

## Object-Oriented Analysis and Design Using UML

**Duration:** 5 Days

### What you will learn

El curso Análisis y Diseño Orientados a Objetos mediante UML (Unified Modeling Language) proporciona formación y experiencia práctica, centrándose en el uso eficaz de las tecnologías orientadas a objetos y un uso juicioso del modelado de software en su aplicación al proceso de desarrollo de software. Este curso con instructor utiliza clases teóricas, discusiones en grupo y actividades guiadas para presentar una guía básica de análisis y diseño orientados a objetos (OOAD) práctica y completa que abarca desde la recopilación de requisitos hasta el diseño del sistema. El curso proporciona un acercamiento pragmático al desarrollo de software orientado a objetos (OO) siguiendo patrones, principios y tecnologías OO demostrados y aplicables a los lenguajes OO, como el lenguaje de programación Java (TM).

Los alumnos disfrutarán de las ventajas de utilizar el lenguaje de modelado gráfico con amplia implantación, el lenguaje unificado de modelado (UML) versión 2.2, que les permitirá comunicar conceptos y decisiones, entender el problema y la solución propuesta, así como gestionar la complejidad de los artefactos describiendo el problema y la solución propuesta. El curso está estructurado de tal manera que se sigue una forma genérica del proceso de desarrollo de software que se centra en los aspectos de análisis y diseño aplicables a un proyecto de software OO. Este proceso genérico se puede adaptar fácilmente a los procesos concretos, que se tratarán posteriormente en el curso. El curso también permite comprender los patrones y marcos que pueden permitir la creación de componentes de software más flexibles y reutilizables.

Alumnos para los que se ha diseñado el curso:

Arquitectos de sistemas, ingenieros de software, analistas de sistemas y diseñadores responsables de la concepción y creación de aplicaciones de software orientadas a objetos.

Los arquitectos responsables de la concepción y creación de aplicaciones de software orientado a objetos también se pueden beneficiar de este curso.

Learn To:

Utilizar tecnologías orientadas a objetos

Utilizar el lenguaje unificado de modelado 2.2

Realizar análisis y diseño orientados a objetos

Seguir un proceso de desarrollo de software utilizando un proyecto de software OO

### Related Training

#### *Required Prerequisites*

Comprender los aspectos básicos del proceso de desarrollo de sistemas

Comprender los conceptos y la metodología orientados a objetos

Demostrar un conocimiento general sobre programación, preferiblemente utilizando el lenguaje de programación Java

## Course Objectives

Describe the object-oriented software development process

including object-oriented methodologies and workflows

Gather system requirements through interviews with stakeholders

Analyze system requirements to determine the use cases and domain model of the problem domain (the Requirements model)

Create a system architecture (the Architecture model) supporting the nonfunctional requirements (NFRs) and development constraints

Create a system design (the Solution model) supporting the functional requirements (FRs)

## Course Topics

### **Examen de Terminología y Conceptos Orientados a Objetos**

Describir los conceptos orientados a objetos (OO) importantes

Describir la terminología OO fundamental

### **Introducción al Modelado y al Proceso de Desarrollo de Software**

Describir el proceso de desarrollo de software orientado a objetos (OOSD)

Describir la forma en que el modelado soporta el proceso OOSD

Describir las ventajas del modelado de software

Explicar la finalidad, actividades y artefactos de los siguientes flujos de trabajo OOSD (disciplinas): recopilación de requi

### **Creación de Diagramas de Casos de Uso**

Justificar la necesidad de utilizar un diagrama de casos de uso

Identificar y describir los elementos esenciales de un diagrama de casos de uso UML

Desarrollar un diagrama de casos de uso para un sistema de software basado en los objetivos del propietario del negocio

Desarrollar elaborados diagramas de casos de uso basados en los objetivos de todos los participantes

Reconocer y documentar las dependencias de casos de uso utilizando la notación UML para ampliaciones, inclusiones y

Describir la forma de gestionar la complejidad de los diagramas de casos de uso mediante la creación de vistas UML en

### **Creación de Escenarios y Formularios de Casos de Uso**

Identificar y documentar escenarios para un caso de uso

Crear un formulario de caso de uso que describa el resumen de los escenarios en los flujos principal y alternativo

Describir la forma de hacer referencia a casos de uso incluidos y ampliables

Identificar y documentar requisitos no funcionales (NFR), reglas de negocio, riesgos y prioridades para un caso de uso  
Identificar la finalidad de un documento de especificaciones complementarias

### **Creación de Diagramas de Actividades**

Identificar los elementos esenciales de un diagrama de actividades  
Modelar un flujo de casos de uso de eventos utilizando un diagrama de actividades

### **Determinación de las Abstracciones Clave**

Identificar un juego de abstracciones clave candidatas  
Identificar las abstracciones clave mediante el análisis CRC

### **Construcción del Modelo de Dominio del Problema**

Identificar los elementos esenciales de un diagrama de clases UML  
Construir un modelo de dominio utilizando un diagrama de clases  
Identificar los elementos esenciales de un diagrama de objetos UML  
Validar el modelo de dominio con uno o varios diagramas de objetos

### **Transición del Análisis al Diseño mediante Diagramas de Interacción**

Explicar el propósito y los elementos del modelo de diseño  
Identificar los elementos esenciales de un diagrama de comunicación UML  
Crear una vista de diagrama de comunicación del modelo de diseño  
Identificar los elementos esenciales de un diagrama de secuencias UML  
Crear una vista de diagrama de secuencias del modelo de diseño

### **Modelado del Estado del Objeto con Diagramas de Máquina de Estados**

Modelar el estado del objeto  
Describir los elementos esenciales de un diagrama de máquina de estados UML

### **Aplicación de Patrones de Diseño al Modelo de Diseño**

Definir los elementos esenciales de un patrón de software  
Describir el patrón compuesto  
Describir el patrón de estrategia  
Describir el patrón observador  
Describir el patrón de fábrica abstracto

### **Introducción a los Diagramas y Conceptos Arquitectónicos**

Distinguir entre arquitectura y diseño  
Describir niveles, capas y cualidades sistémicas  
Describir el flujo de trabajo de arquitectura  
Describir los diagramas de las vistas de arquitectura clave  
Seleccionar el tipo de arquitectura  
Crear los artefactos del flujo de trabajo de arquitectura

### **Introducción a los Niveles Arquitectónicos**

Describir los conceptos de los niveles de cliente y de presentación  
Describir los conceptos del nivel empresarial  
Describir los conceptos de los niveles de recursos e integración  
Describir los conceptos del modelo de solución

### **Acotación del Modelo de Diseño de Clases**

Acotar los atributos del modelo de dominio  
Acotar las relaciones del modelo de dominio

- Acotar los métodos del modelo de dominio
- Acotar los constructores del modelo de dominio
- Anotar el comportamiento del método
- Crear componentes con interfaces

### **Visión General Sobre los Procesos de Desarrollo de Software**

- Explicar las prácticas recomendadas de las metodologías OOSD
- Describir las funciones de varias metodologías comunes
- Elegir la metodología que mejor se adapte al proyecto
- Desarrollar un plan de iteración

### **Visión General de los Marcos**

- Definir un marco
- Describir las ventajas y desventajas del uso de marcos
- Identificar varios marcos comunes
- Comprender el concepto de creación de sus propios marcos de dominio de negocio

### **Revisión del Curso**

- Revisar las funciones clave de la orientación a objetos
- Revisar los principales diagramas UML
- Revisar los flujos de trabajo de análisis de requisitos (análisis) y diseño