

PROGRAMA DE ASIGNATURA

I. IDENTIFICACIÓN DE LA ASIGNATURA

Asignatura: Análisis y Diseño de Software		Sigla: INF-236	Fecha de aprobación 11/10/2016 (Acuerdo CC.DD. 13/2016)		
Créditos UTFSM : 3	Prerrequisitos: INF-253 INF-239	Examen: No tiene	Unidad Académica que la imparte.		
Créditos SCT : 5			Departamento de Informática		
Horas Cátedra Semanal : 3	Horas Ayudantía Semanal : 1,5	Horas Laboratorio Semanal: 0	Semestre en que se dicta		
			Impar	Par X	Ambos
Eje formativo : Ingeniería Aplicada - Desarrollo de Software					
Tiempo total de dedicación a la asignatura : 150 horas cronológicas					

Descripción de la Asignatura

El estudiante construye software utilizando técnicas de análisis y diseño orientado a objetos. Desarrolla un pequeño sistema Web de 3 capas, aplicando técnicas de casos de uso para el análisis y especificación de requisitos de proyectos pequeños, y técnicas orientadas a objeto (patrones y frameworks) para el diseño de sistemas de software a partir de las especificaciones. Las actividades prácticas fortalecen el trabajo en equipo.

Requisitos de entrada

- Comprender el paradigma orientado a objetos.
- Programar en algún lenguaje orientado a objetos (Java, C++ u otros).
- Diseñar modelos de datos e implantarlos en bases de datos.
- Comprender textos técnicos en inglés.

Contribución al perfil de egreso

Competencia Específica

- Desarrollar, implantar y mantener sistemas de software confiable, eficiente y factible.

Competencias Transversales

- Comunicar información oral y escrita de manera eficaz al interior de las organizaciones en las que se desempeña, como con entidades del entorno.
- Actuar con autonomía, flexibilidad, iniciativa, y pensamiento crítico al enfrentar problemáticas de la profesión.
- Desarrollar su quehacer con sólidos criterios que le permitan asegurar calidad desde una perspectiva sistémica.

Resultados de Aprendizaje que se espera lograr en esta asignatura

- **Analiza** diversos dominios, **aplicando** paradigmas de orientación a objetos: herencia y polimorfismo.
- **Elabora** especificaciones de requisitos funcionales para sistemas Web, **aplicándolos** en casos de uso.
- **Elabora** modelos estructurales y dinámicos de software orientado a objetos, utilizando las especificaciones de casos de uso.
- **Identifica** oportunidades de aplicación de patrones de diseño de software, **aplicándolos** a situaciones concretas de diseño.
- **Evalúa** modelos de diseños de software, **ajustándolos** a requisitos y calidad intrínseca.
- **Describe** la lógica utilizada para llegar a la solución propuesta, **argumentando** sus decisiones de análisis y diseño.

Contenidos temáticos

- Orientación a objetos: objetos, tipos/polimorfismo, clases/herencia.
- Requisitos de software: casos de uso, modelos de dominio.
- Modelado dinámico: colaboraciones, secuencias, estados, actividades.
- Diseño preliminar: modelos de clases.
- Técnicas de diseño: frameworks orientados a objeto, patrones de diseño.
- Patrones de diseño para sistemas Web: presentación, concurrencia, persistencia.
- Calidad de diseño: métricas, revisiones de diseño, trazabilidad.

Metodología de enseñanza y aprendizaje

- Clases expositivas con apoyo de medios audiovisuales.
- Aprendizaje basado en el análisis y discusión de casos.
- Aprendizaje basado en el desarrollo de un proyecto.
- Estudio independiente y exposiciones de estudiantes acerca de temas específicos.

Evaluación y calificación de la asignatura (Ajustado a Reglamento Institucional-Rglto. N°1)

Requisitos de aprobación y calificación	<p>Proceso de evaluación y calificación:</p> <p>La nota final se calcula como el promedio ponderado de las notas de:</p> <ul style="list-style-type: none"> • dos certámenes (C_1 y C_2) 30% c/u. • tareas asociadas a un proyecto (T) 30% y, • participación y presentaciones (NP) - 10% <p>• Promedio semestral (PS) se calcula según: Si $(C_1+C_2)/2 \geq 50$ & $T \geq 50$ & $NP \geq 50$ $PS = C_1 * 30\% + C_2 * 30\% + T * 30\% + NP * 10\%$ Sino $PS = \min((C_1 + C_2)/2, T, NP)$</p>
---	--

Recursos para el aprendizaje.

Plataforma virtual

Bibliografía:

Texto Guía	<ul style="list-style-type: none"> • Martin, Fowler. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley Professional. • Doug Rosenberg and Matt Stephens. (2007). Use Case Driven Object Modeling with UML: Theory and Practice. Apress.
Complementaria u Opcional	<ul style="list-style-type: none"> • Alistair Cockburn.(2000).Writing Effective Use Cases Addison-Wesley Professional • Daryl Kulak and Eamonn Guiney. (2003). Use Cases: Requirements in Context (2nd Ed.) Addison-Wesley Professional. • Axel van Lamsweerde. (2009) Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley.

II. CÁLCULO DE CANTIDAD DE HORAS DE DEDICACIÓN- (SCT-Chile)- CUADRO RESUMEN DE LA ASIGNATURA

ACTIVIDAD	Cantidad de horas de dedicación		
	Cantidad de horas por semana	Cantidad de semanas	Cantidad total de horas
PRESENCIAL			
Cátedra o Clases teóricas	3	15	45
Ayudantía/Ejercicios	1,5	14	21
Visitas industriales (de Campo)			
Laboratorios / Taller			
Evaluaciones (certámenes, otros)	2	2	4
Otras (Especificar)			
NO PRESENCIAL			
Ayudantía			
Tareas obligatorias	3	16	48
Trabajo Individual y Grupal			
Otras: Estudio Personal	2	16	32
TOTAL (HORAS RELOJ)			150
Número total en CRÉDITOS TRANSFERIBLES			5