

# UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

## COORDINACIÓN GENERAL DE FORMACIÓN PROFESIONAL

### PROGRAMA DE UNIDAD DE APRENDIZAJE

#### I. DATOS DE IDENTIFICACIÓN

- 1. Unidad Académica:** Facultad de Ingeniería, Arquitectura y Diseño, Ensenada; Facultad Ciencias Químicas e Ingeniería, Tijuana; y Facultad de Ciencias de la Ingeniería y Tecnología, Valle de las Palmas
- 2. Programa Educativo:** Ingeniero en Software y Tecnologías Emergentes
- 3. Plan de Estudios:** 2022-1
- 4. Nombre de la Unidad de Aprendizaje:** Análisis y Diseño de Software
- 5. Clave:** 40015
- 6. HC:** 02 **HT:** 01 **HL:** 03 **HPC:** 00 **HCL:** 00 **HE:** 02 **CR:** 08
- 7. Etapa de Formación a la que Pertenece:** Disciplinaria
- 8. Carácter de la Unidad de Aprendizaje:** Obligatoria
- 9. Requisitos para Cursar la Unidad de Aprendizaje:** Ninguno



#### Equipo de diseño de PUA

Manuel Castañón Puga.  
Christian Xavier Navarro Cota.  
Héctor Zatarain Aceves  
Thelma Violeta Ocegueda Miramontes

#### Vo.Bo. de subdirector(es) de Unidad(es) Académica(s)

Humberto Cervantes De Ávila  
Daniela Mercedes Martínez Platas  
Noemí Hernández Hernández

**Fecha:** 20 de febrero de 2021

## **II. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE**

La unidad de aprendizaje proporciona los fundamentos para el análisis y diseño del Software; lo que permite aplicar las técnicas de modelado para la definición de la arquitectura, componentes, e interfaces apropiadas al tipo de Software que se desea construir utilizando el Lenguaje de Modelado Unificado (UML).

Se ubica en la etapa disciplinaria, con carácter de obligatoria y forma parte del área de Métodos y Tecnologías de Software del programa educativo Ingeniero en Software y Tecnologías Emergentes.

## **III. COMPETENCIA GENERAL DE LA UNIDAD DE APRENDIZAJE**

Diseñar software, basado en los requerimientos establecidos y la aplicación de técnicas de modelado, para definir su arquitectura, componentes, interfaces y otras características, con una actitud creativa, organizada y con habilidad de interpretación.

## **IV. EVIDENCIA(S) DE APRENDIZAJE**

Reporte técnico del análisis y diseño de un sistema de software que especifique la arquitectura, componentes e interfaces externas e internas de acuerdo a los requerimientos de Software, y considerando las diferentes técnicas, herramientas y metodologías existentes.

**V. DESARROLLO POR UNIDADES**  
**UNIDAD I. Lenguaje Unificado de Modelado (UML)**

**Competencia:**

Aplicar el Lenguaje Unificado de Modelado (UML), a través de la identificación de sus elementos, para documentar el análisis y diseño de un sistema de software, con actitud analítica, sistemática y creativa.

**Contenido:**

**Duración:** 10 horas

- 1.1. El Lenguaje Unificado de Modelado (UML)
- 1.2. UML en las diferentes etapas del proceso de desarrollo
  - 1.2.1 UML en el documento de Especificación de Requerimientos
  - 1.2.2 UML en el documento de diseño
  - 1.2.3 UML en la construcción
- 1.3. Los diagramas UML
  - 1.3.1 Diagramas estructurales
  - 1.3.2 Diagramas de comportamiento
  - 1.3.3 Diagramas de Interacción
  - 1.3.4 Diagramas de Implementación
- 1.4. La adopción de UML en la ingeniería de software

## UNIDAD II. Modelado y análisis de software

### Competencia:

Distinguir los conceptos principales del modelado y análisis, a través de la identificación de los diferentes tipos de modelos, para examinar los requerimientos de software, con pensamiento objetivo, metódico y actitud colaborativa.

### Contenido:

**Duración:** 10 horas

#### 2.1. Fundamentos de modelado

2.1.1 Principios de modelado (por ejemplo, descomposición, abstracción, generalización, proyección/vistas y uso de enfoques formales).

2.1.2. Precondiciones, postcondiciones, invariantes y diseño por contrato

2.1.3. Introducción a los modelos matemáticos y la notación formal

#### 2.2 Tipos de modelos

2.2.1 Modelado de información (por ejemplo, modelado de relación de entidad y diagramas de clase)

2.2.2 Modelos de comportamiento (por ejemplo, diagramas de estado, análisis de casos de uso, diagramas de interacción, análisis de efectos y modos de falla, y análisis de árbol de fallas)

2.2.3 Modelado arquitectónico (por ejemplo, patrones arquitectónicos y diagramas de componentes)

2.2.4 Modelado de dominios (por ejemplo, enfoques de ingeniería de dominios)

2.2.5 Modelado empresarial (por ejemplo, procesos de negocios, organizaciones, objetivos y flujo de trabajo)

2.2.6 Modelado de sistemas embebidos (por ejemplo, análisis de calendarización en tiempo real y protocolos de interfaz)

#### 2.3 Fundamentos de análisis

2.3.1 Análisis de la forma (por ejemplo, integridad, consistencia y robustez)

2.3.2 Análisis de la exactitud (por ejemplo, análisis estático, simulación y verificación de modelos)

2.3.3 Análisis de confiabilidad (por ejemplo, análisis de modo de falla y árboles de fallas)

2.3.4 Análisis formal (por ejemplo, prueba de teoremas)

## UNIDAD III. Diseño de software

### Competencia:

Aplicar los conceptos de diseño, a través de la selección de estrategias adecuadas para definir una arquitectura, componentes, interfaces y otras características de un sistema de software, con actitud crítica, creativa y de compañerismo.

### Contenido:

**Duración:** 12 horas



### 3.1. Conceptos de diseño.

- 3.1.1. Definición de diseño.
- 3.1.2. Cuestiones de diseño fundamentales (por ejemplo, datos persistentes, administración de almacenamiento y excepciones)
- 3.1.3. Contexto de diseño dentro de múltiples ciclos de vida de desarrollo de software.
- 3.1.4. Principios de diseño (ocultación de información, cohesión y acoplamiento).
- 3.1.5. Interacciones entre diseño y requisitos.
- 3.1.6. Diseño para atributos de calidad (por ejemplo, confiabilidad, facilidad de uso, facilidad de mantenimiento, rendimiento, capacidad de prueba, seguridad y tolerancia a fallas).

### 3.2. Estrategias de diseño.

- 3.2.1. Concesiones (trade-offs) en diseño.
- 3.2.2. Diseño orientado a funciones.
- 3.2.3. Diseño orientado a objetos.
- 3.2.4. Diseño centrado en la estructura de datos.
- 3.2.5. Diseño orientado a aspectos.

### 3.3. Diseño arquitectónico.

- 3.3.1. Estilos arquitectónicos, patrones y marcos de trabajo.
- 3.3.2. Concesiones (trade-offs) arquitectónicas entre varios atributos.
- 3.3.3. Consideraciones de ingeniería de hardware y sistemas en la arquitectura de software.
- 3.3.3. Requisitos de trazabilidad en la arquitectura.

3.3.4. Arquitecturas orientadas a servicios

3.3.5. Arquitecturas para redes, móviles y sistemas embebidos.

3.3.6. Relación entre la arquitectura del producto y la estructura de la organización de desarrollo y el mercado.

3.4. Evaluación del diseño

3.4.1. Atributos de diseño (por ejemplo, acoplamiento, cohesión, ocultación de la información y separación de intereses)

3.4.2. Métricas de diseño

3.4.3. Análisis de diseño formal

## VI. ESTRUCTURA DE LAS PRÁCTICAS DE TALLER

No.	Nombre de la Práctica	Procedimiento	Recursos de Apoyo	Duración
<b>UNIDAD I</b>				
1	<b>Diagramas estructurales</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> <li>2. Crea un instancia de cada tipo diagrama estructural</li> <li>3. Agrega elementos a los diagramas.</li> <li>4. Conecta los elementos de los diagramas.</li> <li>5. Documenta los elementos de los diagramas.</li> <li>6. Interpreta los diagramas.</li> <li>7. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> <li>● Internet</li> </ul>	1 hora
2	<b>Diagramas de comportamiento</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> <li>2. Crea una instancia de cada tipo diagrama de comportamiento.</li> <li>3. Agrega elementos a los diagramas.</li> <li>4. Conecta los elementos de los diagramas.</li> <li>5. Documenta los elementos de los diagramas.</li> <li>6. Interpreta los diagramas.</li> <li>7. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> <li>● Internet</li> </ul>	1 hora
3	<b>Diagramas de Interacción</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> </ul>	1 hora

		<ol style="list-style-type: none"> <li>2. Crea una instancia de cada tipo diagrama de interacción.</li> <li>3. Agrega elementos a los diagramas.</li> <li>4. Conecta los elementos de los diagramas.</li> <li>5. Documenta los elementos de los diagramas.</li> <li>6. Interpreta los diagramas.</li> <li>7. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Internet</li> </ul>	
4	<b>Diagramas de Implementación</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad</li> <li>2. Crea una instancia de cada tipo diagrama de implementación.</li> <li>3. Agrega elementos a los diagramas.</li> <li>4. Conecta los elementos de los diagramas.</li> <li>5. Documenta los elementos de los diagramas.</li> <li>6. Interpreta los diagramas.</li> <li>7. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> <li>• Internet</li> </ul>	1 hora
<b>UNIDAD II</b>				
5	<b>Modelos de información</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad</li> <li>2. Analiza los requerimientos proporcionados por el docente.</li> <li>3. Diseña el modelo de información con base en los requerimientos.</li> <li>4. Entrega reporte de la actividad al docente para su</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> <li>• Internet</li> </ul>	1 hora

		retroalimentación.		
6	<b>Modelo de comportamiento</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> <li>2. Analiza los requerimientos proporcionados por el docente.</li> <li>3. Diseña el modelo de comportamiento con base en los requerimientos.</li> <li>4. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> <li>• Internet</li> </ul>	1 hora
7	<b>Modelo arquitectónico</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> <li>2. Analiza los requerimientos proporcionados por el docente.</li> <li>3. Diseña el modelo arquitectónico con base en los requerimientos.</li> <li>4. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> <li>• Internet</li> </ul>	1 hora
8	<b>Modelo de dominio</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> <li>2. Analiza los requerimientos proporcionados por el docente.</li> <li>3. Diseña el modelo de dominio con base en los requerimientos.</li> <li>4. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> <li>• Internet</li> </ul>	1 hora
9	<b>Modelo de sistemas embebidos</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> </ul>	1 hora

		<ol style="list-style-type: none"> <li>2. Analiza los requerimientos proporcionados por el docente.</li> <li>3. Diseña el modelo de sistemas embebidos con base en los requerimientos.</li> <li>4. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Internet</li> </ul>	
10	<b>Análisis estático de software</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> <li>2. A partir de un conjunto de código fuente proporcionado por el docente, hacer ingeniería de reversa para generar un modelo estático del sistema.</li> <li>3. Analiza con el docente el modelo estático obtenido para la comprensión del programa o revisión del código.</li> <li>4. Entrega reporte del diseño al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para ingeniería de reversa y modelado de UML</li> <li>• Internet</li> </ul>	1 hora
<b>UNIDAD III</b>				1 hora
11	<b>Diseño orientado a funciones</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> <li>2. Analiza los modelos proporcionados por el docente.</li> <li>3. Define el diseño orientado a funciones con base en los modelos.</li> <li>4. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> </ul>	1 hora

12	<b>Diseño orientado a objetos</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> <li>2. Analiza los modelos proporcionados por el docente.</li> <li>3. Define el diseño orientado a objetos con base en los modelos.</li> <li>4. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> </ul>	1 hora
13	<b>Diseño centrado en la estructura de datos</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> <li>2. Analiza los modelos proporcionados por el docente.</li> <li>3. Define el diseño centrado en la estructura de datos con base en los modelos.</li> <li>4. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> </ul>	1 hora
14	<b>Diseño orientado a aspectos</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> <li>2. Analiza los modelos proporcionados por el docente.</li> <li>3. Define el diseño orientado a aspectos con base en los modelos.</li> <li>4. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora.</li> <li>● Herramienta CASE para modelado de UML.</li> </ul>	1 hora
15	<b>Diseño arquitectónico</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> </ul>	1 hora

		<ol style="list-style-type: none"> <li>2. Analiza los modelos proporcionados por el docente.</li> <li>3. Define el diseño orientado arquitectónico con base en los modelos.</li> <li>4. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>		
16	<b>Evaluación del diseño</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para participar en la actividad.</li> <li>2. Identifica las métricas de los diseños de software.</li> <li>3. Evalúa los diseños proporcionados por el docente.</li> <li>4. Entrega reporte de la actividad al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> </ul>	1 hora

## VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO

No.	Nombre de la Práctica	Procedimiento	Recursos de Apoyo	Duración
<b>UNIDAD I</b>				
1	<b>Diagramas estructurales</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Crea el tipo de diagrama estructural adecuado, basándose en los requerimientos del software del caso de estudio.</li> <li>3. Agrega elementos a los diagramas.</li> <li>4. Conecta los elementos de los diagramas.</li> <li>5. Documenta los elementos de los diagramas.</li> <li>6. Interpreta los diagramas.</li> <li>7. Entrega reporte de la práctica al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> <li>● Internet</li> </ul>	3 horas
2	<b>Diagramas de comportamiento</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Crea el tipo de diagrama de comportamiento adecuado, basándose en los requerimientos del software del caso de estudio.</li> <li>3. Agrega elementos a los diagramas.</li> <li>4. Conecta los elementos de los diagramas.</li> <li>5. Documenta los elementos de los diagramas.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> <li>● Internet</li> </ul>	3 horas

		<ol style="list-style-type: none"> <li>6. Interpreta los diagramas.</li> <li>7. Entrega reporte de la práctica al docente para su retroalimentación.</li> </ol>		
3	<b>Diagramas de Interacción</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Crea el tipo de diagrama de interacción adecuado, basándose en los requerimientos del software del caso de estudio.</li> <li>3. Agrega elementos a los diagramas.</li> <li>4. Conecta los elementos de los diagramas.</li> <li>5. Documenta los elementos de los diagramas.</li> <li>6. Interpreta los diagramas.</li> <li>7. Entrega reporte de la práctica al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> <li>● Internet</li> </ul>	3 horas
4	<b>Diagramas de Implementación</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Crea el tipo de diagrama de implementación adecuado, basándose en los requerimientos del software del caso de estudio.</li> <li>3. Agrega elementos a los diagramas.</li> <li>4. Conecta los elementos de los diagramas.</li> <li>5. Documenta los elementos de los diagramas.</li> <li>6. Interpreta los diagramas.</li> <li>7. Entrega reporte de la práctica al</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> <li>● Internet</li> </ul>	3 horas

		docente para su retroalimentación.		
<b>UNIDAD II</b>				
5	<b>Modelos de información</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Analiza los requerimientos del sistema de software.</li> <li>3. Diseña el modelo de información del sistema de software del caso de estudio</li> <li>4. Entrega reporte del diseño al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> <li>• Internet</li> </ul>	3 horas
6	<b>Modelo de comportamiento</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Analiza los requerimientos del sistema de software.</li> <li>3. Diseña el modelo de comportamiento del sistema de software requerimientos del caso de estudio</li> <li>4. Entrega reporte del diseño al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> <li>• Internet</li> </ul>	3 horas
7	<b>Modelo arquitectónico</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Analiza los requerimientos del sistema de software.</li> <li>3. Diseña el modelo arquitectónico del sistema de software requerimientos del caso de estudio.</li> <li>4. Entrega reporte del diseño al docente para su</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> <li>• Internet</li> </ul>	3 horas

		retroalimentación.		
8	<b>Modelo de dominio</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Analiza los requerimientos del sistema de software.</li> <li>3. Diseña el modelo de dominio del sistema de software del caso de estudio.</li> <li>4. Entrega reporte del diseño al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> <li>• Internet</li> </ul>	3 horas
9	<b>Modelo de sistemas embebidos</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Analiza los requerimientos del sistema de software.</li> <li>3. Diseña el modelo de sistemas embebidos del sistema de software.</li> <li>4. Entrega reporte del diseño al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> <li>• Internet</li> </ul>	3 horas
10	<b>Análisis estático de software</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. A partir de un conjunto de código fuente propuesto por el estudiante, hacer ingeniería de reversa para generar un modelo estático.</li> <li>3. Analiza el modelo estático obtenido para estudiar su incorporación al modelo del sistema de software del caso de estudio.</li> <li>4. Incorpora a conveniencia el modelo estático obtenido.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para ingeniería de reversa y modelado de UML</li> <li>• Internet</li> </ul>	3 horas

		5. Entrega reporte del diseño al docente para su retroalimentación.		
<b>UNIDAD III</b>				
11	<b>Diseño orientado a funciones</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Analiza los modelos del sistema de software del caso de estudio.</li> <li>3. Define el diseño orientado a funciones para el sistema de software.</li> <li>4. Entrega reporte del diseño al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> </ul>	3 horas
12	<b>Diseño orientado a objetos</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Analiza los modelos del sistema de software del caso de estudio.</li> <li>3. Define el diseño orientado a objetos para el sistema de software.</li> <li>4. Entrega reporte del diseño al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> </ul>	3 horas
13	<b>Diseño centrado en la estructura de datos</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Analiza los modelos del sistema de software del caso de estudio.</li> <li>3. Define el diseño centrado en la estructura de datos para el sistema de software.</li> <li>4. Entrega reporte del diseño al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>• Computadora</li> <li>• Herramienta CASE para modelado de UML</li> </ul>	3 horas

14	<b>Diseño orientado a aspectos</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Analiza los modelos del sistema de software del caso de estudio.</li> <li>3. Define el diseño orientado a aspectos para el sistema de software.</li> <li>4. Entrega reporte del diseño al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> </ul>	3 horas
15	<b>Diseño arquitectónico</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Analiza los modelos del sistema de software del caso de estudio.</li> <li>3. Define el diseño orientado arquitectónico para el sistema de software.</li> <li>4. Entrega reporte del diseño al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> </ul>	3 horas
16	<b>Evaluación del diseño</b>	<ol style="list-style-type: none"> <li>1. Atiende las orientaciones del profesor para elaborar la práctica.</li> <li>2. Analiza los diseños del sistema de software del caso de estudio.</li> <li>3. Identifica las métricas de los diseños de software.</li> <li>4. Evalúa los diseños del sistema de software.</li> <li>5. Entrega reporte de la evaluación de los diseños al docente para su retroalimentación.</li> </ol>	<ul style="list-style-type: none"> <li>● Computadora</li> <li>● Herramienta CASE para modelado de UML</li> </ul>	3 horas

## VII. MÉTODO DE TRABAJO

**Encuadre:** El primer día de clase el docente debe establecer la forma de trabajo, criterios de evaluación, calidad de los trabajos académicos, derechos y obligaciones docente-alumno.

### **Estrategia de enseñanza (docente):**

- Presenta información sobre los conceptos básicos
- Presenta ejercicios prácticos relacionados con las temáticas
- Dirige, supervisa y retroalimenta las prácticas
- Propicia la participación activa de los estudiantes
- Revisa y evalúa reportes de prácticas y actividades
- Elabora y aplica evaluaciones

### **Estrategia de aprendizaje (alumno):**

- Investiga y analiza información sobre conceptos básicos
- Resuelve ejercicios prácticos proporcionados por el profesor
- Realiza las prácticas
- Participa activamente en clase
- Elabora y entrega reportes de prácticas
- Trabaja en equipo
- Elabora y entrega actividades y prácticas en tiempo y forma
- Presenta evaluaciones

## VIII. CRITERIOS DE EVALUACIÓN

La evaluación será llevada a cabo de forma permanente durante el desarrollo de la unidad de aprendizaje de la siguiente manera:

### **Criterios de acreditación**

- Para tener derecho a examen ordinario y extraordinario, el estudiante debe cumplir con los porcentajes de asistencia que establece el Estatuto Escolar vigente.
- Calificación en escala del 0 al 100, con un mínimo aprobatorio de 60.

### **Criterios de evaluación**

- Evaluaciones parciales .....	10%
- Prácticas de laboratorio y taller .....	40%
- Reporte técnico del análisis y diseño de un sistema de software .....	50%
<b>Total</b> .....	<b>100%</b>

## IX. REFERENCIAS

Básicas	Complementarias
<p>Dennis, A., Wixom, B. H., y Roth, R. M. (2018). <i>Systems Analysis and Design</i>. Estados Unidos: Wiley.</p> <p>Dennis, A., Wixom, B. H., y Tegarden, D. (2015). <i>Systems analysis and design: An object-oriented approach with UML</i>. Estados Unidos: John Wiley &amp; Sons.</p> <p>Jiménez de Parga, C. (2015). <i>UML. Aplicaciones en Java Y C</i>. España: Ra-Ma.</p> <p>Rajaraman, V. (2018). <i>Analysis and design of information systems</i>. India. PHI Learning Pvt. Ltd.</p> <p>Tilley, S., y Rosenblatt H. J. (2016). <i>Systems Analysis and Design (Shelly Cashman Series)</i>. Estados Unidos: Cengage Learning</p>	<p>Balarin, F., Giusto, P., Jurecska, A., Chiodo, M., Hsieh, H., Passerone, C., y Suzuki, K. (1997). <i>Hardware-software co-design of embedded systems: the POLIS approach</i>. Estados Unidos: Springer Science &amp; Business Media. [clásica]</p> <p>Chapman, W. (2018). <i>Engineering modeling and design</i>. Routledge.</p> <p>Davis, W. S. y Yen, D. C. (2018). <i>The Information System Consultant's Handbook Systems Analysis and Design</i>. Estados Unidos, CRC press.</p> <p>DeMicheli, G. y Sami, M. G. (Eds.). (2013). <i>Hardware/software Co-design</i> (Vol. 310). Estados Unidos: Springer Science &amp; Business Media. [clásica]</p> <p>Martín, J. y López, L. (2014). <i>UML práctico: aprende UML paso a paso</i>. México, Edición Kindle.</p> <p>Miles, R., y Hamilton, K. (2008). <i>Learning UML 2.0</i>. Estados Unidos: O'Reilly Media, Inc. [clásica]</p> <p>Niemann, R. (1998). <i>Hardware/software co-design for data flow dominated embedded systems</i>. Estados Unidos: Springer Science &amp; Business Media. [clásica]</p> <p>Schaumont, P. R. (2012). <i>A practical introduction to hardware/software codesign</i>. Estados Unidos: Springer Science &amp; Business Media. [clásica]</p> <p>Staunstrup, J. y Wolf, W. (2013). <i>Hardware/software co-design: principles and practice</i>. Estados Unidos: Springer Science &amp; Business Media. [clásica]</p>

## **X. PERFIL DEL DOCENTE**

El docente que imparta la unidad de aprendizaje de Análisis y Diseño de Software debe contar con título de Ingeniero de Software o carreras afín, preferentemente con estudios de posgrado en Ciencias de la Computación, deberá comprobar al menos dos años de experiencia en el análisis y diseño de software y contar con experiencia en la docencia a nivel superior. Además, el docente deberá ser creativo, propositivo y tolerante con los estudiantes.