



## PROGRAMA DE ESTUDIO

### Programación Orientada a Objetos

#### Objetivo(s) del curso:

El alumno conocerá los conceptos de la Programación y el Diseño Orientado a Objetos que le permitirán la construcción de software reutilizable, escalable, de fácil mantenimiento y de mayor calidad.

#### Temario:

Tema	Nombre	Horas
1	Introducción al Paradigma Orientado a Objetos	14.0
2	Conceptos de Programación Orientada a Objetos	16.0
3	Lenguaje Unificado de Modelado (UML)	6.0
4	Principios del Diseño Orientado a Objetos	14.0
5	Patrones de diseño	14.0
	Total	<hr/> 64.0

#### 1. Introducción al Paradigma Orientada a Objetos

**Objetivo:** El alumno se familiarizará con los conceptos básicos de la teoría de Orientación a Objetos.

#### Contenido:

##### 1.1. Antecedentes históricos y problemática

1.1.1. Complejidad del software

1.1.2. Modularización

1.1.3. Reutilización de código

##### 1.2. Paradigma Orientado a Objetos

1.2.1. Objetos

1.2.2. Mensajes

1.2.3. Interfaces

### 1.3. Introducción al lenguaje de programación Java

- 1.3.1. La plataforma Java
- 1.3.2. Aspectos básicos del lenguaje (expresiones, operadores y control de flujo)
- 1.3.3. El API de Java y algunas utilerías

## 2. Conceptos de Programación Orientada a Objetos

**Objetivo:** El alumno conocerá a fondo los conceptos de la teoría Orientada a Objetos, sus implicaciones y alcances.

### Contenido:

#### 2.1. Conceptos de Programación Orientada a Objetos (POO)

- 2.1.1. Abstracción
- 2.1.2. Encapsulamiento
- 2.1.3. Composición
- 2.1.4. Herencia
- 2.1.5. Polimorfismo

#### 2.2. Implementación de conceptos POO en Java

- 2.2.1. Clases y creación de objetos
- 2.2.2. Modificadores de acceso
- 2.2.3. Paquetes y distribución de aplicaciones
- 2.2.4. Herencia
- 2.2.5. Clases abstractas e Interfaces
- 2.2.6. Polimorfismo

## 3. Lenguaje Unificado de Modelado (UML)

**Objetivo:** El alumno utilizará UML para representar relaciones entre objetos, su estructura y dinámica.

**Contenido:**

- 3.1. Diagramas de estructura
- 3.2. Diagramas de comportamiento
- 3.3. Herramientas para UML

**4. Principios de Diseño Orientado a Objetos**

**Objetivo:** El alumno aprenderá la importancia del uso del Diseño Orientado a Objetos en el desarrollo de software.

**Contenido:**

- 4.1. Principios de Diseño Orientado a Objetos
  - 4.1.1. Principios *SOLID* para diseño de clases.
- 4.2. Temas Java
  - 4.2.1. Manejo de excepciones
  - 4.2.2. Flujos de entrada/salida
  - 4.2.3. Hilos

**5. Patrones de diseño**

**Objetivo:** El alumno conocerá los patrones de diseño básicos para su aplicación en el diseño de software reusable.

**Contenido:**

- 5.1. Introducción a los patrones de diseño
  - 5.1.1. Intención
  - 5.1.2. Estructura
  - 5.1.3. Clasificación
- 5.2. Patrones de diseño básicos
  - 5.2.1. Patrones de creación
  - 5.2.2. Patrones de comportamiento
  - 5.2.3. Patrones estructurales

### 5.3. Temas especiales Java

5.3.1. Programación genérica

5.3.2. Expresiones lambda

### 5.4. Pruebas de software

5.4.1. Pruebas unitarias

5.4.2. Diseño Orientado a pruebas

#### Bibliografía básica.

#### Temas para los que se recomienda:

GAMMA, Erich [et al.] <b><i>Design Patterns. Elements of Reusable Object-Oriented Software.</i></b> Addison-Wesley, 1994	1,2,5
FREEMAN Eric [et al.] <b><i>Head First Design Patterns.</i></b> O'Reilly, 2004	2,5
FOWLER, Martin. <b><i>UML gota a gota.</i></b> Addison-Wesley, 1999	3,4,5
RUMBAUGH , James [et al.] <b><i>El Lenguaje Unificado de Modelado.</i></b> Pearson, 1999	3
RIEL , Arthur. <b><i>Object-Oriented Design Heuristics.</i></b> Addison Wesley, 1996	1,2