



Objetivos de Aprendizaje

El objetivo principal de la asignatura consiste en capacitar al estudiante para analizar, diseñar e implementar aplicaciones de entidad y complejidad manifiestamente superiores a las que están acostumbrados tras su primer año y medio de curriculum. Se pretende por tanto dotarles de herramientas y métodos que hagan viable el salto a la producción de un software más cercano al del mundo real. El corpus teórico que vertebra los distintos elementos utilizados en el curso es el Proceso Unificado de Desarrollo del Software (UP), considerado hoy en día como un estándar.

Así se introduce la dificultad de capturar los requisitos y producir un diseño correcto y eficaz, y al mismo tiempo se proporciona el lenguaje de modelado UML como herramienta en la que apoyarse para enfrentar dicha dificultad y desarrollar los proyectos siguiendo un proceso sistemático.

El desarrollo de software de calidad requiere que sus componentes sean reusables y que se pueda modificar a bajo coste para afrontar cambios en las necesidades o portabilidad a otros entornos, por lo que se introduce el concepto de arquitectura software de varios niveles.

Por último se introducen asimismo herramientas de desarrollo para incorporar a las aplicaciones desarrolladas por los estudiantes dos características ubicuas en las aplicaciones de uso real: la capa de persistencia y el entorno cliente/servidor.

Competencias:

Una vez finalizado el curso, el estudiante deberá ser capaz de:

1. Enunciar, explicar y seguir las diversas etapas que componen todo proceso de Ingeniería del Software.
2. Identificar los requisitos de un sistema propuesto a partir de las especificaciones (QUÉ QUIERE) de un determinado cliente.
3. Obtener un diseño a partir de los requisitos del sistema (CÓMO SE HACE).



4. Expresar de forma visual los resultados del análisis y diseño del sistema propuesto en el Proyecto, usando el lenguaje UML.
5. Usar una herramienta de diagramación (StarUML) para expresar los resultados del análisis y diseño.
6. Aplicar conceptos básicos de patrones en el diseño de una solución.
7. Usar un entorno de desarrollo (ECLIPSE) para implementar un sistema software cliente-servidor siguiendo una arquitectura de tres niveles.

Prerequisitos

El análisis de requisitos y el diseño se realizan utilizando el paradigma orientado a objetos. Los estudiantes necesitan cierta destreza en el manejo de los principios que sustentan dicho paradigma. Estos conocimientos se estudian en profundidad en las asignaturas Programación Modular y Orientada a Objetos y Estructuras de Datos y Algoritmos.

Respecto a la fase de implementación, para abordarla con garantías de éxito es imprescindible que el estudiante tenga soltura en la concepción, codificación y prueba de algoritmos en Java, utilizando un entorno de desarrollo como puede ser Eclipse. Esta capacidad se adquiere de manera gradual desde la asignatura introductoria Programación Básica, pasando por Programación Modular y Orientada a Objetos, y Estructuras de Datos y Algoritmos.

Sólo se asume un conocimiento superficial de Bases de Datos, ya que en el curso se utiliza un SGBDOO (Sistemas de Gestión de Bases de Datos Orientadas a Objetos) que permite realizar operaciones relativamente complejas sin necesidad de aparato teórico adicional.

Temario

TEMA 1: Introducción a la Ingeniería del Software

TEMA 2: Captura de requisitos usando UML

2.1.- Modelo de Casos de Uso



2.2.- Modelo del Dominio

TEMA 3: Diseño

3.1.- Diagramas de secuencia UML

3.2.- Patrones de responsabilidad GRASP

3.3.- Arquitecturas Software de varios niveles

TEMA 4: Implementación usando Java

4.1.- Interfaces de usuario gráficas: SWING/AWT

4.2.- Persistencia de objetos: db4o

4.3.- Computación distribuida: RMI

Descripción del Temario

A continuación se detallan los contenidos del temario.

Tema 1: En este tema se abordará la necesidad de métodos y procesos para el desarrollo de aplicaciones software de envergadura. Concretamente se presentará el Proceso Unificado como metodología que vertebrará las diferentes etapas en las que está dividida la asignatura. Finalmente se presentará el proyecto a desarrollar durante la asignatura.

Tema 2: En este tema se abordará la captura formal de los requisitos del proyecto presentado en el tema 1. El objetivo consiste en que el estudiante sea capaz de capturar y modelar los requisitos funcionales del sistema utilizando Casos de Uso y Flujos de Eventos, los requisitos de datos, a través del Modelo del Dominio, y la interdependencia que existe entre los diferentes modelos.

Tema 3: En este tema, el estudiante adquirirá la destreza necesaria para obtener los artefactos del diseño a partir de los requisitos obtenidos en el tema anterior. Más concretamente, se estudiará cómo obtener los diagramas de secuencia a partir de los casos de uso y el flujo de eventos, el modelo del diseño a partir del modelo del dominio, y cómo mantener la consistencia entre los diferentes diagramas.

Tema 4: En este tema se tratarán principalmente tecnologías de implementación de sistemas cliente-servidor (RMI) y bases de datos orientadas a objetos (Db4o).



Metodología

La asignatura incorporará como actividad destacada la realización de un Proyecto de software a lo largo del curso, que pretende que todos los conocimientos teóricos adquiridos en cada tema puedan ser puestos en práctica de manera inmediata en el proyecto. El proyecto se desarrollara siguiendo el proceso unificado, lo que conlleva el desarrollo de 3 actividades:

Captura de requisitos del proyecto: Antes de abordar esta actividad con éxito, el estudiante deberá conocer los artefactos necesarios (Tema 1), haber entendido los ejercicios propuestos para esta fase, y realizado los laboratorios correspondientes (StarUML). El entregable de esta actividad será la captura de requisitos del proyecto en UML utilizando la herramienta StarUML.

Diseño del proyecto. Para desarrollar esta actividad, el estudiante deberá conocer los artefactos, patrones de diseño y las arquitecturas de varios niveles (Tema 2), haber entendido y desarrollado los ejemplos planteados y haber desarrollado los laboratorios propuestos. El entregable de esta actividad será el diseño del proyecto en UML utilizando la herramienta StarUML.

Implementación del proyecto. Antes de implementar en proyecto el alumno deberá conocer ciertas tecnologías. Esto conlleva el estudio de los contenidos teóricos (Tema 3) y el desarrollo de los laboratorios correspondientes. En entregable final será la implementación del proyecto.

Cronograma

ACTIVIDADES	TAREAS DIRIGIDAS	hrs	DESARROLLO DEL PROYECTO	hrs
INTRODUCCION	Apuntes	2	Lectura del proyecto	
	Lab 1	2		
CAPTURA DE REQUISITOS	Apuntes	2	Obtención de los Casos de Uso, Modelo del Dominio, Flujo de Eventos	5
	Lab 2	3		
	Resolución Ejercicios	4		
DISEÑO	Apuntes	4	Obtención de los Diagramas de secuencia y diagrama de Clases	8
	Labs 3 y 4	4		
	Resolución	4		
	Ejercicios	4		
IMPLEMENTACIÓN	Apuntes	6	Desarrollo del código	12
	Labs 5, 6 y 7	6		