



A. IDENTIFICACIÓN

CARRERA:	INGENIERÍA INFORMÁTICA
ASIGNATURA:	TALLER DE DESARROLLO DE SW
SIGLA:	INF 3741
DURACIÓN:	Un semestre académico (20 semanas)
HORAS SEMANALES:	Teóricas: 4, Prácticas: 1, Laboratorio 1 TOTAL: 6
PLAN DE ESTUDIOS:	2011

B. CONTRIBUCIÓN AL PERFIL

Objetivos:

Analizar las causas de la necesidad de una metodología para el proceso de desarrollo de software y establecer las definiciones para la selección de dicha metodología de desarrollo y administración de proyectos.

Desarrollar y gestionar los requerimientos de un proyecto de SW desde la licitación, la construcción de modelos para estos requerimientos, el análisis de los mismos hasta la elaboración de un modelo de pruebas funcionales.

Utilizar herramientas CASE que permitan el análisis y diseño a través del Lenguaje Unificado de Modelado como herramienta de desarrollo.

Utilizar un SGBD en el proceso de implementación del SW.

Establecer la trascendencia fundamental de las pruebas en el proceso de desarrollo mostrando herramientas y métodos para llevarlas a cabo

Unidades de competencia:

- Desarrollar aplicaciones empleando lenguajes de programación.
- Implantar soluciones algorítmicas a problemas complejos.
- Desarrollar sistemas de información en sus diferentes categorías.
- Incorporar modelos formales en el desarrollo de sistemas de información.

C. CONTENIDO PROGRAMÁTICO

Contenido mínimo:

Enfoque de la metodología de desarrollo de SW. - Requerimientos.- Modelo de requerimientos.- Modelo del dominio.- Análisis.- Diseño de arquitecturas.- Diseño detallado.- Trazabilidad.- Pruebas

Contenido analítico:

Tema 1: Enfoque de la metodología de desarrollo de SW.

- 1.1 Estudio crítico del Desarrollo de Software.
- 1.2 Los miembros del grupo de desarrollo y sus roles.
- 1.3 UML.
- 1.4 Metodología de desarrollo.



- 1.5 Análisis crítico y comparativo del RUP con metodologías ágiles (XP, Cristal,...).
- 1.6 El factor humano.
- 1.7 Administración y gestión de proyectos.

Tema 2: Requerimientos.

- 2.1 Presentación del sistema a desarrollar a partir de su especificación textual.
- 2.2 Toma de requerimientos.
- 2.3 Clasificación de requerimientos:
 - 2.3.1 Funcionales.
 - 2.3.2 No funcionales.
 - 2.3.3 Otros.
- 2.4 Documento de Especificación de Requerimientos (ERS).
- 2.5 Determinación del alcance.
- 2.6 Funciones incluidas.
- 2.7 Funciones no incluidas.
- 2.8 Administración de cambios en los requerimientos.

Tema 3: Modelo de requerimientos.

- 3.1 Casos de Uso.
- 3.2 Construcción.
- 3.3 Actores.
- 3.4 Especificación de escenarios.
- 3.5 Modelo de interfaces.
- 3.6 Patrones.
- 3.7 Relaciones entre Casos de Uso.
- 3.8 Entidades del Modelo de Dominio.

Tema 4: Modelo del dominio.

- 4.1 Conceptos centrales.
- 4.2 Paquetes lógicos.
- 4.3 Relación entre paquetes.

Tema 5: Análisis.

- 5.1 Descomposición en entidades.
- 5.2 Revisión de relaciones entre paquetes lógicos.
- 5.3 Definición de interfaces de paquetes lógicos.

Tema 6: Diseño de arquitecturas.

- 6.1 Definición de la Arquitectura.
- 6.2 Paquetes físicos.
- 6.3 Definición de criterios de diseño globales.
- 6.4 Definición del ambiente de desarrollo.
- 6.5 Modelo de deployment.



Tema 7: Diseño detallado.

- 7.1 Diagramas estáticos (clases).
- 7.2 Diagramas dinámicos (colaboración, secuencia).
- 7.3 Modelo de persistencia (diagrama de entidad relación).

Tema 8: Trazabilidad.

- 8.1 Relaciones entre requerimientos y casos de uso.
- 8.2 Relaciones entre Casos de uso y clases de análisis.
- 8.3 Relaciones entre Clases de análisis y clases de diseño.

Tema 9: Pruebas.

- 9.1 Estándares de programación.
- 9.2 Testeo continuo.
- 9.3 Pruebas de sistema, plan, diseño y ejecución.
- 9.4 Casos de prueba y procedimientos.

D. REFERENCIAS BIBLIOGRÁFICAS

- [1] Survival G., McConnell S. (1998). **Software Project**. Microsoft Press
- [2] Grady Booch. (1994). **Object-Oriented Analysis and Design with Applications** (2nd Edition)
- [3] Larman G. (2004). **UML y patrones**.. Editorial Prentice Hall, México
- [4] Jacobson I., Boch G., Rumbaugh (2002). **El proceso Unificado de Desarrollo de Software**. Editorial Adisson Wesley.
- [5] Cecil Martin R. (2002). **Agile Software Development. Principles, patterns and practices**.
- [6] Beck K. (2000). **Extreme Programming Explained**. Addison-Wesley