



PROGRAMA DE ASIGNATURA

I. IDENTIFICACIÓN				
Carrera o programa: Ingeniería en Tecnologías de Información				
Unidad responsable: Escuela de Ingeniería				
Nombre de la asignatura: Ingeniería de Software				
Código: ECIN-00610				
Semestre en la malla¹: 5				
Créditos SCT – Chile: 5				
Ciclo de Formación	Básico		Profesional	X
Tipo de Asignatura	Obligatoria	X	Electiva	
Clasificación de área de conocimiento²				
Área: Ingeniería y Tecnología			Subárea: Ingeniería Informática	
Requisitos:				
Prerrequisitos:			Requisitos para:	
<ul style="list-style-type: none">Estructuras de Datos			<ul style="list-style-type: none">Desarrollo de Soluciones Web/MóvilSistemas de InformaciónPráctica Profesional	

II. ORGANIZACIÓN SEMESTRAL							
Horas Dedicación Semanal (Cronológicas)	Docencia Directa	4.5		Trabajo Autónomo	3.5	Total	8
Detalle Horas Directas	Cátedra	Ayudantía	Laboratorio	Taller	Terreno	Exp. Clínica	Supervisión
	3		1.5				

III. APOORTE AL PERFIL DE EGRESO
La asignatura contribuye al dominio 1 del perfil de egreso, “Conocimiento científico y disciplinario” y dominio 2 “Habilidades y Actitudes Personales y Profesionales”. También contribuye al dominio 4 “Habilidades para la Práctica de la Ingeniería”. Al finalizar la asignatura, los estudiantes serán capaces de realizar el análisis, diseño orientado a objetos y construcción de un producto de software de calidad que satisfaga los requisitos, necesidades y restricciones detectadas, utilizando un proceso iterativo e incremental. En este análisis y diseño, se utilizará el lenguaje de modelado UML y se aplicarán buenas prácticas de la industria.

¹ Este campo sólo se completa en caso de carreras con programas semestrales.

² Clasificación del curso de acuerdo a la OCDE



IV. COMPETENCIAS

La carrera declara las siguientes habilidades:

- 1.3. Aplicar conocimientos, métodos y herramientas con un enfoque sistémico principalmente en la ejecución de proyectos TI en el ámbito de las ciencias de la computación, infraestructura TI e ingeniería de software.
- 2.4. Demostrar habilidades personales que contribuyen a una práctica exitosa de la ingeniería: iniciativa, toma de decisiones, perseverancia, pensamiento crítico, aprendizaje continuo, pensamiento creativo, orientación al logro, flexibilidad, autoevaluación, gestión del tiempo y recursos.
- 4.3. Concebir soluciones que involucren, por ejemplo, aplicaciones TI, infraestructura TI, toma de decisiones, gestión de datos y gestión de proyectos.
- 4.4. Diseñar soluciones que involucren, por ejemplo, aplicaciones TI, infraestructura TI, toma de decisiones, gestión de datos y gestión de proyectos.
- 4.5. Implementar soluciones que involucren, por ejemplo, aplicaciones TI, infraestructura TI, toma de decisiones, gestión de datos y gestión de proyectos.
- 4.7. Participar en iniciativas de innovación de nuevos productos, procesos o servicios.

V. RESULTADOS DE APRENDIZAJE

1. Aplicar los conceptos asociados a los procesos de Ingeniería de Software.
2. Seleccionar el modelo de proceso más adecuado para la aplicación a desarrollar.
3. Aplicar el proceso de Ingeniería de Requisitos para la obtención de las necesidades de los clientes y usuarios.
4. Diseñar un producto de software utilizando UML.
5. Construir un producto de software de alta calidad siguiendo las mejores prácticas recomendadas por los modelos de procesos.
6. Realizar actividades de aseguramiento de la calidad en el desarrollo de software.
7. Identificar los objetivos y requisitos de las soluciones TIC.
8. Seleccionar los procesos, técnicas y herramientas adecuados de acuerdo con los requisitos.
9. Desarrollar la solución tecnológica más apropiada basada en las características del problema y los recursos disponibles.

VI. ÁREAS TEMÁTICAS

1. Proceso de Ingeniería de Software
 - 1.1 Definición del proceso de software
 - 1.2 Ciclo de vida del software
 - 1.3 Modelos de proceso de desarrollo
 - 1.4 Metodologías ágiles
 - 1.5 Desarrollo de Software Basado en Componentes
 - 1.6 Áreas claves de procesos
2. Ingeniería de Requisitos
 - 2.1 Fundamentos de Ingeniería de requisitos
 - 2.2 Proceso de requisitos
 - 2.3 Elicitación de requisitos
 - 2.4 Análisis de Requerimientos
 - 2.5 Especificación de requerimientos



2.6 Validación de requisitos

2.7 Gestión de requisitos

3. Gestión de Calidad

3.1 Plan de calidad

3.2 Testing de software

3.3 Actividades de Validación y Verificación

4. Diseño de Software

4.1 Fundamentos de Diseño de Software

4.2 Aspectos claves del diseño de software

4.3 Estructura y Arquitectura de Software

4.4 Diseño de Interfaz de Usuario

4.5 Análisis de calidad y evaluación del diseño de software

4.6 Notación de diseño de software

4.7 Métodos y estrategias de diseño de software

4.8 Patrones de diseño

5. Construcción de Software

5.1 Construcción de software seguro

5.2 Herramientas de desarrollo de software

5.3 Herramientas de gestión de configuración y gestión de versiones.

5.4 Bug tracking

VII. ORIENTACIONES METODOLÓGICAS

La asignatura se implementa bajo la metodología A+S.

1. La metodología a desarrollar en esta asignatura es principalmente práctica, por lo que se debe fomentar la interacción entre los estudiantes a través de trabajos prácticos colaborativos.
2. Se sugiere abordar la teoría mediante metodologías activas como la clase invertida, el aprendizaje basado en problemas (ABP), entre otras, creando oportunidades para presentaciones orales individuales y/o grupales, promoviendo así el aprendizaje contextualizado.
3. Las experiencias en la cátedra y el taller deben llevarse a cabo utilizando software moderno aplicable a la asignatura.
4. Se sugiere que la metodología A+S se aplique en los talleres del curso.
5. Se recomienda que los estudiantes realicen presentaciones periódicas sobre el trabajo realizado en el taller.

VIII. ORIENTACIONES Y CRITERIOS PARA EVALUACIÓN

1. Se recomienda la aplicación de una evaluación diagnóstica al inicio de la asignatura.
2. En la cátedra, se sugiere implementar estrategias de evaluación sumativa, con un peso del 70% en la nota final de la asignatura.
3. En el taller, se sugiere implementar estrategias de evaluación sumativa, con un peso del 30% en la nota final de la asignatura.
4. Las actividades pueden ser individuales o grupales.
5. La asistencia mínima requerida para las actividades de cátedra es del 70%.
6. Se requerirá un logro del 60% de los objetivos para aprobar las actividades de evaluación.



7. Se recomienda llevar a cabo evaluaciones de carácter formativo con retroalimentación de carácter personal.

IX. RECURSOS BIBLIOGRÁFICOS

Bibliografía Mínima

– Larman, C. (2003). UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado. España: Pearson Educación.

Bibliografía Complementaria

– Sommerville, I. (2016). Software Engineering. Reino Unido: Pearson.

– Gamma, E. (2002). Patrones de diseño. España: Pearson Educación.