

## ASIGNATURA DE PROGRAMACIÓN ORIENTADA A OBJETOS

<b>1. Competencias</b>	Desarrollar soluciones tecnológicas para entornos Web mediante fundamentos de programación orientada a objetos, base de datos y redes de área local que atiendan las necesidades de las organizaciones.
<b>2. Cuatrimestre</b>	Segundo
<b>3. Horas Teóricas</b>	27
<b>4. Horas Prácticas</b>	63
<b>5. Horas Totales</b>	90
<b>6. Horas Totales por Semana Cuatrimestre</b>	6
<b>7. Objetivo de aprendizaje</b>	El alumno generará aplicaciones de software mediante el Paradigma Orientado a Objetos aplicando buenas prácticas en un lenguaje de programación para la solución de problemas específicos.

Unidades de Aprendizaje	Horas		
	Teóricas	Prácticas	Totales
<b>I. Paradigma Orientado a Objetos (POO).</b>	2	4	6
<b>II. Fundamentos de Programación Orientada a Objetos.</b>	7	17	24
<b>III. Programación Orientada a Objetos.</b>	18	42	60
<b>Totales</b>	<b>27</b>	<b>63</b>	<b>90</b>

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

# PROGRAMACIÓN ORIENTADA A OBJETOS

## UNIDADES DE APRENDIZAJE

<b>1. Unidad de aprendizaje</b>	<b>I. Paradigma Orientado a Objetos (POO).</b>
<b>2. Horas Teóricas</b>	2
<b>3. Horas Prácticas</b>	4
<b>4. Horas Totales</b>	6
<b>5. Objetivo de la Unidad de Aprendizaje</b>	El alumno representará gráficamente mediante un modelo conceptual el Paradigma Orientado a Objetos para resolver problemas determinados.

Temas	Saber	Saber hacer	Ser
Introducción al Paradigma Orientado a Objetos (POO).	Definir los principios del Paradigma Orientado a Objetos: abstracción, encapsulamiento, herencia y polimorfismo.		Analítico. Ordenado. Sistemático. Lógico.
Abstracción: clases y objetos.	Identificar la estructura básica de clase, atributo, método y objeto.	Esquematizar la estructura de una clase.	Analítico. Ordenado. Sistemático. Lógico.
Herencia.	Identificar las características y comportamiento de una clase padre a sus clases hijas.	Esquematizar la estructura de la herencia de clases.	Analítico. Ordenado. Sistemático. Lógico.
Polimorfismo.	Distinguir el polimorfismo de un objeto.	Esquematizar el polimorfismo.	Analítico. Ordenado. Sistemático. Lógico.

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

# PROGRAMACIÓN ORIENTADA A OBJETOS

## PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
Entrega la representación gráfica de clases (atributos y métodos) y herencia a partir de problemas planteados.	<ol style="list-style-type: none"><li>1. Comprender el concepto de clase y objeto, abstracción, encapsulamiento, herencia y polimorfismo.</li><li>2. Identificar los conceptos de clases, métodos, atributos, objetos, encapsulamiento, herencia en problemas planteados.</li><li>3. Comprender las clases, métodos, atributos y herencia involucrados en la solución de un problema.</li></ol>	<ul style="list-style-type: none"><li>- Ejercicios prácticos.</li><li>- Lista de cotejo.</li></ul>

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

# PROGRAMACIÓN ORIENTADA A OBJETOS

## PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none"><li>- Práctica demostrativa.</li><li>- Solución de problemas.</li><li>- Tareas de investigación.</li></ul>	<ul style="list-style-type: none"><li>Pizarrón.</li><li>Plumones.</li><li>Computadora.</li><li>Internet.</li><li>Equipo.</li><li>Multimedia.</li><li>Ejercicios prácticos.</li><li>Plataformas virtuales.</li></ul>

### ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
X		

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

# PROGRAMACIÓN ORIENTADA A OBJETOS

## UNIDADES DE APRENDIZAJE

<b>1. Unidad de aprendizaje</b>	<b>II. Fundamentos de Programación Orientada a Objetos.</b>
<b>2. Horas Teóricas</b>	7
<b>3. Horas Prácticas</b>	17
<b>4. Horas Totales</b>	24
<b>5. Objetivo de la Unidad de Aprendizaje</b>	El alumno codificará clases simples que contengan estructuras de control, tratamiento de cadenas de caracteres y conversión de datos mediante un lenguaje de programación para resolver problemas determinados.

Temas	Saber	Saber hacer	Ser
Arquitectura y configuración del Ambiente de desarrollo de la POO.	Identificar el entorno de desarrollo y configuración para el lenguaje de programación.	Establecer el entorno de desarrollo de acuerdo al lenguaje de programación.	Analítico. Ordenado. Sistemático. Lógico. Autodidacta. Responsable.
Aspectos léxicos.	Identificar la sintaxis de la declaración de clases, tipos de datos, atributos, variables, constantes, métodos, instancias y modificadores de acceso.	Programar clases, atributos y métodos usando diversos tipos de datos, expresiones aritméticas y palabras reservadas en el lenguaje de programación.	Analítico. Ordenado. Sistemático. Lógico. Responsable.
Estructuras de control.	Identificar la sintaxis y el funcionamiento de las estructuras de decisión y repetición.	Programar métodos empleando las estructuras de decisión y repetición en el lenguaje de programación.	Analítico. Ordenado. Sistemático. Lógico. Responsable.
Tratamiento de cadenas de caracteres.	Distinguir el tratamiento de cadenas y subcadenas de caracteres, así como la conversión a los tipos de datos.	Programar las clases que permitan el tratamiento de cadenas y subcadenas de caracteres, así como la conversión a los diferentes tipos de datos en el lenguaje de programación.	Analítico. Ordenado. Sistemático. Lógico. Responsable.

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

# PROGRAMACIÓN ORIENTADA A OBJETOS

## PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<p>Elabora un compendio de programas documentados que contengan la implementación de:</p> <ul style="list-style-type: none"> <li>• Tipos de datos.</li> <li>• Estructuras de decisión y repetición.</li> <li>• Tratamiento de cadenas.</li> <li>• Tipos de conversión.</li> </ul>	<ol style="list-style-type: none"> <li>1. Comprender la estructura de un programa y las reglas sintácticas de los tipos de datos, estructuras de repetición y decisión, tratamiento y conversiones de datos.</li> <li>2. Identificar la estructura de un programa y las reglas sintácticas de los tipos de datos, estructuras de repetición y decisión, tratamiento y conversiones de datos ante una problemática planteada.</li> <li>3. Codificar las reglas sintácticas de los tipos de datos, estructuras de repetición y decisión, tratamiento y conversiones de datos en la solución de un problema dado.</li> <li>4. Analizar los resultados obtenidos de la solución comparándolos contra lo esperado.</li> </ol>	<ul style="list-style-type: none"> <li>- Ejercicios prácticos.</li> <li>- Lista de cotejo.</li> </ul>

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

# PROGRAMACIÓN ORIENTADA A OBJETOS

## PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none"><li>- Práctica demostrativa.</li><li>- Solución de problemas.</li><li>- Prácticas de laboratorio.</li></ul>	<ul style="list-style-type: none"><li>Pizarrón.</li><li>Plumones.</li><li>Computadora.</li><li>Internet.</li><li>Equipo.</li><li>Multimedia.</li><li>Ejercicios.</li><li>Prácticos.</li><li>Plataformas virtuales.</li><li>IDE de desarrollo.</li></ul>

### ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
X	X	

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

# PROGRAMACIÓN ORIENTADA A OBJETOS

## UNIDADES DE APRENDIZAJE

<b>1. Unidad de aprendizaje</b>	<b>III. Programación Orientada a Objetos.</b>
<b>2. Horas Teóricas</b>	18
<b>3. Horas Prácticas</b>	42
<b>4. Horas Totales</b>	60
<b>5. Objetivo de la Unidad de Aprendizaje</b>	El alumno codificará clases empleando el paradigma de la Programación Orientada a Objetos para el desarrollo de aplicaciones.

Temas	Saber	Saber hacer	Ser
Clases, métodos y objetos.	Identificar la sintaxis de la creación de clases y métodos (constructores estáticos y no estáticos) así como la sintaxis de la declaración de objetos y la comunicación entre ellos.	Programar clases que implementen métodos (parametrizados, sin parámetros, estáticos, sin tipo de retorno y con tipo de retorno), constructores estáticos y no estáticos y la comunicación entre objetos.	Analítico. Ordenado. Sistemático. Lógico. Responsable.
Encapsulamiento.	Describir el encapsulamiento a través del funcionamiento y sintaxis de los modificadores de acceso (private, protected, public).	Programar clases empleando propiedades de los modificadores de acceso y métodos de acceso.	Analítico. Ordenado. Sistemático. Lógico. Responsable.
Herencia.	Identificar la sintaxis de la herencia, método sobrescrito, clases abstractas e interfaces.	Programar clases e interfaces que implementen la herencia.	Analítico. Ordenado. Sistemático. Lógico. Responsable.
Polimorfismo.	Identificar el funcionamiento del polimorfismo que utilice sobrecarga, sobrescritura de métodos y de referencias (casting).	Programar clases que implementen el polimorfismo.	Analítico. Ordenado. Sistemático. Lógico. Responsable.

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

Temas	Saber	Saber hacer	Ser
Manejo de Excepciones.	Identificar los tipos de excepciones y la jerarquía de clases correspondientes a éstas, así como la sintaxis de las cláusulas para el manejo de excepciones (try, catch, final, throw y throws).	Programar las excepciones generadas a través de las cláusulas apropiadas en el manejo de errores en tiempo de ejecución.	Analítico. Ordenado. Sistemático. Lógico. Responsable.

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

# PROGRAMACIÓN ORIENTADA A OBJETOS

## PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<p>Entrega en medio electrónico el ejecutable de la aplicación y código fuente comentado que incluya:</p> <ul style="list-style-type: none"><li>• Clases (atributos y métodos).</li><li>• Encapsulamiento.</li><li>• Herencia.</li><li>• Polimorfismo.</li><li>• Excepciones.</li></ul>	<ol style="list-style-type: none"><li>1. Comprender la sintaxis de clases, métodos, atributos, y objetos involucrados en el problema y las relaciones entre ellos.</li><li>2. Codificar la aplicación empleando encapsulamiento, abstracción, herencia y polimorfismo.</li><li>3. Aplicar el uso y el manejo de excepciones.</li><li>4. Analizar los resultados obtenidos de la solución comparándolos contra lo esperado.</li></ol>	<ul style="list-style-type: none"><li>- Ejercicios prácticos.</li><li>- Lista de cotejo.</li></ul>

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

# PROGRAMACIÓN ORIENTADA A OBJETOS

## PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none"><li>- Práctica demostrativa.</li><li>- Solución de problemas.</li><li>- Análisis de casos.</li></ul>	<p>Pizarrón. Plumones. Computadora. Internet. Equipo multimedia. Ejercicios prácticos. Plataformas virtuales. IDE de desarrollo.</p>

### ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
X	X	

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

## PROGRAMACIÓN ORIENTADA A OBJETOS

### CAPACIDADES DERIVADAS DE LAS COMPETENCIAS PROFESIONALES A LAS QUE CONTRIBUYE LA ASIGNATURA

Capacidad	Criterios de Desempeño
Establecer requerimientos funcionales y no funcionales mediante técnicas y metodologías de análisis de requerimientos para atender la necesidad planteada.	Entrega un documento de levantamiento de requerimientos que incluya: <ul style="list-style-type: none"> <li>• Fecha.</li> <li>• Nombre del Proyecto.</li> <li>• Objetivo.</li> <li>• Alcance.</li> <li>• Descripción funcional.</li> <li>• Requerimientos:                             <ul style="list-style-type: none"> <li>○ Software.</li> <li>○ Hardware.</li> </ul> </li> </ul>
Codificar aplicaciones Web a través de los fundamentos de programación orientada a objetos y conexión a base de datos para desarrollarla.	Entrega el código fuente documentado de la aplicación Web: <ul style="list-style-type: none"> <li>• Métodos.</li> <li>• Atributos.</li> <li>• Variables.</li> <li>• Conexión a la base de datos.</li> <li>• Componentes.</li> </ul>

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	

# PROGRAMACIÓN ORIENTADA A OBJETOS

## FUENTES BIBLIOGRÁFICAS

Autor	Año	Título del Documento	Ciudad	País	Editorial
Paul deitel	2016 9786073238021	<i>Cómo programar en java, 10a edición.</i>	CDMX	México	Pearson Educacion
Paul deitel	2015 9786073227391	<i>Cómo programar en C++, 12a edición.</i>	CDMX	México	Pearson Educacion
Bruno López Takeyas	2017 9786076226599	Curso de programación orientada a objetos en C#.net. Ejemplos con aplicaciones visuales y de consola.	CDMX	México	Alfaomega Grupo Editor
Dusty Phillips	2015 9781784395957	<i>Python 3 object-oriented programming - second edition.</i>	California	Estados Unidos	Packt Publishing
Julie Anderson, Hervé Franceschi	2016 9781284045314	<i>Java illuminated an active learning approach 4ta edición.</i>	N/D	Estados Unidos	Library of Congress Cataloging
Boyarsky, Jeanne	2016 9781119272090	<i>OCA/OCP JAVA SE 8 programmer certification kit.</i>	California	Estados Unidos	Sybex

<b>ELABORÓ:</b>	Comité técnico académico de diseño curricular del subsistema CGUTyP de la familia de carreras de Tecnologías de la Información.	<b>REVISÓ:</b>	Dirección Académica	
<b>APROBÓ:</b>	C. G. U. T. y P.	<b>FECHA DE ENTRADA EN VIGOR:</b>	Septiembre de 2018	