

## PROGRAMA DE ASIGNATURA

<b>ASIGNATURA:</b> Ingeniería del Software I	<b>AÑO:</b> 2012
<b>CARÁCTER:</b> Obligatoria	
<b>CARRERA:</b> Licenciatura en Ciencias de la Computación	
<b>RÉGIMEN:</b> cuatrimestral	<b>CARGA HORARIA:</b> 120 hs.
<b>UBICACIÓN en la CARRERA:</b> Tercer año – Segundo cuatrimestre	

### FUNDAMENTACIÓN Y OBJETIVOS

Lograr que el alumno sea capaz de:

- entender actividades de análisis, diseño y testing del software.
- Manejar elementos de planificación, especificación y documentación de proyectos usando el paradigma de OO en análisis y diseño de sistemas.

### CONTENIDO

#### 1 - Introducción:

- El dominio del problema. Teoría general de sistemas. Sistemas de Información. Privacidad, integridad y seguridad en sistemas de información.
- El desafío de la Ingeniería del Software.
- El enfoque de la Ingeniería del Software.

#### 2 - Ciclo de vida del software:

- Procesos. Modelo de procesos. Componentes. Enfoque ETVX.
- Características deseadas del proceso del software: Predecible y repetible, Tolerante a cambios, Testeable y Mantenible.
- Proceso de desarrollo del software, etapas fundamentales.
- Reingeniería
- Modelos de procesos de desarrollo: Cascada, Prototipado, Iterativo.
- Otros procesos del software: Administración del proyecto, Proceso de inspección, Administración de configuración, Administración de cambios, Administración del proceso (CMM). Sistemas

colaborativos.

- Ingeniería de software de sistemas de tiempo real
- Nociones de auditoría y peritaje.

### 3 - Ingeniería de requerimientos:

- Requerimientos del software, Necesidad de la especificación de requerimientos, Proceso de requerimientos.
- Análisis del problema: Enfoque informal, Modelo de flujo de datos (DFD), Modelo orientado a objetos (UML), Prototipado.
- Especificación de los requerimientos del software: Características, Componentes, Lenguajes de especificación, Estructura de un documento.
- Especificación funcional con Casos de Uso: Conceptos, Estructura, Abstracción.
- Validación.
- Métricas: Tamaño, Calidad.

### 4 - Arquitectura del software:

- Rol de la arquitectura del software.
- Vistas: Módulos, Componentes y conectores, Asignación de recursos.
- Vista de Componentes y Conectores (C&C). Estilos arquitectónicos para C&C: Tubos y Filtros, Datos compartidos, Cliente-servidor, Estratificados.
- Conceptos de Arquitecturas Orientadas a Servicios
- Documentación del diseño arquitectónico.
- Arquitectura en comparación con el diseño. Preservación de la integridad de una arquitectura.
- Evaluación de las arquitecturas (método de análisis ATAM).

### 5 - Planeamiento del proyecto de software:

- Planeamiento del proceso.
- Estimación del esfuerzo: Incertidumbres, Construcción de los modelos (estimaciones top-down y bottom-up), El modelo COCOMO.
- Planificación y recursos humanos: Planificación global y detallada, Estructura del equipo de trabajo.
- Plan del Control de Calidad: Introducción y eliminación de errores, Enfoques, Plan.
- Administración del Riesgo: Conceptos, Evaluación, Control.
- Planeamiento del seguimiento del proyecto: Mediciones,

---

Seguimiento observacional, Registro del seguimiento.

6 - Diseño orientado a funciones:

- Niveles en el proceso de diseño.
- Principios del diseño: Particionado y jerarquía, Abstracción, Modularidad. Estrategias top-down y bottom-up.
- Acoplamiento y Cohesión.
- Notación y especificación del diseño.
- Metodología de diseño estructurado: Cuatro pasos elementales, Heurísticas de diseño, Análisis de transacción.
- Verificación.
- Métricas: de red, de estabilidad y de flujo de información.
- Patrones

7 - Diseño orientado a objetos:

- Conceptos de la orientación a objetos: Clases, Objetos, Relación entre objetos, Herencia, Polimorfismo.
- Conceptos de diseño: Acoplamiento, Cohesión, Principio abierto-cerrado.
- UML.
- Una metodología de diseño: Modelado dinámico, Modelado funcional, Definición de clases y operaciones, Optimización.
- Métricas.
- Diseño centrado en el usuario
- Patrones

8 - Diseño detallado:

- Lenguaje de diseño de procesos (PDL). Diseño lógico (del algoritmo). Modelo de estado de clases (Autómatas de estado finito). Refinamiento en abstracciones de datos e invariantes de representación.
- Verificación: "Recorrido" del diseño, Revisión crítica (bajo proceso de inspección), Verificación de consistencia y Uso de técnicas formales.
- Métricas: Complejidad Ciclomática, Vínculos de datos, Métricas de cohesión.

9 – Codificación:

- Principios y pautas para la programación: Errores comunes,

Programación estructurada, Ocultamiento de la información, Prácticas de programación, Estándares de codificación.

- Proceso de Codificación: Incremental, Dirigido por test, Programación de a pares, Control del código fuente y construcción (build).
- Refactorización: Conceptos básicos, “Malos olores”, Refactorizaciones comunes.
- Verificación: Inspección del código, Test de unidad, Análisis Estático, Métodos formales.
- Métricas: Tamaño y Complejidad.

#### 10 – Testing:

- Conceptos fundamentales: Defecto y desperfecto (fault & failure), Oráculos, Casos de test y criterios de selección, Psicología del test.
- Testing de caja negra: Particionado por clases de equivalencia, Análisis de valores límites, Grafo de causa-efecto, Testing de a pares, Casos especiales, Testing basado en estados (Máquinas de estado finitas).
- Testing de caja blanca: Criterios basados en flujo de control, Criterios basados en flujo de datos, Testing por mutación, Generación de casos de tests y herramientas de soporte.
- El proceso de testing: Niveles, Plan, Especificación de los casos de test, ejecución de los casos de test, Análisis, Registro de defectos y seguimiento.
- Análisis y Prevención de los defectos.
- Métricas. Estimación de la confiabilidad

#### 11 – Aspectos profesionales y sociales

- Computación y sociedad.
- Propiedad intelectual, licencia de software y contratos informáticos.
- Aspectos legales.
- Responsabilidad y ética profesional.

### **BIBLIOGRAFÍA**

#### BIBLIOGRAFÍA BÁSICA

- Pankaj Jalote. An Integrated Approach to Software Engineering, Third Edition. Springer. 2005. ISBN: 0-387-20881-X.

#### BIBLIOGRAFÍA COMPLEMENTARIA

- F. Brooks. The Mythical Man Month. Addison-Wesley. 1995.
- R. Glass. The Relationship Between Theory and Practice in Software Engineering. Communications of the ACM, 39(11):11–13. Nov. 1996.
- R. Glass. Managing for Innovation. Communications of the ACM, 51(3):17–18. Mar. 2008.
- IEEE. Diversos estándares de la IEEE sobre la Ingeniería del Software.

[http://standards.ieee.org/reading/ieee/std\\_public/description/se/](http://standards.ieee.org/reading/ieee/std_public/description/se/).

- B. Meyer. Object-Oriented Software Construction (2nd Edition). Prentice Hall. 2000.
- D. Parnas. Software Engineering: An Unconsummated Marriage. Communications of the ACM, 40(9):128. Sep. 1997.
- Sun Microsystems. Java Code Convention. 1997
- R. Stallman et al. GNU coding standards. 2007.

## **METODOLOGÍA DE TRABAJO**

La materia se organiza en clases teóricas, clases prácticas y actividades de laboratorio. En las clases teóricas se brindan los contenidos fundamentales de la asignatura.

En las clases prácticas se ejercita sobre los temas cubiertos en la teoría, con especial énfasis en actividades de análisis y diseño orientado a objetos, especificación de requisitos y testing.

Las clases de laboratorios se utilizan para llevar adelante un proyecto de desarrollo, de tamaño mediano, que es resuelto en grupos y en el cual los alumnos experimentan los problemas que surgen en el desarrollo de un sistema real, siguiendo todas las etapas que involucra el desarrollo de un proyecto real.

Las clases teóricas son complementadas con charlas de temáticas variadas vinculadas a la ingeniería de software, brindadas por docentes de la asignatura e invitados de la industria local.

## **EVALUACIÓN**

### Condiciones para la regularidad

- Aprobación del proyecto, pasando satisfactoriamente cuatro instancias de seguimiento/evaluación del mismo.

#### Condiciones para la aprobación

- Aprobación de un examen final escrito.
- Defensa del proyecto.

#### Promoción

- Aprobación del proyecto
- Aprobación de dos evaluaciones parciales con nota no inferior a 7 (siete).