

- Terry George R., "Principios de Administración", México, Ed. CECSA, 1998  
747 p

## 10. PERFIL DE LOS DOCENTES PARTICIPANTES EN LA UNIDAD DE APRENDIZAJE:

ESTUDIOS REQUERIDOS: Preparación profesional preferentemente en esta área o similares.  
EXPERIENCIA PROFESIONAL DESEABLE: 3 años de experiencia en el área  
OTROS REQUISITOS: Cursos de pedagogía y/o idioma inglés.



### Universidad Autónoma de Nayarit Área Económico-Administrativa Licenciatura en Informática

#### 1. DATOS DE IDENTIFICACIÓN

##### NOMBRE Y CLAVE DE LA UNIDAD DE APRENDIZAJE

Programación Orientada Objetos	
--------------------------------	--

##### DOCENTE(S) RESPONSABLE(S)

--

TIPO DE UNIDAD DE APRENDIZAJE	ACADEMIA
Obligatoria	Programación e Ingeniería de Software

ÁREA DE FORMACIÓN	LÍNEA DE FORMACIÓN	T.U.D.C.
Disciplinar	Programación e Ingeniería de Software	Curso - Taller

Horas teoría	Horas práctica	Horas de estudio independiente	Total de horas	Valor en créditos
40	39	17	96	6

FECHA DE ELABORACIÓN	FECHA DE ACTUALIZACIÓN
Junio 2012	

ELABORADO POR:	ACTUALIZADO POR:
Mtro. Rubén Paul Benítez Cortes Mtro. Marco Antonio Fernández Zepeda Revisado por: Comité Curricular de Informática	

## 2. PRESENTACIÓN

Tradicionalmente, la orientación a objetos se ha presentado como un estilo de programación donde el diseño se realiza alrededor de los datos a ser manipulados, en vez de centrarlo en las operaciones que debe realizar el sistema objeto del diseño. Las ventajas de este enfoque frente a la programación “tradicional” radican en la mayor estabilidad en el tiempo de los datos frente a las operaciones, lo que repercute en mejores posibilidades de reutilización y extensión del software desarrollado con este enfoque de programación de computadoras.

De ahí que el Comité Curricular del Programa Académico de Informática plantee la necesidad de implementar una unidad de aprendizaje que desarrolle las competencias en la Programación OO, denominado esta unidad de aprendizaje como Programación Orientada a Objetos, para impartirse en el área de formación disciplinar del plan de estudios

En este curso – taller se abordan los conceptos y la implementación del enfoque de programación orientado a objetos, con el propósito de analizar y diseñar soluciones informáticas que resuelvan problemas de gestión de información y de toma de decisiones de la empresa u organización.

Para cursar esta unidad de aprendizaje es necesario que el estudiante cuente con conocimientos previos en Análisis y Diseño de Algoritmos.

Esta unidad de aprendizaje corresponde al área disciplinar y cuenta con un valor curricular de 6 créditos con un total de 79 horas presenciales las que se dividirán en sesiones de 5 horas a la semana.

## 3. UNIDAD DE COMPETENCIA

**Objeto de estudio:** Programación Orientada a Objetos como un elemento clave en programación.

**Competencia:** Capacidad analítica y organizada para modelar y desarrollar sistemas de software orientado a objetos, con el propósito de mejorar la gestión y la toma de decisiones en las organizaciones públicas, sociales y privadas.

## 4. SABERES

Saberes Teóricos	Comprender los conceptos de la programación orientada a objetos. Conocer e identificar lenguajes orientados a objetos. Conocer e identificar herramientas para el modelado y desarrollo de programas orientados a objetos.
Saberes Prácticos	Aplicar los conceptos de la programación orientada a objetos con un lenguaje de programación. Resolver problemas computacionales a través del enfoque orientado a objetos. Modelar y desarrollar software orientado a objetos con el uso de herramientas para ese propósito.
Saberes Metodológicos	Métodos para modelar la realidad percibida por el estudiante a un enfoque computacional orientado a objetos.

Saber Formativos	Capacidad de abstracción Capacidad de análisis y síntesis. Capacidad de organizar y planificar. Solución de problemas. Toma de decisiones.
---------------------	--

## 5. DESGLOSE DE CONTENIDO (temática)

### Unidad 1 Conceptos de la Programación Orientación a Objetos

- 1.1 Introducción a la teoría de objetos
- 1.2 Abstracción
- 1.3 Encapsulamiento
  - Visibilidad
  - Ocultamiento
- 1.4 Análisis y Diseño de Clases (Entidades)
  - Características (Atributos)
  - Comportamientos (Métodos)
  - Modificadores de acceso
  - Polimorfismo estático
- 1.5 Herencia
  - Simple
  - Múltiple

### Unidad 2 Implementación de Clases en Java

- 2.1 Clases
  - Atributos
  - Métodos
- 2.2 Modificadores de acceso
- 2.3 Objetos
  - Constructores
  - Polimorfismo estático
- 2.4 Herencia
  - Herencia simple
  - Polimorfismo dinámico
  - Herencia múltiple
- 2.5 Empaquetado de clases
  - Ventajas de la utilización de paquetes
  - Organizar las clases de usuario en paquetes
  - Importar clases y paquetes de clases

### Unidad 3 Aspectos Básicos de un Lenguaje Orientado a Objetos

- 3.1 Arreglos y Colecciones
- 3.2 Manejo de errores y excepciones
- 3.3 Arreglos y colecciones
- 3.4 Corrientes de entrada/salida (de archivos)
- 3.5 Almacenamiento de objetos (serialización)
- 3.6 Desarrollo de un proyecto de altas, bajas, cambios y consultas con objetos serializados.

## 6. ACCIONES

### Unidad I Conceptos de la Programación Orientación a Objetos

- Investigar conceptos
- Realizar investigación y presentar los resultados para la discusión grupal.
- Resolver casos de estudios
- Lecturas obligatorias

### Unidad II Implementación de Clases en Java

- Reporte de Lecturas
- Ejercicios prácticos
- Resolver casos de estudios
- Dinámicas grupales

### Unidad III Aspectos Básicos de un Lenguaje Orientado a Objetos

- Revisión de vídeos
- Resolver casos de estudios
- Realizar mesas redondas para discutir conceptos que se manejan en el tema.
- Elaborar mapas conceptuales
- Elaborar mapas mentales
- Dinámicas grupales

## 7. CAMPO DE APLICACIÓN

Áreas de desarrollo de software orientado a objetos en los departamentos de Informática en las empresas del sector privado y organismos.

## 8. SISTEMA DE EVALUACIÓN

Evidencias de aprendizaje	Criterios de desempeño
<ul style="list-style-type: none"><li>— Mapa conceptual y resumen por unidad.</li><li>— Cuestionario por unidad</li><li>— Participación en clase.</li><li>— Examen departamental</li><li>— Exposición por parte de los alumnos</li><li>— Lecturas</li><li>— Mapas mentales</li><li>— Análisis de videos tutoriales</li><li>— Análisis de los casos de estudio</li></ul>	<ul style="list-style-type: none"><li>— Elabora un mapa conceptual por cada unidad que incluya los conceptos claves de la referencia bibliográfica.</li><li>— Realiza un resumen correspondiente a la unidad, con la amplitud indicada por el docente.</li><li>— Contesta el cuestionario correspondiente de cada unidad.</li><li>— Participación en clase con aportaciones sustantivas relacionadas con el tema.</li><li>— La exposición de clase se realiza utilizando los medios adecuados para presentarla en forma visual ya sea</li></ul>

por medio de acetatos o cañón. (No se permite leer el contenido).

## 9. CRITERIOS DE CALIFICACIÓN

Evidencias de aprendizaje	Porcentaje
Ejercicios	20%
Exámenes departamentales	40%
Portafolio de evidencias	5%
Resolución de casos de estudio	25%
Participación en plenarios grupales	10%

## 10. CRITERIOS DE ACREDITACIÓN

- Cumplir con el 80% de asistencias
- Calificación mínima para acreditar la unidad de aprendizaje será un acumulado de 60
- Los criterios escritos de los Trabajos de Investigación y la Resolución de casos de estudio son obligatorios a acreditarse con una calificación mínima de 60 c/u.

## 11. ACERVOS DE CONSULTA

### ACERVOS BÁSICOS

DAVID ARNOW, GERALD WEISS. **INTRODUCCIÓN A LA PROGRAMACIÓN CON JAVA**. PERSON 2009

CEBALLOS, FRANCISCO JAVIER. **JAVA CURSO DE PROGRAMACIÓN**, 3ª ED., RAMA, ESPAÑA, 2007.

GARCÍA FÉLIX, CARRETERO JESÚS, FERNÁNDEZ JAVIER, CALDERÓN ALEJANDRO. **EL LENGUAJE DE PROGRAMACIÓN JAVA**, EDIT. PEARSON EDUCACIÓN, ESPAÑA, 2002.

ZHANG, TONY. **APRENDIENDO JAVA EN 24 HORAS**, EDIT. PEARSON EDUCACIÓN, ESPAÑA, 2001.

### ACERVOS COMPLEMENTARIOS

JOHN S. MURPHY. **FUNDAMENTOS DE PROGRAMACIÓN DE COMPUTADORAS DIGITALES**, EDITORIAL LIMUSA, 1984.

ROBERT L, KRUSE. **ESTRUCTURAS DE DATOS Y DISEÑO DE PROGRAMAS**, MÉXICO, PRENTICE HALL, 1998.

SHNERDER, G. MICHAEL. **INTRODUCCIÓN A LA PROGRAMACIÓN Y LA SOLUCIÓN DE PROBLEMAS CON PASCAL**, MÉXICO, LIMUSA, 1986.

TENENBAUM AARON M. **ESTRUCTURAS DE DATOS EN C**, PRENTICE HALL, 1993.

WIRTH, NIKLAUS. **ALGORITMOS Y ESTRUCTURAS DE DATOS**, MÉXICO, PRENTICE HALL, 1987.

## **12. PERFIL DE LOS DOCENTES PARTICIPANTES EN LA UNIDAD DE APRENDIZAJE:**

Docentes con Licenciatura o grado de Maestría que dominen el conocimiento de las áreas computacionales y dominio de programación estructurada y orientada a objetos.

Docentes con la experiencia mínima de 3 años que dominen los aspectos mencionados.

Constancia de cursos didácticos y disciplinados. Manejo de técnicas didácticas.



**Universidad Autónoma de Nayarit**  
Área Económico-Administrativa  
Licenciatura en Informática

### **1. DATOS DE IDENTIFICACIÓN**

#### **NOMBRE Y CLAVE DE LA UNIDAD DE APRENDIZAJE**

Programación Visual y por Eventos	
-----------------------------------	--

#### **DOCENTE(S) RESPONSABLE(S)**

--