



Programa de INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE

1. NOMBRE DE LA UNIDAD CURRICULAR

Introducción a la Ingeniería de Software

2. CRÉDITOS

10 créditos

3. OBJETIVOS DE LA UNIDAD CURRICULAR

La ingeniería de software es la disciplina relacionada a la aplicación de teoría, conocimiento y práctica para construir de forma efectiva y eficiente sistemas de software confiables que satisfagan los requisitos de los clientes y usuarios. Aplica a sistemas pequeños, medianos y grandes y acompaña todas las fases del ciclo de vida del software, desde su concepción hasta su operación por parte de los usuarios.

El objetivo de esta unidad curricular es brindar un panorama general de la disciplina a los estudiantes profundizando en los aspectos más relevantes. Busca introducir al estudiante en el uso de algunas técnicas y herramientas particulares para que pueda participar en distintos roles en un proyecto de ingeniería de software.

4. METODOLOGÍA DE ENSEÑANZA

En dos clases semanales de 2 horas los docentes presentan los principales aspectos teóricos a los estudiantes. En cada clase se incluyen segmentos de presentaciones de los docentes y trabajos en grupos de los estudiantes para resolver problemas o ejercicios.

Dada la dinámica del trabajo en clase se requiere que el estudiante realice un importante esfuerzo no presencial para: estudiar en profundidad los temas o prácticas introducidos por los docentes.

Además se realizan actividades adicionales que permiten fomentar el involucramiento del estudiante. En particular, se lleva a cabo un trabajo en grupo que tendrá la forma de un laboratorio. Se plantean a los estudiantes uno o más problemas para su resolución. Aunque estos problemas pueden ser introducidos o discutidos inicialmente en clase, su estudio debe hacerse de forma no presencial. Los estudiantes tienen asignado un docente que oficia de tutor. Se pueden realizar otras actividades adicionales que los docentes consideren oportunas, como por ejemplo, controles de lectura.

Carga horaria:

- Clases teóricas - 4hs semanales.
- Estudio no presencial, trabajo en el laboratorio (sumando las reuniones con el tutor) - 6hs semanales.

5. TEMARIO

1. Introducción a la ingeniería de software:

Subtemas: Descripción del desarrollo profesional de software, descripción de la ingeniería de software, ética en la ingeniería de software y algunos casos de estudio que permitan introducir los problemas particulares a tratar en el resto del curso.

2. Procesos de software: *Un proceso de software es un conjunto coherente de actividades para la producción de software.*

Subtemas: Descripción de los modelos de procesos de software: modelo en cascada, desarrollo incremental e integración y configuración (reuso de software). Descripción de la actividades del proceso: especificación del software, diseño e implementación, validación y evolución. Cómo hacer frente al cambio: prototipado, entrega incremental. Mejora del proceso. Desarrollo ágil de software: descripción de los métodos ágiles, técnicas ágiles de desarrollo, gestión ágil de proyectos, escalando los métodos ágiles.

3. Ingeniería de requisitos: *Descripción de las funciones o servicios que el sistema proveerá y de las posibles restricciones.*

Subtemas: Requerimientos funcionales y no funcionales. Proceso de ingeniería de requisitos. Obtención de requisitos: técnicas, historias y escenarios. Especificación de requisitos. Validación de requisitos. Cambio en los requisitos. Modelado del sistema: modelos de contexto, modelos de interacción, modelos estructurales, modelos de comportamiento, arquitectura guiada por modelos.

4. Diseño de software: *Descripción de la organización general del sistema y diseño de sus componentes e interacciones.*

Subtemas: Diseño de la arquitectura: decisiones, vistas, patrones y arquitecturas de aplicaciones. Diseño orientado a objetos usando UML. Patrones de diseño.

5. Construcción de software: *Construcción de la solución de software.*

Subtemas: Código robusto mediante manejo de excepciones, prácticas de programación defensiva, estrategias de integración, implementación de cambios.

6. Verificación y validación de software: *Su objetivo es mostrar que el software cumple con su especificación, encontrar comportamiento no esperado del software y apoyar en la validación de su adecuación al uso.*

Subtemas: Calidad del software. Revisiones e inspecciones. Descripción de pruebas durante el desarrollo, desarrollo guiado por las pruebas, pruebas de liberación y pruebas de usuario.

7. Evolución del software: Descripción de los cambios que sufre el software luego de su liberación.

Subtemas: Proceso de evolución. Sistemas legados. Mantenimiento de software.

8. Gestión de proyectos de software: Descripción de las principales actividades en la gestión de proyectos de software y aplicación de algunas técnicas específicas.

Subtemas: Gestión de riesgos del proyecto: identificación, análisis, planificación y monitoreo. Gestión de las personas: motivación, trabajo en equipo, selección, organización y comunicación. Planificación y seguimiento del proyecto: precios del software, desarrollo basado en planes, cronograma del proyecto, planificación ágil, técnicas de estimación.

9. Herramientas y ambientes: Herramientas que se utilizan y su contribución a la ingeniería de software.

Subtemas: Gestión de la configuración del software y control de versiones. Gestión de liberaciones. Herramientas de modelado de análisis de requisitos y diseño. Herramientas para pruebas (dinámicas y estáticas). Herramientas para armado de versiones.

10. Confiabilidad del software: Conceptos de la ingeniería de confiabilidad.

Subtemas: Disponibilidad y confiabilidad. Requisitos de confiabilidad. Arquitecturas tolerantes a fallas. Programación de software confiable. Medición de la confiabilidad.

6. BIBLIOGRAFÍA

Tema	Básica	Complementaria
Introducción a la ingeniería de software	(1) - Capítulo 1	
Procesos de software	(1) - Capítulos 2 y 3	
Ingeniería de requisitos	(1) - Capítulos 4 y 5	
Diseño de software	(1) - Capítulos 6 y 7	
Construcción de software	(1) - Capítulo 7	
Verificación y validación de software	(1) - Capítulos 8 y 24	
Evolución del software	(1) - Capítulo 9	
Gestión de proyectos de software	(1) - Capítulos 22 y 23	
Herramientas y ambientes	(1) - Capítulo 25	
Confiabilidad del software	(1) - Capítulo 11	

6.1 Básica

1. Sommerville, Ian (2015). Software Engineering (10th Edition). USA: Pearson.

6.2 Complementaria

N/A

7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

7.1 Conocimientos Previos Exigidos: Conocimientos teóricos y prácticos de la programación imperativa y orientación a objetos.

7.2 Conocimientos Previos Recomendados: —

ANEXO A Para todas las Carreras

A1) INSTITUTO

Instituto de Computación.

A2) CRONOGRAMA TENTATIVO

Semana 1	Introducción a ingeniería de software (4hs de clase).
Semana 2	Procesos de software (4hs de clase).
Semana 3	Procesos de software (2hs de clase). Ingeniería de requisitos (2hs de clase).
Semana 4	Ingeniería de requisitos (4hs de clase).
Semana 5	Ingeniería de requisitos (2hs de clase). Diseño de software (2hs de clase).
Semana 6	Diseño de software (4hs de clase).
Semana 7	Construcción de software (4hs de clase).
Semana 8	Verificación y validación (4hs de clase).
Semana 9	Verificación y validación (4hs de clase).
Semana 10	Verificación y validación (2hs de clase). Evolución de software (2hs de clase).
Semana 11	Evolución de software (2hs de clase). Gestión de proyectos de software (2hs de clase).
Semana 12	Gestión de proyectos de software (4hs de clase).
Semana 13	Gestión de proyectos de software (2hs de clase). Herramientas y ambientes (2hs de clase).
Semana 14	Herramientas y ambientes (2hs de clase). Confiabilidad del software (2hs de clase).
Semana 15	Confiabilidad del software (4hs de clase).

A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

Dos pruebas parciales a efectuarse durante el curso que sumarán entre 80% y 90% del puntaje del curso y actividades adicionales que permitan fomentar el involucramiento del estudiante a lo largo del curso (trabajo grupal en formato laboratorio, controles de lectura, etc.) que sumarán entre 10% y 20% del puntaje del curso.

Escala de evaluación:

- Reprobación: Puntaje por debajo del 25%.
- Aprobación: Puntaje entre 25% y 60%. Teniendo en cuenta que se debe obtener por lo menos un 25% en las pruebas parciales y por los menos un 25% en las actividades adicionales.

Aprobado por resolución N°113 del CFI de fecha 04.07.2017

- Exoneración: Puntaje igual o mayor a 60%. Teniendo en cuenta que se debe obtener por lo menos un 60% en las pruebas parciales y por lo menos un 60% en las actividades adicionales.

A4) CALIDAD DE LIBRE

Los estudiantes no podrán acceder a la Calidad de Libre.

A5) CUPOS DE LA UNIDAD CURRICULAR

Sin cúpo.

ANEXO B para las carreras Ingeniería en Computación (plan 97) y Licenciatura en Computación

B1) ÁREA DE FORMACIÓN

Ingeniería de Software

B2) UNIDADES CURRICULARES PREVIAS

Para el Curso: Cursos de Fundamentos de Bases de Datos, Programación 4 y Taller de Programación

Para el Examen: Curso de Introducción a la Ingeniería de Software.

ANEXO B para las carreras Ingeniería en Computación (plan 87)

B1) ÁREA DE FORMACIÓN

No corresponde

B2) UNIDADES CURRICULARES PREVIAS

Para el Curso: Curso de Taller III
Examen de Bases de Datos

Para el Examen: Curso de Introducción a la Ingeniería de Software.