



## INGENIERÍA DE SOFTWARE 1

### 1. IDENTIFICACIÓN

Ubicación: Variable según mención

Prelación: Algoritmos y Complejidad 3 (Ing.) o PF3 (TSU)

T P L C: 3 1 0 3

Condición: Obligatoria

Departamento: Computación

Área Curricular de Formación: IS, SI, TIS, TSI.

Nivel de Formación: Formativo.

### 2. JUSTIFICACIÓN

Hoy en día el *software* es un producto (un sistema) y, al mismo tiempo, es el medio que permite el desarrollo de tal producto (una herramienta). Los sistemas programados son el elemento computacional clave que permite aprovechar verdaderamente el potencial de los avances tecnológicos obtenidos en el área de equipo físico (o *hardware*). De la calidad, eficiencia y confiabilidad de los sistemas programados depende el apoyo real a la toma de decisiones que un equipo computacional brindará a sus usuarios.

La ingeniería de la programación es el área de la informática que ofrece métodos, técnicas y herramientas para desarrollar y mantener sistemas programados de alta calidad manteniéndose dentro de tiempos, costos y uso de recursos previamente establecidos.

### 3. OBJETIVOS

- Introducir al estudiante en los conceptos fundamentales y tendencias de la Ingeniería de Software, en los métodos, técnicas y herramientas comúnmente utilizadas en el desarrollo de sistemas programados.
- Adiestrar al estudiante en el uso de métodos y técnicas orientadas por objetos (OO) para modelar y construir un sistema programado.
- Realizar un proyecto de desarrollo de *software* aplicando métodos, técnicas y herramientas OO.



#### 4. CONTENIDO PROGRAMÁTICO

##### UNIDAD 1. LEVANTAMIENTO DE REQUISITOS

Tema 1. Captura de requerimientos Funcionales y no funcionales. Interfaz H/M.

Tema 2. Especificación de Requerimientos. Métodos y técnicas orientadas por Objetos.

##### UNIDAD 2. ANÁLISIS DE REQUISITOS

##### UNIDAD 3. REQUISITOS FUNCIONALES Y NO FUNCIONALES

##### UNIDAD 4. ESPECIFICACIÓN Y DOCUMENTACIÓN DE REQUERIMIENTOS

- UNIDAD 5. HERRAMIENTAS Y LENGUAJES DE MODELADO DE REQUISITOS

(EJ. HISTORIAS DE USUARIO, UML: CASOS DE USO, DIAGRAMAS DE ACTIVIDADES)

##### UNIDAD 6. VERIFICACIÓN Y VALIDACIÓN DE REQUISITOS

Tema 3. Verificación y validación de requerimientos

#### 5. METODOLOGÍA DE ENSEÑANZA

El proceso de enseñanza de la materia se realiza a través de cuatro estrategias complementarias de enseñanza:

- La realización de clases magistrales.
- Las discusiones en grupo sobre tópicos específicos relacionados con el material visto en clase o luego de una búsqueda documental (bibliográfica o por Internet).
- La realización de ejercicios prácticos
- La realización de un proyecto de desarrollo de software.

#### 6. RECURSOS

Transparencias, proyector multimedia, apuntes de clase, libro y revistas de referencia, discusiones, lenguajes, herramientas y paquetes de desarrollo de software.

#### 7. EVALUACIÓN



- Evaluaciones escritas por tema culminado que incluyen teoría y práctica, representando un 45% de la nota final.
- Informes de proyecto (avance + final) con un porcentaje de 45% nota final.
- Intervenciones en clase y participaciones en las discusiones con un porcentaje del 5% de la nota final.
- Interrogatorios orales para un porcentaje del 5% de la nota final.

## **8. BIBLIOGRAFÍA**

Pressman, R. Ingeniería de Software. Un enfoque práctico. McGraw Hill. séptima edición. 2008.

Muller, P. Modelado de objetos con UML. Eyrolles-Gestión 2000, Barcelona, España. 1997.

Pfleeger, S. Software Engineering. Theory and Practice. Prentice Hall. 4ta edición, 2007.

Sommerville, I. Ingeniería de Software. Addison Wesley. 1999.

Revistas: Information & Management, IEEE Software, IEEE transactions on Software Engineering.