

Análisis y Diseño de Software.

(72 Hrs)

Objetivo del curso:

- Aplicar las técnicas de Análisis y Diseño Orientado a Objetos en el proceso de desarrollo de software.
- Especificación de los componentes de los modelos de análisis y diseño utilizando UML.

Programa de estudios.

Unidad 1. Conceptos básicos. (10 Hrs).

Objetivo:

Reforzar la comprensión de tópicos y conceptos relacionados con el proceso de desarrollo del software (Ciclo de vida del software) haciendo énfasis en lo referente al análisis y diseño del software.

- 1.1. Importancia del Análisis y el Diseño.
 - 1.1.1 Fundamentación.
 - 1.1.2 Objetivos del análisis.
 - 1.1.3 Objetivos del diseño.
- 1.2. El análisis y el diseño en el contexto de las metodologías de la Ingeniería de Software.
 - 1.3.1 El ciclo de vida del desarrollo de software.
 - 1.3.2 El enfoque en cascada.
 - 1.3.2 El enfoque iterativo e incremental.
 - 1.3.3 El enfoque en los métodos ágiles.
- 1.3. Técnicas y herramientas.
 - 1.4.1 Uso de patrones para análisis y diseño.
 - 1.4.2 Herramientas CASE.

Unidad 2. El proceso de análisis. (25 Hrs.)

Objetivo:

Conocer y aplicar técnicas de análisis OO de software para la construcción del modelo de análisis.

- 2.1 La importancia del Análisis.
- 2.2 El modelo de análisis.
 - 2.2.1 Modelado del dominio.
 - 2.2.1.1 Objetivo.
 - 2.2.1.2. Clases conceptuales, atributos y relaciones.
 - 2.2.1.3. Diagramas UML.
 - 2.2.1.3.1. El diagrama de clases durante el análisis.
 - 2.2.1.3.2. El diagrama de actividad durante el análisis.
 - 2.2.1.3.3. El diagrama de máquina de estados.
 - 2.2.1.4. Técnicas de modelado.

2.2.2 Glosario de términos.

2.3 La gestión de los requisitos.

2.3.1 Requisitos funcionales y no funcionales.

2.3.2 Descubrimiento.

2.3.3 Especificación.

2.3.3.1 Casos de uso.

2.3.4.1.1 Conceptos.

2.3.4.1.2 Formatos de escritura.

2.3.4.1.3 Diagrama de casos de uso.

2.3.3.2 Las historias de uso (historias de usuario).

2.3.4 Operaciones del sistema.

2.3.4.1 Contratos.

2.3.5 Uso de prototipos.

2.3.6 Priorización de los requisitos.

2.4 Prototipo de la interfaz del usuario.

Unidad 3. **El proceso de diseño.** (30 Hrs.)

Objetivo:

Conocer y aplicar técnicas básicas de diseño OO de software para la creación de los productos del diseño y expresarlos utilizando el lenguaje de modelado UML.

3.1 Objetivos del diseño.

3.2 Transición del análisis al diseño.

3.2.1 El diseño dirigido por las responsabilidades.

3.2.1 El concepto de responsabilidad.

3.2.2 Diseño de las colaboraciones.

3.3 Perspectivas estáticas y dinámicas del diseño.

3.3.1 Modelado estático.

3.3.1.1 Clases de diseño y clases de software.

3.3.1.2 Agrupamientos de clases.

3.3.1.2.1 Modelado de paquetes.

3.3.1.2.1.1 El diagrama de paquetes de UML.

3.3.1.2.2 Patrones para la cohesión y dependencia de paquetes.

3.3.2 Modelado dinámico.

3.3.2.1 Diseño de colaboraciones.

3.3.2.1.1 Asignación de responsabilidades.

3.3.2.1.2 Patrones GRASP.

3.3.2.1.3 Las tarjetas CRC.

3.3.2.1.4 Definición de las relaciones entre clases.

3.3.2.2 UML para el diseño.

3.3.2.2.1 Diagramas de secuencia y colaboración.

3.3.2.2.2 Diagrama de máquina de estados.

3.3.2.2.3 Diagrama de actividad.

3.4.2.3 Principios SOLID de diseño .

3.4.2.4 Patrones de diseño (Gang of 4).

3.4 Modelado de la arquitectura del software.

3.4.1 Conceptos sobre arquitectura del software.

3.4.2 Modelos y estilos arquitectónicos.

3.4.3 UML para el modelado de la arquitectura.

3.4.3.1 El diagrama de paquetes.

3.4.3.2 El Diagrama de componentes.

Unidad 4. **Diseño de la fuente de datos.** (5 Hrs.).

Objetivo.

Aplicar la práctica de desarrollo iterativo e incremental al modelo relacional de la fuente de datos.

4.1 Clases, relaciones y persistencia.

4.2 UML para el diseño de la fuente de datos.

4.3 Generación de scripts para la creación de la fuente de datos.

4.4 Refactorización de la fuente de datos.

Unidad 5. **Proyecto final.** (2 Hrs).

Objetivo.

Mostrar el dominio de las técnicas adquiridas durante el curso en un proyecto diseñado con dicho objetivo.

5.1 Asignación del proyecto.

5.2 Programación de revisiones de avances.

Bibliografía:

1. Applying UML and Patterns An Introduction to O O Analysis and Design and the Unified Process 2nd Ed - Larman - PH PTR 2002.
2. Software Requirements (3rd Edition) Karl Weigers, Joy Beatty. Microsoft Press.
3. Mastering the Requirements Process 2nd Edition - Robertson,Robertson - Addison Wesley(2006)
4. Extreme Programming Explained: Embrace Change, Second Edition, Kent Beck, Cynthia Andres, Addison Wesley Professional (2004)
5. Análisis y Diseño OO de Sistemas usando UML, Bennet, McRobb, Farmer, 3^a. Edición, McGraw Hill.