



INGENIERÍA DE SOFTWARE I

1. OBJETIVOS

La asignatura tiene como propósito brindar al alumno los conocimientos teóricos, prácticos y de laboratorio para que sea capaz de concebir el diseño del software como una ingeniería mediante la descripción de los problemas, principios, métodos y tecnologías asociadas, enfocándolo como la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Plantea un Proyecto Final integrador, su presentación y defensa.

Presenta los siguientes resultados de aprendizaje:

- Identificar los modelos de ciclo de vida y las etapas del desarrollo de un Producto Software.
- Reconocer terminología de la especialidad y la relación del desarrollo de software con las otras etapas del proceso software.
- Comprender la importancia del Análisis y el Diseño dentro del proceso de desarrollo de software, y del Paradigma de la Orientación a Objetos.
- Comprender técnicas, métodos y herramientas orientadas a analizar, especificar, diseñar, implementar y probar Sistemas Software.
- Desempeñar un rol activo como miembro de un equipo de trabajo.
- Desempeñar una comunicación eficaz de manera oral y escrita.
- Implementar y Resolver casos en los cuales se involucran técnicas, métodos y herramientas de la Ingeniería de Software en el desarrollo de un proyecto.
- Proponer contextos para desarrollar mecanismos innovadores.
- Proponer criterios de selección y evaluación de proyectos de ingeniería.

2. CONTENIDOS

1) LOS SISTEMAS DE SOFTWARE Y SU NATURALEZA

Ingeniería del Software: Diferencia entre Ingeniería y Artesanía - La Ingeniería de Sistemas - Sistemas de Negocio - Sistemas Industriales. Modelos de Ciclos de Vida: Definiciones – Estados Software – Proceso Software. Modelo Clásico: Modelo en Cascada en V - Prototipado Evolutivo. Sistemas: Definición de Sistema - Sistemas Naturales y Sistemas Artificiales - Componentes de un sistema - Sistemas de Computación - Sistemas de Información. Componentes Principales. La complejidad: Estrategias para atacar la complejidad - Principios Generales – Ejemplos.

2) EL PARADIGMA DE OBJETOS

El Modelo de Objetos: Evolución, Fundamentos y Elementos del Modelo de Objetos – Aplicación del modelo de objetos. Clases y Objetos: La naturaleza de los Objetos – Relaciones entre objetos – La naturaleza de una Clase – Relaciones entre clases – La Interacción entre clases y objetos. Clasificación: La importancia de una clasificación correcta – Identificando clases y objetos – Enfoques Clásicos – Análisis del Comportamiento – Análisis de Dominios – Análisis de Casos de Uso – Otros enfoques. La Notación UML 2: Elementos de la Notación – Diagramas de Clases - Diagramas de Objetos - Ejemplos.



3) METODOLOGÍA SCRIPT Y REQUERIMIENTOS

Metodología Script: Filosofía Script - Fases de la metodología - Diagrama Script - Procesos Script - Documentos de Entrada y Salida de las Fases. Principios y modelos de requerimientos: Técnicas de Educación de Requerimientos - Entrevista abierta, guiada y estructurada - Comunicación con el usuario. Modelo IEEE/ANSI 830-1993. Proceso de Especificación: Objetivos del Proceso de Especificación - Formas y Contenidos de las documentaciones del Cliente y los Usuarios. Modelado de Casos de Uso para el Negocio. Diagramas de Casos de Uso - Relaciones - Descripción Textual de Casos de Uso. Diagramas de Actividad. Modelado: Modelos de Clases - Diagramas de Secuencia - Diagramas de transición de Estados de Clases.

4) MODELADO CONCEPTUAL

La Especificación C: El proceso Script propiamente dicho - Fases de la Especificación C - Definición de Interfaces de Usuario - Guiones y Escenarios - Tablas de Transición de Escenarios - Modelado y Documentación UML 2.

5) DISEÑO, CODIFICACIÓN Y PRUEBAS

La Especificación D: Introducción a la Arquitectura de Software - Diseño Estructural - Diseño Detallado - Técnicas de Diseño Script - Fichas Técnicas de Servicios de Objeto. Diseño de Datos: Modelo Lógico de Datos - Modelo Físico de Datos - Documentación. Codificación: Elección del Lenguaje - Documentación del Código - Reglas Generales. Pruebas: Test de Caja Negra - Test de Caja Blanca - Test de Integración - Test de Aceptación. Calidad. Auditoria y peritaje. Administración de proyectos de Software.

6) TALLERES COMPLEMENTARIOS

Talleres complementarios obligatorios: “Taller de Inserción Laboral y Emprendedorismo” – “Taller de Organización y Gestión de Empresas de Proyectos de Software: Aspectos legales y de Presupuestación” – “Taller de Análisis e Implementación de Modelos Contables en OO” – “Taller de Implementación de la interfaz IGestor”

7) DESARROLLO DE UN PROBLEMA COMPLETO CON CLIENTES REALES.

A desarrollarse en etapas durante el cursado de la asignatura, siendo los Trabajos Prácticos la presentación de los avances del mismo. Terminado el cursado, el alumno deberá presentar el trabajo de forma escrita y tendrá una presentación oral ante sus pares y su correspondiente evaluación.

3. **BIBLIOGRAFÍA**

- G. Booch G. Análisis y Diseño Orientado a Objetos con Aplicaciones, 2º edición. Addison-Wesley/Díaz de Santos. 1996.
- R. Pressman. Ingeniería del Software, un Enfoque Práctico, 3º edición. Mc Graw-Hill. 2005.
- J. Rumbaugh. Modelado y Diseño Orientado a Objetos. Prentice Hall. 1997.
- J. Rumbaugh, I. Jacobson & G. Booch. El Lenguaje Unificado de Modelado. Manual de Referencia. Addison-Wesley. 2000.
- C. Larman. UML y Patrones, 2º edición. Prentice-Hall. 2003.



- M. Odstrcil. Conceptos de Sistemas y Análisis Orientado a Objetos, Asoc. Cooperadora FACET. 2001.
- E. Yourdon. Análisis Estructurado Moderno, 1º edición. Prentice Hall. 1993.
- S. Pfleeger. Ingeniería de Software, Teoría y Práctica, 1º edición. Prentice Hall. 2002.
- A. Gómez. N. Juristo, C. Montes & J. Pazos. Ingeniería del Conocimiento, 1º edición. Editorial Centro de Estudios Ramón Areces, S.A. 1997.

4. METODOLOGÍA DE ENSEÑANZA

La asignatura se desarrollará en la modalidad teórica en aula y práctica en laboratorio, aplicando las metodologías activas que fomenten la discusión crítica y el planeamiento de criterios personales respecto a los temas tratados. Las experiencias de aprendizaje se desarrollarán orientadas por los siguientes métodos activos:

- Material de lectura dirigida.
- Trabajos en Laboratorios (10 horas).
- Demostraciones.
- Proyectos en Grupos (8 horas).
- Presentación oral grupal.
- Presentación escrita grupal.
- Seminario – Taller (8 horas).
- Clases guiadas por el alumno con exposición oral.

5. EVALUACIÓN

Los métodos de evaluación a usar en la asignatura son:

- Exámenes Parciales.
- Examen Final.
- Evaluación de Trabajos Prácticos y Laboratorios.
- Evaluación de la Defensa del Trabajo Grupal.
- Asistencia a Laboratorios.

Con las siguientes condiciones cumplidas:

- Nota promedio de exámenes parciales igual o superior a 50/100 (EP).
- 100% de Trabajos Prácticos y Laboratorios presentados.
- Promedio de Trabajos prácticos y Laboratorios igual o superior a 50/100 (TP).
- Nota del trabajo grupal igual o superior a 50/100.
- Nota de examen final igual a superior a 50/100 (EF).

La nota final (NF) se calcula como se muestra a continuación:

$$NF = 0,7 * EF + 0,2 * TP + 0,1 * EP$$

6. CARGA HORARIA

Perteneciente al Bloque de Tecnologías Aplicadas, módulo VII del cuarto año de la carrera de Ingeniería en Computación, la asignatura tiene una carga horaria de 96 horas, a saber:



- Clases teóricas: 44 horas.
- Clases prácticas: 30 horas.
- Proyectuales: 8 horas.
- Laboratorios: 10 horas.
- Exámenes: 4 horas.

7. **OTRA INFORMACIÓN**

Horarios: lunes, miércoles y viernes de 10 a 12 horas. Clases Teóricas y Prácticas en aula 1-3-02. Laboratorios, Talleres y Exposiciones en Gabinete de Informática I y Laboratorio de Software.

Régimen de Promoción:

Con las siguientes condiciones cumplidas:

- Nota promedio de exámenes parciales igual o superior a 60/100 (EP)
- 100% de Trabajos Prácticos y Laboratorios presentados.
- Promedio de Trabajos prácticos y Laboratorios igual o superior a 50/100. (TP)
- Nota del trabajo grupal igual o superior a 60/100.
- Nota de examen final igual a superior a 70/100 (EF)

La nota final (NF) se calcula como se muestra a continuación: $NF = 0,7 * EF + 0,2 * TP + 0,1 * EP$

Sitio de Cátedra:

<https://catedras.facet.unt.edu.ar/ingsoftware>

<https://www.facebook.com/groups/489942524441675>

Aula:

Notebook de la Cátedra.

Proyector y Pizarra.

Software:

Software libre: NetBeans 8.0, MySQL 5.0, OpenOffice.

Software versión de evaluación: Enterprise Architect 6.5.

Laboratorios: Laboratorio de Software.

Recursos Web:

UML: <https://yuml.me>

OpenProject: <https://www.openproject.org>

SEI: <http://www.sei.cmu.edu>

IFPUG: <http://www.ifpug.org/?lang=es>