

**Asignatura:** Ingeniería de Software I

**Carrera/s:** Ingeniería en informática

**Ciclo lectivo:** 2016

**Docente/s:** Ing. Oscar Leonardo Cortes Bracho; Ing. Bernardo Pagola

**Carga horaria semanal:** 5 Horas Semanasles.

**Tipo de asignatura:** teórico-práctica.

### **Fundamentación**

Ingeniería de Software I, es una materia obligatoria correspondiente al tercer año de la carrera Ingeniería en Informática. En la materia los estudiantes abordarán conocimientos generales acerca de la Ingeniería de Software. Se aplicaran los conceptos obtenidos en las materias de "Fundamentos de informática" , "Algoritmos y programación" , y "Complejidad temporal" enfocados desde la producción de software. Se sentaran las bases teórico prácticas a desarrollar en la materia "Proyecto de Software.

### **Objetivos:**

Se pretende que al finalizar la cursada el estudiante:

- tendrá una visión integrada de las disciplinas que componen la ingeniería de software, a partir de su formación en programación y diseño de de aplicaciones previamente adquiridos.
- conocerá y utilizará técnicas avanzadas de la Ingeniería de Requerimientos.
- tendrá la capacidad y criterios de análisis de sistemas de software.

### **Contenidos mínimos:**

El proceso de software. Ciclos de vida de software. Herramientas para el proceso de software. Testing. Ingeniería de requerimientos. Introducción a la Administración de Proyectos.

### **Contenidos temáticos o unidades:**

#### **Unidad I**

Introducción a la Ingeniería de Software. Definiciones. Ingeniería de Software y Calidad.

#### **Unidad II**

Procesos. Modelos de Procesos. Cascada. Modelos Incrementales. Modelos Evolutivos. Prototipación. Proceso Unificado. Metodologías Ágiles. Herramientas y Técnicas.

#### **Unidad III**

Ingeniería de Requerimientos: Administración de requerimientos. Técnicas de relevamiento. Herramientas para el relevamiento de requerimientos. Planificación de administración de requerimientos. Importancia de entender los procesos de negocio. Administración de cambios de requerimientos. Matriz de trazabilidad

#### **Unidad IV**

Análisis de requerimientos: Herramientas para el análisis de requerimientos. Analista de requerimientos como primer paso de la arquitectura de una aplicación.

#### **Unidad V**

Documentación del diseño de aplicaciones: Documentación en su medida justa. Actualización de la documentación.

#### **Unidad VI**

Testing de aplicaciones: Importancia de testing de aplicaciones. Planificación del testing. Armado de casos de test. Ejecución de casos de test. Administración de bugs.

## **Unidad VII**

Introducción a la administración de proyectos. Planificación de proyectos. Seguimiento de proyectos.

### **Bibliografía:**

- Sommerville, Ian. Ingeniería de Software. Editorial Pearson. Editorial Addison Wesley (ISBN: 84-7829-074-5). Ed. 7°. Año 2005.
- Pressman, Roger. Ingeniería De Software: Un Enfoque Práctico. Editorial McGraw Hill (ISBN: 9789701054734). Ed. 6°. Año 2006.
- Standish Group. <http://www.standishgroup.com/>

### **Modalidad de dictado:**

Las clases se desarrollarán en el Laboratorio de Informática. Se organizarán en modalidades teórico-prácticas con soporte de presentaciones digitales y prácticas en función de cada clase.

En las clases se presentan los contenidos teóricos y se van exponiendo situaciones reales en forma de ejemplos que ayuden a comprender los nuevos conceptos introducidos.

La formación práctica está basada en la resolución de problemas tipo y de problemas abiertos de ingeniería, cuyas resoluciones se realizan principalmente en las computadoras, utilizando aplicaciones de uso en la industria que permitan un contacto directo con las tecnologías actuales.

En cuanto a los problemas abiertos de ingeniería, se realizarán trabajos relacionados con cada una de las etapas del desarrollo del software: especificación, diseño, desarrollo y mantenimiento, incorporando como parte del aprendizaje en cada etapa las herramientas asistidas por computadoras que soportan el desarrollo

de las aplicaciones. El desarrollo de estos trabajos se realizará utilizando metodologías ágiles, lo que implica que las entregas son iterativas y de forma incremental, permitiendo al docente y a los estudiantes mejorar constantemente sobre el desarrollo del trabajo en función de los comentarios de los docentes y la evolución propia de los estudiantes. La realización de los trabajos implica la identificación de un problema dado y la solución del mismo, a partir de la aplicación de los conocimientos adquiridos hasta entonces, lo cual constituye la base formativa para que el estudiante adquiera las habilidades que le permitan encarar proyectos y diseños de ingeniería de software.

El material correspondiente a las clases teóricas, así como los documentos de la práctica se encontrarán disponibles a través de un grupo Web al cual los estudiantes tendrán acceso. Este mecanismo también será utilizado para realizar consultas simples.

### **Régimen de aprobación:**

La asignatura puede aprobarse por promoción directa en caso que la nota final sea superior o igual a siete (7) o mas puntos promedios en todas las instancias evaluativas, sean estas parciales o recuperatorios, debiendo tener una nota igual o mayor a 6 en cada una de ellas.

Los detalles de la evaluación son:

- Dos Exámenes Parciales
  - Dos Recuperatorios
  - Trabajos Prácticos(Proyecto)
  - Asistencia 75%
- Al finalizar la cursada cada estudiante tendrá una calificación de la parte teórica (NT) obtenida del promedio de los dos parciales aprobados, una calificación de la parte práctica (NP) obtenida del promedio de los diez 10 Trabajos entregados, y una calificación del Trabajo Final (NTF) Correspondiente a la integración de los 10 Trabajos prácticos desarrollados durante la materia en un informe final.La nota final de la asignatura se obtiene de la siguiente manera:
  - $NOTA\ FINAL = 0.6 * NT + 0.3 * NP + 0.1 * NTF$