

laboratorio

idtv

Propuesta de solución
formativa

Curso de UML



Universidad de Vigo

OBJETIVOS DEL CURSO

El objetivo de este curso es aportar al alumno una visión amplia y generalista del lenguaje UML (*Unified Modeling Language*). Este lenguaje de modelado de sistemas software ha cobrado gran importancia en los últimos años y, actualmente, se ha convertido en el estándar *de facto* en este ámbito, siendo apoyado, además, por el OMG (*Object Management Group*).

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Este estándar especifica cómo describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

El conocimiento sobre UML revierte directamente en la adquisición de conocimiento en el campo de la Ingeniería del Software. Ya que se aprenden los conceptos básicos involucrados en la aplicación de un método sistemático, estructurado y cuantificable para el diseño, desarrollo, operación y mantenimiento de productos software de calidad. Adicionalmente, la realización de prácticas sobre UML en un entorno real (trabajando con ordenadores personales) permite dotar al curso de un enfoque práctico fundamental para la adecuada adquisición del conocimiento impartido.

El objetivo de las prácticas (el 50% de la carga docente) es que los alumnos experimenten en primera persona la aplicación de UML en un modelo de proceso software iterativo e incremental. El alumno adquirirá los conocimientos y habilidades prácticas necesarios para utilizar tanto las técnicas de modelado UML, como las herramientas software de soporte, en el diseño y desarrollo de un producto software utilizando un paradigma orientado a objetos. Se pretende que las diferentes prácticas abarquen en su totalidad el diseño UML de un proyecto software emulando, lo más fielmente posible, el trabajo en un contexto profesional del sector: cubriendo desde la fase de captura de requisitos y la elaboración del plan de proyecto, hasta un diseño próximo a la implementación.

En consecuencia, al finalizar el curso, los alumnos habrán adquirido:

- las nociones fundamentales sobre las que se asienta el proceso de desarrollo de software, así como de las fases y actividades que comprende y las técnicas y notaciones de uso extendido para cada una de dichas actividades;
- conocimiento amplio sobre UML, sus notaciones y su aplicación al proceso software;
- destrezas prácticas de aplicación de los conocimientos anteriores en la elaboración de un caso de estudio que refleje lo más fielmente posible la organización del trabajo en el contexto profesional de la ingeniería del software.

PLAN FORMATIVO

El plan formativo en esta propuesta cubre los objetivos planteados estructurando la docencia teórica en un total de 10 módulos de dos horas y la docencia práctica en un total de 10 talleres prácticos, también de dos horas, en los que cada alumno abordará su propio caso de estudio para un posterior análisis y discusión conjunta. Las sesiones tal y como se recoge en la convocatoria serán de 8 horas. Se propone, en consecuencia, cubrir cada sesión diaria con dos módulos teóricos y dos talleres prácticos intercalados.

SESIÓN DÍA 1

MÓDULO 1. INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE

Desarrollo:

- Concepto de ingeniería del software y producto software.
- Características de un producto software de calidad.
- Concepto de proceso software.
- Modelos de ciclo de vida

MÓDULO 2. INTRODUCCIÓN A OO y UML

Desarrollo:

- Conceptos de Modelado.
- Crisis del software y paradigmas orientados a objetos (OO).
- Introducción a OO.
- Visión general del UML: arquitectura y estructura del lenguaje.
- Vistas UML del sistema: 4+1.

TALLER 1. UN CASO DE ESTUDIO CON UML

Desarrollo: El primero de los talleres prácticos del curso se centra en la presentación de un modelo de proceso software según el espíritu UML, así como en la presentación y discusión del caso de uso que se abordará de forma progresiva en el resto de talleres del curso.

TALLER 2. LA HERRAMIENTA Poseidon UML

Desarrollo: Una vez establecido el contexto en que el alumno va a trabajar, en el segundo de los talleres se presentará al alumno la herramienta que se utilizará durante el curso: *Poseidon for UML*.

SESIÓN DÍA 2

MÓDULO 3. MODELADO DE CASOS DE USO

Desarrollo:

- Casos de uso y actores: Identificación.
- Descripción de casos de uso: Flujos de eventos.
- Obtención y utilidad de los casos de uso.
 - Relaciones de extensión y de inclusión entre casos de uso.
 - Relaciones de herencia entre actores y caso de uso.
 - Modelo de Requisitos.

MÓDULO 4. MODELADO DE FLUJO DE TRABAJO: DIAGRAMA DE ACTIVIDADES

Desarrollo:

- Modelado de actividades
- Unión y división concurrente.
- Condiciones y decisiones.
- Uso de los diagramas de actividades en la captura de requisitos.

TALLER 3. OBTENCIÓN DEL MODELO DE CASOS DE USO

Desarrollo: En este taller el alumno se enfrentará a la especificación del caso de uso introducida en los talleres del día 1 para proceder a la identificación de actores y casos de uso.

TALLER 4. MODELO DE DOMINIO. DIAGRAMA DE ACTIVIDADES

Desarrollo: En este taller el alumno abordará la construcción de un modelo de actividades para el dominio, lo que permitirá completar el análisis inicial del sistema para el establecimiento de un plan iterativo de diseño. El diagrama de actividades describirá el flujo de trabajo en el dominio, es decir, los pasos que se siguen en el dominio para llevar a cabo las labores objetivo del sistema software.

SESIÓN DÍA 3

MÓDULO 5. MODELADO DE ESTRUCTURA ESTÁTICA

Desarrollo:

- Clases y objetos: representación.
- Relaciones entre clases: asociación y herencia.
- Diagramas de clases y objetos.
- Perspectivas de análisis y diseño.
- Diseño de la arquitectura.
- Usos de los diagramas de clases y objetos en el proceso software.

MÓDULO 6. ASPECTOS AVANZADOS DE LOS DIAGRAMAS DE CLASES

Desarrollo:

- Aspectos avanzados de las clases
- Otras relaciones entre clases: dependencia, agregación y composición.
- Relaciones de realización e interfaces.
- Conceptos avanzados de herencia.
- Paquete: organización del modelado.

TALLER 5. MODELO DE DOMINIO. DIAGRAMA DE CLASES DE ANÁLISIS

Desarrollo: En este taller el alumno abordará la obtención de un diagrama de clases desde una perspectiva conceptual que capture de forma adecuada el lenguaje de dominio. El objetivo es establecer los conceptos que utilizan los expertos del dominio así como la vinculación de conceptos entre sí.

TALLER 6. DIAGRAMA DE OBJETOS y PLAN DEL PROCESO

Desarrollo: En este taller el alumno abordará el desarrollo de un conjunto de diagramas de objetos que documente el análisis o el diseño desde un punto de vista de instancia. Al finalizar este taller se habrá construido una base común del caso de estudio sobre la que se sustentarán las distintas iteraciones que construyen el producto de forma incremental. En base a esta información (casos de uso y modelo de dominio), se elaborará un plan del proceso software que contenga un ordenamiento temporal de las iteraciones.

SESIÓN DÍA 4

MÓDULO 7. MODELADO DE INTERACCIONES

Desarrollo:

- Concepto de interacción
- Tipos de mensajes y flujos de control.
- Organización temporal: Diagramas de secuencia
- Organización estructural: Diagramas de colaboración.
- Comportamiento condicional, iterativo y recursivo.
- Creación y destrucción de instancias.
- Usos en el proceso software.

MÓDULO 8. DIAGRAMAS DE ESTADOS

Desarrollo:

- Concepto de diagrama de estados
- Mensajes, acciones y guardas
- Modelado de flujo de datos.
- Diagramas de estados jerárquicos.
- Diagramas de actividades: aspectos avanzados.
- Usos de diagramas de estados y de actividades en el proceso software.

TALLER 7. DISEÑO DE UNA ITERACIÓN. DIAGRAMAS DE SECUENCIA

Desarrollo: En este taller se abordará el análisis y diseño del caso de uso simulando una iteración del proceso software. El alumno comenzará detallando la especificación del caso de uso de la iteración y traduciendo dicha especificación detallada en un conjunto de diagramas de secuencia que representen los escenarios representativos del caso de uso. A partir de la información recabada en los diagramas de secuencia, posiblemente utilizando diagramas de colaboración como paso intermedio, el alumno actualizará el modelo de análisis (modelo de dominio) en un diagrama de clases de diseño.

TALLER 8. DISEÑO DE UNA ITERACIÓN. DIAGRAMAS DE ESTADO

Desarrollo: En este taller, y siguiendo con el diseño de la iteración que se ha abordado en el taller anterior, el alumno, guiado por el profesor, seleccionará una clase relevante en el diagrama de clases de diseño, y elaborará un diagrama de estados para dicha clase.

SESIÓN DÍA 5

MÓDULO 9. DIAGRAMAS DE IMPLEMENTACIÓN Y GENERACIÓN DE CÓDIGO

Desarrollo:

- Diagramas de componentes.
- Diagramas de despliegue.
- Paquetes, subsistemas y modelos.
- OOCASE y generación de código.

MÓDULO 10. MECANISMOS DE EXTENSIÓN y CONCLUSIONES

Desarrollo

- Bases arquitectónicas para la extensión UML.

- Estereotipos, valores etiquetados, restricciones.
- Definición de perfiles.
- Reingeniería con UML.
- Soporte para la reutilización: patrones.
- Críticas a UML. Conclusiones.

TALLER 9. GENERACIÓN DE CÓDIGO E IMPLEMENTACIÓN DEL SISTEMA

Desarrollo: En este taller los alumnos abordarán los aspectos del proceso software más cercano a la implementación del sistema sin entrar a enfrentarse a tareas de programación.

TALLER 10. UML PARA JAVA e INTERNET

Descripción y objetivos: En este taller que concluye el curso, el objetivo es permitir al alumno establecer una conexión práctica con aplicaciones reales. El taller se desarrollará a través de la inspección guiada de un caso de estudio complejo proporcionado como material.