

## **M2: Programación Orientada a Objetos: Lenguajes, metodología y herramientas.**

Master de Computación

Facultad de Ciencias

Universidad de Cantabria.

Profesores: José M. Drake [drakej@unican.es](mailto:drakej@unican.es)

Elena Mediavilla Bolado [mediavie@unican.es](mailto:mediavie@unican.es)

Patricia López Martínez [lopezpa@unican.es](mailto:lopezpa@unican.es)

### **I.- Página WEB de la asignatura:**

[http://www.ctr.unican.es/asignaturas/MC\\_OO/index.html](http://www.ctr.unican.es/asignaturas/MC_OO/index.html)

### **II.- Objetivos generales de la asignatura**

- Conocer y practicar el paradigma de programación orientada a objetos para especificar, diseñar, implementar y verificar aplicaciones informáticas haciendo uso de herramientas CASE.
- Aprender a analizar un problema utilizando la metodología de orientación a objetos y saber realizar su descomposición modular en subsistemas, componentes y objetos funcionalmente correctos, robustos y reutilizables.
- Conocer y llevar a cabo un proceso de desarrollo de aplicaciones que sea adecuado para el trabajo en equipo, y que pueda ser realizado en un entorno de herramientas automáticas de especificación, análisis, diseño y generación automática de código.
- Conocer los lenguajes UML para la representación simbólica de software. Así como saber aplicarlo en las fases de diseño del software.
- Conocer los principios de programación basada en componentes. Analizar la tecnología de componentes JavaBeans y los patrones de diseño en que se basa. Conocer y saber aplicar los componentes del catálogo SWT para el desarrollo de interfaces de usuario (GUI).

### **III.- Capacidades que van ser adquiridas en el curso.**

- Ser capaz de modelar una aplicación informática utilizando el lenguaje gráfico UML.
- Ser capaz de especificar, analizar y diseñar aplicaciones informáticas utilizando el paradigma de orientación a objetos y de implementarlas utilizando el lenguaje JAVA.
- Saber desarrollar aplicaciones utilizando componentes JavaBean disponibles en catálogo, así como saber diseñar nuevos componentes reutilizables.

### **IV.- Programa de la asignatura.**

#### **Módulo I: Paradigma de programación orientada a objetos**

- Metodología orientada a objetos.
- Proceso de desarrollo orientado a objetos.

## **Módulo II: Modelos y herramientas UML**

- Especificación de la aplicación: Casos de usos.
- Modelado estructural.
- Modelado del comportamiento.

## **Módulo III: Proceso de desarrollo de aplicaciones orientada a objetos**

- Herramientas y productos en el desarrollo de una aplicación orientada a objetos.

## **Módulo IV.- Metodología de programación basada en componentes.**

- Tecnología de componentes y proceso de diseño basado en componentes.
- Componentes JavaBean
- Familia de componentes: SWT para el diseño de GUIs en la plataforma Eclipse.

La docencia se desarrolla a través de sesiones de docencia presencial y de trabajo dirigido que realiza el alumno individualmente pero como parte de un equipo.

Docencia presencial: *(Su calendario se describe en “Calendario y Horario”)*

- 1.- Desarrollo conceptual de la asignatura: Utilizando documentos PowerPoint se desarrollará de forma ordenada los conceptos e ideas que constituyen el cuerpo de la asignatura. *(Los documentos en formato .pdf pueden descargarse de la página WEB de la asignatura)*
- 2.- Aplicación de los conceptos a ejemplos: Su finalidad es doble: a través de los ejemplos se muestra la aplicación de los conceptos a casos reales, y a través de su ejecución en el laboratorio se aprenderá a utilizar las herramientas.

Trabajo dirigido en equipo: Consiste en un proyecto de complejidad similar a los ejemplos desarrollados en clase, que será asignado a grupos de tres alumnos. Cada alumno debe completarlo fuera del horario presencial aplicando los conceptos que se irán desarrollando en él.

## **VI.- Evaluación de la docencia.**

La evaluación de la asignatura se realizará con calificaciones en el rango 0-10. Se realizará a través de la presentación al profesor de la solución de proyecto individual que se le ha asignado.

El criterio de evaluación es: (Haber obtenido la funcionalidad de proyecto: 3.0 puntos; Defensa de la solución propuesta: 3.0 puntos; Idoneidad de los medios utilizados: 2.0 puntos y Originalidad y elegancia de la solución: 2.0 puntos)

La evaluación se realizará antes del día 19-4-08 (1 semana después de haber terminado el curso).

## **VII. Bibliografía**

La docencia de la asignatura se realizará siguiendo los textos:

- G. Booch: “Object-Oriented Analysis and Design” Benjamin/Cummings Publishing Company Inc., 1994.
- G.Booch, J.Rumbaugh y I. Jacobson: “The Unified Modelling Language” AddisonWesley, 1999.

- I. Jacobson, G.Booch y J.Rumbaugh: “The Unified Software Development process”. AddisonWesley, 1999.
- B.P. Douglass: “Real-Time UML: Developing Efficient Objects for Embedded Systems”. Addison-Wesley, 1998.
- B.P. Douglass: “Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems”. Addison-Wesley, 2003.
- K.Hamilton y R. Mules: “Learning UML 2.0” O’Reilly, 2006
- R. Englander: “Developing Java Beans” O’Reilly, 1997
- Sun Microsystems : JavaBeans 1.01 Specification .  
<http://java.sun.com/javase/technologies/desktop/javabeans/index.jsp>