

CI-1417 DISEÑO Y AUDITORIA DE SOFTWARE

Conocimientos previos:

- Dominio completo de lenguajes de programación.
- Conocimiento de técnicas de especificación de tipos de datos y de encapsulamiento de procedimientos.
- Conocimiento de Bases de Datos.
- Conocimientos de Análisis de Sistemas

Objetivos general:

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

Objetivos específicos:

- Domine técnicas para modularización de sistemas.
- Habitúe a ambientes de programación en los que se requiera de actividades de planificación, control y documentación.
- Desarrollar un sistema de mediana complejidad, mediante la aplicación de herramientas sistemáticas para la construcción de "software".

Contenido:

- 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.
- Planificación de productos de software
  - Herramientas básicas de planeación:  
Gant y Pert. Análisis y uso de algún producto típico para esta tarea, por ejemplo MANAGER, HTPM, Mc PROJECT.
  - Planificación a Nivel de Sistema.
  - Planificación a Nivel de Programa.
  - Estudio de un modelo de estimación de costos.
- Técnicas de Diseño.  
Conceptos fundamentales de diseño: Acople, cohesión, rendimiento, flexibilidad.
  - Herramientas y metodologías de diseño.
  - Uso de planos estructurados.
  - Diseño por prototipos.
  - Técnicas de desarrollo con orientación a objetos.

- Controles de sistemas
  - Controles lógicos
  - Controles físicos
  - Pistas de Auditoría
- Diseño de entradas y salidas del sistema
- Implantación de Sistemas
  - Instrucción y capacitación
  - Diseño de pruebas
  - Conversión
  - Puesta en marcha

**Evaluación:**

- Exámenes Parciales	35%
- Proyecto	25%
- Quices	15%
- Participación	10%
- Tareas Investigación	15%

**Bibliografía**

1. Fairley, Richard. Ingeniería de Software.
2. Shooman, XX. Software Engineering.
3. Pressman, Roger. Software Engineering, a practitioner approach.
4. Myers, XX. The art of Software Testing.
5. Boehm Bary. Software Engineering Economics.
6. Ramírez, Rafael. Diseño estructurado para el desarrollo de Software. Tesis de grado. Universidad de C.R.
7. Burch John y Strater Felix. Sistemas de Información. Teoría y Práctica. Editorial Limusa.
8. De Marco Tom. Structured analysis and system specification. Prentice-Hall.
9. Eliason Alan. System Development. Analysis, Design and Implementation. Scott, Foresman and Company, 1990.

**NOTAS:**

- La carta de aceptación del proyecto por parte del usuario es requisito para la aprobación del curso.