

Universidad Autónoma de Sinaloa

Facultad de Informática Culiacán  
Licenciatura en Informática

Análisis y Diseño de Software

# Objetivos del curso:

- ✓- Aplicar las técnicas de Análisis y Diseño Orientado a Objetos en el proceso de desarrollo de software.
- ✓ Construir los diagramas de los modelos de análisis y diseño utilizando el lenguaje UML.
- ✓ Utilizar una herramienta software que apoye el proceso de modelado.
- ✓ Realizar las actividades de análisis y diseño del software en el marco de una metodología de Ingeniería de Software.

# Prerrequisitos

- **Análisis y diseño orientado a objetos**. Comprender y aplicar los conceptos del análisis y diseño bajo el paradigma de la orientación a objetos.
- **Lenguajes de programación**. Conocer las características deseables de un lenguaje de programación de computadoras para expresar convenientemente un diseño OO .
- **Ingeniería de software**. Conocer las características de las metodologías de uso más común para el manejo del ciclo de vida de desarrollo de software.
- **Fundamentos de base de datos**. Justificar la utilización de una base de datos para realizar la persistencia de datos y comprender la problemática de la transformación objeto-relacional.

## Contenido temático:

- Unidad 1. Conceptos básicos.
- Unidad 2. El modelo de análisis.
- Unidad 3. El modelo de diseño.
- Unidad 4. Nuevos paradigmas.
- Unidad 5. Proyecto final.
- Bibliografía.
- Procedimiento de evaluación del curso.

# Unidad 1. Conceptos básicos.

## 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 Modelo de análisis.

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 El diagrama de clases durante el análisis.

2.2.1.4. El diagrama de actividad durante el análisis.

2.2.1.5. El diagrama de máquina de estados.

2.2.1.6. Técnicas de modelado.

2.2.2 Glosario de términos.

# Unidad 2. El modelo de análisis. ( continuación )

## 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 ).

2.3.3.3 Operaciones del sistema ( Requisitos del software ).

2.3.3.3.1 Contratos.

2.3.4 Uso de prototipos.

2.3.5 Priorización de los requisitos.

## 2.4 Prototipo de la interfaz del usuario.

# Unidad 3. El modelo de diseño.

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.

# Unidad 3. El modelo de diseño. ( Continuación )

## 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 3. El modelo de diseño. ( Continuación )

3.6 Descubrimiento e implementación de la herencia.

3.7 Definición e implementación de interfaces.

3.8 Calidad en el diseño.

3.8.1 Bajo acoplamiento.

3.8.2 Alta cohesión.

3.8.3 Patrones para la refactorización.

# Unidad 5. Proyecto final.

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. Mastering the Requirements Process 2nd Edition - Robertson, Robertson - Addison Wesley(2006).
3. Extreme Programming Explained: Embrace Change, Second Edition, Kent Beck, Cynthia Andres, Addison Wesley Professional ( 2004 ).
4. Análisis y Diseño OO de Sistemas usando UML, Bennet, McRobb, Farmer, 3ª. Edición, McGraw Hill.
5. UML . Gota a Gota.

# Procedimiento de evaluación.

El curso será evaluado en la escala del 1 al 10, siendo 6 la calificación mínima aprobatoria.

La calificación final estará integrada de la forma siguiente:

- 6 puntos que se obtendrán mediante la aplicación de 3 exámenes parciales cubriendo el contenido teórico del curso ( el contenido a evaluar por examen se definirá con 1 semana de anticipación a la fecha de aplicación. ).
- 2 puntos que se obtendrán mediante dos trabajos de investigación que serán expuestos en el grupo y desarrollados por equipos.
- 2 puntos mediante el proyecto final evaluado al final del curso.

## **Importante:**

Para tener derecho a calificación final deberán cubrirse los requisitos siguientes:

- Asistencia mínima al curso del 75% de las horas que lo componen.
- Lograr un promedio mínimo de 6 en los 3 exámenes parciales, y haber aprobado al menos 2.
- Presentar en la fecha que se defina, y aprobar el proyecto final.

## **Examen ordinario.**

Se aplicarán de nueva cuenta los 3 exámenes parciales en caso de no lograr el promedio de 6 ( únicamente se volverán a aplicar los exámenes que se hayan reprobado. ).

## **Examen extraordinario.**

El mismo criterio del examen ordinario. En el caso de no haber terminado el proyecto a la fecha del examen ordinario, tendrán que presentarlo obligatoriamente a la fecha del extraordinario.