

# Curso: Arquitectura de Software Aplicada

**Profesor:** Raúl Santelices A.

IIC 2122

10 créditos, OPR

Horario: M J: 4

Semestre: 1' 2003

Cupos: 50

## Objetivos

- Interiorizar a los alumnos con los principios de un diseño modular, mantenible y reusable de sistemas de software.
- Entregar las herramientas de decisión que permitan estructurar un sistema de software, tanto en lo general como en lo específico.
- Familiarizar al alumno con las técnicas y soluciones recurrentes empleadas en esta disciplina, haciendo énfasis en problemas prácticos.

## Requisitos

Ingeniería de Software (IIC 2142)

## Contenidos

- Introducción: ¿qué es *arquitectura*?
- Motivación: sistemas que crecen sin control. Ejemplos y anti-ejemplos.
- Sistemas modulares de software. Software basado en componentes.
- Modularidad lineal, multiproceso y distribuida.
- Principios: robustez, reusabilidad, portabilidad y mantenibilidad.
- Variables de decisión: cohesión y acoplamiento.
- La necesidad de orden y jerarquía. Arquitectura de capas.
- Orientación a objetos y su papel en la arquitectura.
- Distribución de responsabilidades. Diseño de colaboraciones.
- Interfaces y polimorfismo. Ocultamiento de la implementación.
- Patrones de diseño: introducción. Aplicación a nivel de arquitectura.
- Patrones estructurales y creacionales.
- Patrones de interacción entre módulos.
- Metodologías: diseño iterativo e incremental. Adaptación y generalización.
- Frameworks vs. aplicaciones individuales.
- Arquitecturas cerradas vs. arquitecturas abiertas. Extensibilidad.

## Metodología

Las clases son teórico-prácticas, en estilo tutorial. Se usarán ejemplos de casos ficticios y también reales para introducir tanto los principios generales como las técnicas contenidas en el curso. Algunos de estos casos serán propuestos para discusión en clase, y las diversas soluciones que surjan serán analizadas y comparadas. Se hará énfasis en la importancia de pensar en abstracto con el fin de analizar arquitecturas y extraer patrones.

Junto con las clases, los alumnos deberán completar una serie de tareas, divididas en dos etapas. La primera consiste en resolver una serie de problemas, cuyo objetivo es afianzar la comprensión de los principios fundamentales de la arquitectura de software. La segunda parte, en tanto, consiste en desarrollar un pequeño proyecto en pasos incrementales, que será evaluado y corregido con cada entrega.

## Evaluación

- 2 interrogaciones (15% c/u) y 1 examen (20%).
- 6 tareas (50%). Las 3 últimas conforman el proyecto final.

Se requiere promedio 4.0 en los dos puntos anteriores para aprobar.

## Bibliografía

- Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*. 2001, segunda edición. Prentice Hall, Upper Saddle River, NJ.
- Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. 1995. Addison-Wesley.
- Frank Buschmann et al. *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns*. 1996. John Wiley & Son Ltd, Inglaterra.
- Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. 1998. Addison-Wesley y ACM Press, NY.