



## Programa de Asignatura

I. IDENTIFICACIÓN				
<b>Carrera o programa:</b> Ingeniería Civil en Computación e Informática				
<b>Unidad responsable:</b> Escuela de Ingeniería				
<b>Nombre de la asignatura:</b> Ingeniería de Software				
<b>Código:</b> ECIN-00610				
<b>Semestre en la malla<sup>1</sup>:</b> 6				
<b>Créditos SCT - Chile:</b> 5				
<b>Ciclo de Formación</b>	Básico		Profesional	X
<b>Tipo de Asignatura</b>	Obligatoria	X	Electiva	
<b>Clasificación de área de conocimiento<sup>2</sup></b>				
<b>Área:</b> Ingeniería y Tecnología		<b>Sub área:</b> Ingeniería Informática		
<b>Requisitos:</b>				
<b>Pre-requisitos:</b>		<b>Requisito para:</b>		
• ECIN-00407 Estructuras de Datos		• ECIN-08606 Práctica Profesional		

II. ORGANIZACIÓN SEMESTRAL							
<b>Horas Dedicación Semanal (Cronológicas)</b>	Docencia Directa	4.5	Trabajo Autónomo	3	Total	7.5	
<b>Detalle Horas Directas</b>	Cátedra	Ayudantía	Laboratorio	Taller	Terreno	Exp. Clínica	Supervisión
	3		1.5				

<sup>1</sup>Este campo

<sup>2</sup>Clasificación del curso de acuerdo a la OCDE



### III. APOORTE AL PERFIL DE EGRESO

La asignatura contribuye al dominio 1 del perfil de egreso, “Conocimiento científico y disciplinario”. Además, contribuye al dominio 4 “Habilidades para la Práctica de la Ingeniería”. Al finalizar la asignatura las y 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, en base a un proceso iterativo e incremental. En este análisis y diseño utilizará el lenguaje de modelado UML, y será capaz de aplicar buenas prácticas de la industria.

### IV. HABILIDADES PERFIL DE EGRESO (RELACIÓN)

1.3 Aplicar conocimientos, métodos y herramientas de la especialidad para resolver problemas complejos de Ingeniería de Software, Plataformas y Gestión de Tecnologías.

4.3 Concebir las soluciones TIC requeridas en las organizaciones haciendo uso eficiente de los recursos como personas, información, y procesos. Estas soluciones consideran las arquitecturas TI junto a sus modelos de servicios y modelos operativos; los sistemas de software; y las plataformas de cómputo y comunicaciones junto a sus servicios asociados.

4.4 Diseñar las soluciones TIC para la industria intensiva en procesamiento de la información. Estas soluciones consideran las arquitecturas TI junto a sus modelos de servicios y modelos operativos; los sistemas de software; y las plataformas de cómputo y comunicaciones junto a sus servicios asociados.

4.5 Implementar las soluciones TIC. Estas soluciones consideran las arquitecturas TI junto a sus modelos de servicios y modelos operativos; los sistemas de software; y las plataformas de cómputo y comunicaciones junto a sus servicios asociados.



## V. RESULTADOS DE APRENDIZAJE

1. Comprender los conceptos asociados a los procesos de la 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 elicitación de las necesidades de los clientes y usuarios
4. Diseñar un producto de software utilizando UML. Este diseño considera el diseño de alto nivel, la arquitectura del sistema y la interfaz de usuario
5. Construir un producto de software de calidad en base a las buenas práctica recomendadas por los modelos de procesos
6. Realizar actividades de aseguramiento de la calidad en el desarrollo de software
7. Identificar los objetivos y requerimientos de las soluciones TIC
8. Seleccionar los procesos, técnicas y herramientas adecuados de acuerdo a los requerimientos.
9. Desarrollar la solución tecnológica más adecuada en base a las características del problema y los recursos disponibles.

## VI. ÁREAS TEMÁTICAS

1. 1. Proceso de Ingeniería de Software (1 semana)
  - 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 (4 semana)
  - 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 (2 semana)
  - 3.1 Plan de calidad
  - 3.2 Testing de software
  - 3.3 Actividades de Validación y Verificación
- 4. Diseño de Software (5 semana)
  - 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 (4 semana)
  - 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

1. La metodología a desarrollar en esta asignatura es principalmente práctica, por lo que se debe favorecer la interacción entre los y las estudiantes, a través de trabajos prácticos colaborativos.
2. Se sugiere trabajar la teoría a través de metodologías activas como clase invertida, ABP, entre otras, generando instancias de presentación oral individual y/o grupal, favoreciendo el aprendizaje contextualizado.
3. Las experiencias de cátedra y taller deben ser realizadas por medio de la utilización de software moderno aplicable a la asignatura.
4. Se recomienda que las y los estudiantes realicen presentaciones periódicas sobre el trabajo realizado en taller.



## VII. ORIENTACIONES Y CRITERIOS PARA LA EVALUACIÓN

1. Se recomienda la aplicación de una evaluación diagnóstica al inicio de la asignatura.
2. En cátedra, se sugiere implementar estrategias de evaluación sumativas, con una ponderación del 70% de la nota final de la asignatura.
3. En taller se sugiere implementar estrategias de evaluación sumativas, con una ponderación del 30% de la nota final de la asignatura.
4. Las actividades podrán ser individuales o grupales.
5. La asistencia mínima exigida para las actividades de cátedra es del 70%.
6. Se exigirá un 60% de logro de objetivos para aprobar las actividades de evaluación.
7. Se recomienda realizar evaluaciones de carácter formativo con retroalimentación de carácter personal.

## IX. RECURSOS BIBLIOGRÁFICOS

### **Bibliografía Mínima**

- Pierre Bourque, Richard E. Fairley. SWEBOK V3.0 Guide to the Software Engineering Body of Knowledge

### **Bibliografía Complementaria**

- Ian Sommerville. Ingeniería de Software
- Craig Larman. UML
- G. Gordon Schulmeyer. Handbook of Software Quality Assurance, Cuarta Edición
- VV.AA. Patrones De Diseño: Elementos De Software Orientado a Objetos