aprobar el curso.
- En cada proyectos se deberá adjuntar una acta de calificación, confeccionada por los estudiantes de cada uno de los grupos, en ésta el grupo se autocalificará, con una nota de 0 a 10, de acuerdo al trabajo realizado a lo largo del proyecto.
- En los trabajos individuales (exámenes, tareas o quizes), que se detecte copia por uno o más estudiantes, se calificará con nota de 0 (cero) a todos ellos.

BIBLIOGRAFIA:

1. Fairley, Richard. Ingeniería de Software.
2. Shooman, XX. Software Enginnering.
3. Pressman, Roger. Software Engineering, a practitioner approach.
4. Myers, XX. The art of Software Testing.
5. Boehm Bary. Software Engineering Economics.
6. Senn, James. Análisis y Diseño de Sistemas de Información.