



## INGENIERÍA DE PROYECTOS SOFTWARE CON TÉCNICAS DE ORIENTACIÓN A OBJETOS (SW-001)

### *Resumen del curso:*

*Este curso describe el desarrollo de software orientado a objetos de un sistema, desde la fase de definición hasta la de diseño detallado. Se sigue siempre un enfoque basado en la experiencia de numerosos proyectos ejecutados y en lecciones aprendidas, que evitarán problemas potenciales, a la vez que nos enseña a aplicar buenos hábitos en el desarrollo de proyectos software. Asimismo, a fin de hacer el curso más ameno, las lecciones teóricas se complementan con un ejemplo práctico de proyecto que se va desarrollando a lo largo de todo el curso.*

### *Resumen del contenido de los módulos*

#### **Módulo 1: Desarrollo orientado a objetos. Requerimientos de usuario**

*En el primer día del curso se presentan y se justifican las técnicas de orientación a objetos. A continuación, se aborda la problemática de los requerimientos de usuario, siempre desde un punto de vista práctico. Se presenta también el proyecto ejemplo que se utilizará a lo largo de todo el curso para ilustrar diversos aspectos del desarrollo de proyectos con orientación a objetos.*

#### **Módulo 2: Requerimientos hardware/software. Arquitecturas**

*En el segundo día del curso se aborda el problema de trasladar los requerimientos de usuario a requerimientos del software y del hardware, se dan criterios para poder tomar decisiones sensatas en esta etapa temprana del proyecto sobre las distintas alternativas tecnológicas que se nos ofrecen. A continuación, se dan técnicas para analizar el sistema software y dotarlo de una arquitectura.*

#### **Módulo 3: Diseño detallado**

*En el tercer día del curso se aborda el diseño detallado del sistema: aquí se desarrollan los conceptos típicos del diseño orientado a objeto, tales como la herencia de clases, el polimorfismo, etc. También se describen las técnicas para optimizar nuestro diseño. A continuación, se ofrece un recorrido rápido por las fases de implementación, pruebas, verificación, que quedan fuera del alcance de este curso.*



## INGENIERÍA DE PROYECTOS SOFTWARE CON TÉCNICAS DE ORIENTACIÓN A OBJETOS (SW-001)

### Módulo 1: Desarrollo orientado a objetos. Requerimientos de usuario

#### Módulo 1.1. Introducción

- ❖ Presentación del paradigma "orientado a objeto"
- ❖ Espacio del problema, espacio de las soluciones
- ❖ Revisión de los modelos tradicionales
- ❖ Ventajas e inconvenientes de la orientación a objetos
- ❖ Cuando aplicar la orientación a objetos
- ❖ Conceptos básicos
- ❖ En la práctica

#### Módulo 1.2. Ingeniería de sistemas software

- ❖ Fases de un proyecto software
- ❖ Desarrollo en cascada; desarrollo iterativo; metodologías ágiles
- ❖ Relación con la Ingeniería de Sistemas
- ❖ Estándares: RUP, ECSS, UML
- ❖ En la práctica

#### Módulo 1.3. Ingeniería de los requerimientos de usuario

- ❖ Puesta en valor: la importancia de los requerimientos
- ❖ Dificultades típicas: participación del cliente
- ❖ Descripción del proyecto de ejemplo
- ❖ Requerimientos de usuario
  - ✚ Alto nivel
  - ✚ Funcionales y no funcionales
  - ✚ Restricciones
- ❖ Técnicas de captura de requerimientos
  - ✚ Casos de uso
  - ✚ User stories
  - ✚ Reglas de negocio
  - ✚ Diagrama de contexto
  - ✚ Diseño dirigido por pruebas
- ❖ Qué obtenemos al final de esta fase
- ❖ En la práctica

### Módulo 2: Requerimientos hardware/software. Arquitecturas

#### Módulo 2.1. Ingeniería de requerimientos hardware y software

- ❖ Restricciones
- ❖ Sistemas heredados
- ❖ COTS
- ❖ Análisis de alternativas tecnológicas:
  - ✚ Sistema operativo
  - ✚ Lenguaje de programación
  - ✚ Interfaz gráfica
  - ✚ Persistencia
  - ✚ Comunicaciones
- ❖ Criterios de decisión



- ❖ Análisis de viabilidad: prototipos.
- ❖ Qué obtenemos al final de esta fase.
- ❖ En la práctica

## Módulo 2.2. Análisis del sistema orientado a objetos

- ❖ Descomposición del sistema: encontrar objetos
- ❖ Arquitectura
  - ✚ Identificar patrones principales
  - ✚ Identificar arquitectura tipo
  - ✚ Arquitectura en capas y módulos
  - ✚ Dependencias
  - ✚ Concurrencia
- ❖ Revisión del análisis, PDR, SDR.
- ❖ Qué obtenemos al final de esta fase
- ❖ En la práctica

## Módulo 3: Diseño detallado

### Módulo 3.1. Diseño del sistema orientado a objetos

- ❖ Diseño detallado: objetos, clases y relaciones entre clases
- ❖ Generalización y herencia
- ❖ Polimorfismo
- ❖ Interfases
- ❖ Composición
- ❖ Delegación
- ❖ Patrones de diseño
- ❖ Comportamiento dinámico
- ❖ Herramientas de ayuda al diseño
- ❖ Documentar un diseño
- ❖ Rediseñar para un sistema eficiente
- ❖ Rediseñar para unas dependencias manejables
- ❖ Revisión del diseño, CDR
- ❖ Qué obtenemos al final de esta fase
- ❖ En la práctica

### Módulo 3.2. Implementación del sistema

- ❖ Lenguajes de programación
- ❖ Entornos de desarrollo
- ❖ Pruebas y validación
- ❖ Verificación
- ❖ Volver a empezar: requerimientos, análisis, diseño, implementación y pruebas

### Módulo 3.3. Glosario

### Módulo 3.4. Documentos de Referencia