



# UNIVERSIDAD DE GUADALAJARA

## Licenciatura en Tecnologías de la Información

### 1. INFORMACIÓN DEL CURSO<sup>1</sup>

<b>Denominación:</b> Programación Orientada a Objetos	<b>Tipo:</b> Curso-taller	<b>Nivel:</b> Pregrado
<b>Área de formación:</b> Básica común, obligatoria	<b>Modalidad:</b> ( ) Escolarizada ( ) No escolarizada (X) Mixta	<b>Prerrequisitos:</b> I5630 - Estructura de datos
<b>Horas totales:</b> 64 <b>Horas teoría:</b> 32 <b>Horas prácticas:</b> 32	<b>Créditos:</b> 7	<b>Clave:</b> I9667
<b>Elaboró:</b> Centro Universitario de Los Valles – Dr. Francisco Eduardo Oliva Ibarra <b>Revisó:</b> Comité Curricular Intercentros, Academia de Ciencias de la Computación		<b>Fecha de elaboración:</b> 13/06/2017 <b>Fecha de revisión:</b> 01/02/2021

#### Relación con el perfil de egreso

Universidad como polo de desarrollo científico y tecnológico.

#### Relación con el plan de estudios

Fundamentos de programación, Estructura de Datos, Programación Web.

#### Campo de aplicación profesional de los conocimientos que promueve el desarrollo de la unidad de Aprendizaje

Perfil de egreso:

Habilidad	Nivel de aportación		
	Introdutorio	Medio	Avanzado
<i>empleado:</i>			
a. Dominar el paradigma de la programación orientada a objetos			X
b. Aplicar la programación en la gestión de la información			X
c. Desarrollar software para la gestión de la información			X
d. Uso de la programación sobre las redes digitales		X	
e. Generar habilidades interpersonales y analíticas aplicándolo a la programación de sistemas			X
f. Ser competitivo en las empresas			X
g. Liderazgo en una empresa sobre la gestión de la información			X
h. Motivación de crear software que sea adaptativo para las empresas en las que labora.		X	
i. Se promueve la buena comunicación con sus compañeros de equipo y con el profesor			X

<sup>1</sup> Este formato se trabajó con base en los términos de referencia del artículo 21 del Reglamento General de Planes de Estudio de la Universidad de Guadalajara.



## 2. DESCRIPCIÓN

### Objetivo general del curso

- El estudiante contará con la experiencia y bases teóricas/técnicas suficientes para implementar la solución de un problema real a partir del modelado del mismo y la implementación utilizando un lenguaje orientado a objetos.

### Objetivos específicos

- El estudiante comprenderá las características y conceptos generales de los lenguajes de programación orientada a objetos
- El estudiante fortalecerá su capacidad de abstracción para clasificar objetos reales, separando las propiedades relevantes de las que no lo son.
- El estudiante comprenderá los conceptos de protección y ocultamiento de atributos y métodos y los aplicará para el control eficiente de una aplicación.
- El estudiante será capaz de generar un modelo orientado a objetos de un problema dado a partir de: a) la identificación de las entidades participantes, b) clasificación de las mismas y c) las relaciones existentes entre ellas
- El estudiante será capaz de implementar aplicaciones de software, a partir de la traducción del modelo orientado a objetos y utilizando un LOO.

### Contenido temático

#### UNIDAD I

Introducción a la programación orientada a objetos

#### UNIDAD II

Abstracción. Clases y los objetos

#### UNIDAD III

Encapsulamiento

#### UNIDAD IV

Herencia y Polimorfismo

#### UNIDAD V

Diseñando GUIs con un lenguaje orientado a objetos

### Estructura conceptual del curso <sup>2</sup>

#### **UNIDAD I.**

**Introducción a la programación orientada a objetos**

- 1.1 Antecedentes**
- 1.2 Conceptos básicos de la programación orientada a objetos**
- 1.3 Variables primitivas y de referencia**
- 1.4 Lenguaje de modelado unificado (UML)**
- 1.5 Introducción a los IDE's**

#### **UNIDAD II:**

**Abstracción. Clases y objetos**

- 2.1 Abstracción**
- 2.2 Clases y objetos**
- 2.3 Declaración de la clase**
- 2.4 Atributos**
- 2.5 Métodos**
  - 2.5.1 Declaración**
  - 2.5.2 Invocación**

<sup>2</sup> La representación gráfica que permita visualizar la totalidad de componentes temáticos y del objetivo del curso, es la finalidad de representarlos a través de una infografía, mapa, wordle –www.wordle.net-, u otra forma nemotécnica que lo favorezca.



- 2.5.3 Devolución de datos
- 2.6 Ámbito de una variable
  - 2.5.1 Variables locales y variables miembro.
- 2.7 Declaración e instanciación de objetos
  - 2.7.1 Constructores
  - 2.7.2 destructores
- UNIDAD III**
- Encapsulamiento**
- 3.1 Accesibilidad de constructores
- 3.2 Ocultamiento
  - 3.2.1 Control de atributos y métodos
- 3.3 Overloading de constructores y métodos
- 3.4 Atributos y métodos de clase
- 3.5 Atributos de clase vs. atributos de objeto
- 3.6 Métodos de clase vs. métodos de objeto
- UNIDAD IV**
- Herencia y Polimorfismo**
- 4.1 Reusó de código
- 4.2 Conceptos de herencia simple y herencia múltiple
  - 4.2.1 Clases base
  - 4.2.2 Clase derivada
  - 4.2.3 Métodos concretos y Métodos abstractos
  - 4.2.4 Sobreescritura de métodos.
  - 4.2.5 clases abstractas
  - 4.2.6 Interfaces.
- 4.3 Especialización
- 4.4 Generalización
- UNIDAD V**
- Diseñando GUIs con un lenguaje orientado a objetos**
- 5.1 Introducción a librerías para desarrollo de GUIs (awt y Swing)
- 5.2 Componentes visuales. Jerarquía, creación y configuración
- 5.3. Diseño de GUIs con componentes visuales.
- 5.4 Manejo de eventos del teclado y del mouse.
- 5.5 Conexión a una base de datos, mediante una clase conexión

**Modalidades de evaluación**

Instrumento de evaluación	Factor de ponderación	Factor
<b>Convencional:</b> Exámenes por los primeros 4 modulo (5% de cada uno)	10 a 20	20%
<b>Desempeño:</b> Cuestionarios aplicados en todo el semestre	10 a 20	20%
<b>Desempeño:</b> Prácticas síncronas y asíncronas	10 a 20	20%
<b>Desempeño:</b> Proyecto Final	20 a 30	30%
<b>Desempeño:</b> Asistencia, respeto, participación en foros	0 a 5	5%
<b>Autoevaluación</b>	0 a 5	5%
<b>Total</b>		<b>100%</b>

**Elementos del desarrollo de la unidad de aprendizaje (asignatura)**

<b>Conocimientos</b>	<i>El estudiante, obtendrá los conocimientos básicos del paradigma de la programación orientada a objetos. Abstracción, clases, objetos, métodos, constructor, destructor,</i>
----------------------	--



# UNIVERSIDAD DE GUADALAJARA

## Licenciatura en Tecnologías de la Información

	<i>sobrecarga, encapsulamiento, herencia, sobre-escritura, polimorfismo, clases abstractas e interfaces y el uso de herramientas para el diseño de diagramas UML</i>
<b>Habilidades y Destrezas</b>	<i>El estudiante distinguirá las ventajas y desventajas del desarrollo de sistemas utilizando los diferentes paradigmas de la programación en el desarrollo de sistemas, hará el manejo correcto de la memoria dinámica, comprenderá el uso de los diferentes tipos de clases y comprenderá el uso del polimorfismo aplicado a los diferentes objetos creados en una aplicación de software. Aprenderá a proteger y reutilizar el código de una manera eficiente y aprenderá a utilizar diagramas UML.</i>
<b>Actitudes</b>	<i>Se tomará en cuenta la participación en clase, la resolución de problemas y el trabajo en equipo</i>
<b>Valores</b>	<i>Se tomará en cuenta la participación en clase, la resolución de problemas y el trabajo en equipo</i>

### 3. BIBLIOGRAFÍA BÁSICA *Incluir mínimo 3 referencias actualizadas*

<b>Autor(es)</b>	<b>Título</b>	<b>Editorial</b>	<b>Año *</b>	<b>URL o biblioteca digital donde está disponible (en su caso)</b>
<b>Paul Deitel, &amp; Harvey Deitel</b>	<b>Java, Cómo programar</b>	<b>Pearson</b>	<b>2016</b>	
<b>Danny Poo &amp; Derek Kiong, Swarnalatha</b>	<b>Object-Oriented Programming and Java</b>	<b>Springer</b>	<b>2008</b>	

### 4. BIBLIOGRAFÍA COMPLEMENTARIA *Incluir mínimo 5 referencias (podrán considerarse textos clásicos)*

<b>Autor(es)</b>	<b>Título</b>	<b>Editorial</b>	<b>Año *</b>	<b>URL o biblioteca digital donde está disponible (en su caso)</b>
<b>Booch, Grady, Martin, Robert, Newkirk, James</b>	<b>Object-Oriented Analysis and Design whit Applications</b>	<b>Addison Wesley</b>	<b>2007</b>	
<b>Wu, C. Thomas</b>	<b>Introducción a la Programación Orientada a Objetos con Java</b>	<b>Mc. Graw Hill</b>	<b>2001</b>	
<b>Booch, Grady, Rumbaugh, James, Jacobson, Ivar</b>	<b>UML El Lenguaje de Modelado Unificado (Guía de Usuario</b>	<b>Pearson</b>	<b>2012</b>	

*\*Se sugiere incluir el año de las ediciones más recientes, aun cuando se trate de obras que fueron impresas por primera vez hace varios años.*

Perfil del profesor:<sup>3</sup>

*Ing. En Computación o Lic. En Sistemas de Información*

<sup>3</sup> Considerar la formación disciplinar y pedagógica, así como la experiencia profesional de quien enseña la materia, son principios que unifican en la persona, saberes teóricos y prácticos que podrán favorecer el desarrollo de habilidades, aptitudes, valores y capacidades en el discente.



# UNIVERSIDAD DE GUADALAJARA

## Licenciatura en Tecnologías de la Información