

# Diseño de Sistemas Informáticos

4º Ingeniería Informática 2.009/10

1º Programa Oficial de Postgrado 2.009/10

Profesores:

---

Ingeniería Informática	Víctor M. Gulías ( <a href="mailto:gulias@udc.es">gulias@udc.es</a> ) David Cabrero ( <a href="mailto:cabrero@udc.es">cabrero@udc.es</a> ) Laura Castro ( <a href="mailto:lcastro@udc.es">lcastro@udc.es</a> ) Javier París ( <a href="mailto:javierparis@udc.es">javierparis@udc.es</a> ) David Martínez ( <a href="mailto:dmartinez@udc.es">dmartinez@udc.es</a> )
Máster en Informática	Víctor M. Gulías ( <a href="mailto:gulias@udc.es">gulias@udc.es</a> ) David Cabrero ( <a href="mailto:cabalde@udc.es">cabalde@udc.es</a> )

---

*“First create a solution using sound software engineering techniques, then if needed, introduce small violations of good software engineering principles for efficiency’s sake.”*

Donald Knuth

## Objetivos

- Comprender la necesidad de abordar el desarrollo de sistemas de información utilizando los **conceptos propios de la Ingeniería del Software**
- Conocer algunas **técnicas y herramientas de Ingeniería del Software** para el diseño y construcción de aplicaciones
- Valorar la **utilidad del diseño** para construir y modificar aplicaciones
- Valorar las ventajas del **uso de un lenguaje de modelado unificado**
- Identificar los **problemas típicos de diseño** y algunas **soluciones fruto de la experiencia** que los abordan satisfactoriamente

## Programa

- Introducción
  - Definición y objetivos
  - El diseño como fase en el proceso de desarrollo
  - Principios básicos de diseño
  - Nociones de orientación a objetos y modelado UML
- Patrones de Diseño
  - Concepto de patrón de diseño
  - Composicionalidad (*Composición*)
  - Asignación de responsabilidades (*Proxy, Decorador*)
  - Delegación (*Decorador, Cadena de Responsabilidad*)
  - Bajo acoplamiento e instanciación (*Prototipo, Negociador, Instancia Única*)
  - Alta cohesión y división en subsistemas (*Fachada*)
  - Modelado de comportamiento dinámico (*Estado*)
  - Reutilización (*Estrategia, Iterador*)
  - Dependencias entre objetos (*Observador*)
  - Refactorización del diseño (*Método de Fabricación, Método Plantilla*)
  - Preparación para el cambio (*Visitante*)
  - Adaptación de interfaces (*Adaptador, Puente, Fábrica abstracta*)
  - Encapsulación y abstracción (*Comando, Recuerdo*)
  - Manejo de la complejidad (*Mediador*)

## Horario

- Teoría (**Aula 3.3**)

● Mañana:	Lunes	11:30-12:30	Tarde (Máster):	Lunes	16:30-17:30
	Martes	11:30-12:30		Martes	16:30-17:30
	Viernes	11:30-12:30		Miércoles	16:30-17:30

- Prácticas (**Laboratorio 1.3, JAVA**):

● Mañana:	Jueves	10:30-12:30	Tarde:	Jueves	15:30-17:30
	Viernes	8:30-10:30		Jueves	17:30-19:30
	Viernes	12:30-14:30		Viernes	15:30-17:30 (Master)

- Tutorías:

V́ctor M. Gulías (Despacho 4.15)	Lunes	12:30-14:30
	Lunes	18:30-20:30
	Martes	12:30-14:30

## Evaluación (Ingeniería Informática)

- Teoría
  - Examen de cuestiones teóricas + problema diseño (este año **sin** examen de prácticas)
  - Supone 2/3 de la calificación final
- Prácticas
  - Prácticas en parejas, evaluación **independiente**
  - Dos/tres prácticas cortas + una práctica final
  - Valoración de prácticas
    - 1/3 evaluación continua (prácticas cortas, participación)
    - 1/3 evaluación de práctica final
    - 1/3 examen de prácticas (primera semana de junio)
  - Supone 1/3 de la calificación final

## Bibliografía

- Básica:
  - E. Gamma, R. Helm, R. Johnson, J. Vlissides. *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series [on-line](#)
  - M. Grand. *Patterns in Java, a catalog of reusable design patterns illustrated with UML*. volumen I, John Wiley & sons
- Adicional sobre patrones de diseño y frameworks:
  - J. Vlissides. *Pattern Hatching. Design Patterns Applied*. Addison-Wesley
  - M. Grand. *Patterns in Java*. volumen II, John Wiley & sons
  - W. Brown, R. Malveau, H. McCormick, T. Mowbray. *Anti-Patterns*. J. Wiley & sons
  - D. Govoni. *Java Application Frameworks*. John Wiley & sons
  - M. Fowler. *Analysis Patterns. Reusable Object Models*. Addison-Wesley.
  - T. Mowbray, R. Malveau. *CORBA Design Patterns*. John Wiley & sons
- UML y proceso de desarrollo software:
  - C. Larman. *Applying UML and patterns*. Prentice Hall
  - G. Booch, J. Rumbaugh, I. Jacobson. *The Unified Modeling Language*. Addison-Wesley Longman
  - M. Fowler, K. Scott. *UML Distilled* Addison-Wesley
- Programación Java:
  - K. Arnold, J. Gosling. *The Java Programming Language*. Addison-Wesley