|  |  |  |
| --- | --- | --- |
|  | TÉCNICO SUPERIOR UNIVERSITARIO EN TECNOLOGÍAS DE LA INFORMACIÓN ÁREA DESARROLLO DE SOFTWARE MULTIPLATAFORMA  EN COMPETENCIAS PROFESIONALES | descarga |

**ASIGNATURA DE PROGRAMACIÓN ORIENTADA A OBJETOS**

|  |  |
| --- | --- |
| 1. **Competencias** | 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. |
| 1. **Cuatrimestre** | Segundo |
| 1. **Horas Teóricas** | 27 |
| 1. **Horas Prácticas** | 63 |
| 1. **Horas Totales** | 90 |
| 1. **Horas Totales por Semana Cuatrimestre** | 6 |
| 1. **Objetivo de aprendizaje** | 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** |
| Paradigma Orientado a Objetos (POO). | 2 | 4 | 6 |
| Fundamentos de Programación Orientada a Objetos. | 7 | 17 | 24 |
| Programación Orientada a Objetos. | 18 | 42 | 60 |
| **Totales** | **27** | **63** | **90** |

# PROGRAMACIÓN ORIENTADA A OBJETOS

*UNIDADES DE APRENDIZAJE*

|  |  |
| --- | --- |
| 1. **Unidad de aprendizaje** | 1. **Paradigma Orientado a Objetos (POO).** |
| 1. **Horas Teóricas** | 2 |
| 1. **Horas Prácticas** | 4 |
| 1. **Horas Totales** | 6 |
| 1. **Objetivo de la Unidad de Aprendizaje** | 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. |

**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. | 1. Comprender el concepto de clase y objeto, abstracción, encapsulamiento, herencia y polimorfismo.    2. Identificar los conceptos de clases, métodos, atributos, objetos, encapsulamiento, herencia en problemas planteados.    3. Comprender las clases, métodos, atributos y herencia involucrados en la solución de un problema. | - Ejercicios prácticos.  - Lista de cotejo. |

# PROGRAMACIÓN ORIENTADA A OBJETOS

*PROCESO ENSEÑANZA APRENDIZAJE*

|  |  |
| --- | --- |
| **Métodos y técnicas de enseñanza** | **Medios y materiales didácticos** |
| - Práctica demostrativa.  - Solución de problemas.  - Tareas de investigación. | Pizarrón.  Plumones.  Computadora.  Internet.  Equipo.  Multimedia.  Ejercicios prácticos.  Plataformas virtuales. |

*ESPACIO FORMATIVO*

|  |  |  |
| --- | --- | --- |
| **Aula** | **Laboratorio / Taller** | **Empresa** |
| **X** |  |  |

# PROGRAMACIÓN ORIENTADA A OBJETOS

*UNIDADES DE APRENDIZAJE*

|  |  |
| --- | --- |
| 1. **Unidad de aprendizaje** | 1. **Fundamentos de Programación Orientada a Objetos.** |
| 1. **Horas Teóricas** | 7 |
| 1. **Horas Prácticas** | 17 |
| 1. **Horas Totales** | 24 |
| 1. **Objetivo de la Unidad de Aprendizaje** | 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. |

**PROGRAMACIÓN ORIENTADA A OBJETOS**

*PROCESO DE EVALUACIÓN*

|  |  |  |
| --- | --- | --- |
| **Resultado de aprendizaje** | **Secuencia de aprendizaje** | **Instrumentos y tipos de reactivos** |
| Elabora un compendio de programas documentados que contengan la implementación de:   * Tipos de datos. * Estructuras de decisión y repetición. * Tratamiento de cadenas. * Tipos de conversión. | 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.  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.    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.    4. Analizar los resultados obtenidos de la solución comparándolos contra lo esperado. | - Ejercicios prácticos.  - Lista de cotejo. |

# PROGRAMACIÓN ORIENTADA A OBJETOS

*PROCESO ENSEÑANZA APRENDIZAJE*

|  |  |
| --- | --- |
| **Métodos y técnicas de enseñanza** | **Medios y materiales didácticos** |
| - Práctica demostrativa.  - Solución de problemas.  - Prácticas de laboratorio. | Pizarrón.  Plumones.  Computadora.  Internet.  Equipo.  Multimedia.  Ejercicios.  Prácticos.  Plataformas virtuales.  IDE de desarrollo. |

*ESPACIO FORMATIVO*

|  |  |  |
| --- | --- | --- |
| **Aula** | **Laboratorio / Taller** | **Empresa** |
| **X** | **X** |  |

# PROGRAMACIÓN ORIENTADA A OBJETOS

*UNIDADES DE APRENDIZAJE*

|  |  |
| --- | --- |
| 1. **Unidad de aprendizaje** | 1. **Programación Orientada a Objetos.** |
| 1. **Horas Teóricas** | 18 |
| 1. **Horas Prácticas** | 42 |
| 1. **Horas Totales** | 60 |
| 1. **Objetivo de la Unidad de Aprendizaje** | 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, sobreescritura de métodos y de referencias (casting). | Programar clases que implementen el polimorfismo. | Analítico.  Ordenado.  Sistemático.  Lógico.  Responsable. |
| 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. |

**PROGRAMACIÓN ORIENTADA A OBJETOS**

*PROCESO DE EVALUACIÓN*

|  |  |  |
| --- | --- | --- |
| **Resultado de aprendizaje** | **Secuencia de aprendizaje** | **Instrumentos y tipos de reactivos** |
| Entrega en medio electrónico el ejecutable de la aplicación y código fuente comentado que incluya:   * Clases (atributos y métodos). * Encapsulamiento. * Herencia. * Polimorfismo. * Excepciones. | 1. Comprender la sintaxis de clases, métodos, atributos, y objetos involucrados en el problema y las relaciones entre ellos.    2. Codificar la aplicación empleando encapsulamiento, abstracción, herencia y polimorfismo.  3. Aplicar el uso y el manejo de excepciones.  4. Analizar los resultados obtenidos de la solución comparándolos contra lo esperado. | - Ejercicios prácticos.  - Lista de cotejo. |

# PROGRAMACIÓN ORIENTADA A OBJETOS

*PROCESO ENSEÑANZA APRENDIZAJE*

|  |  |
| --- | --- |
| **Métodos y técnicas de enseñanza** | **Medios y materiales didácticos** |
| - Práctica demostrativa.  - Solución de problemas.  - Análisis de casos. | Pizarrón.  Plumones.  Computadora.  Internet.  Equipo multimedia.  Ejercicios prácticos.  Plataformas virtuales.  IDE de desarrollo. |

*ESPACIO FORMATIVO*

|  |  |  |
| --- | --- | --- |
| **Aula** | **Laboratorio / Taller** | **Empresa** |
| **X** | **X** |  |

# 

# 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:   * Fecha. * Nombre del Proyecto. * Objetivo. * Alcance. * Descripción funcional. * Requerimientos: * Software. * Hardware. |
| 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:   * Métodos. * Atributos. * Variables. * Conexión a la base de datos. * Componentes. |

# 

# PROGRAMACIÓN ORIENTADA A OBJETOS

*FUENTES BIBLIOGRÁFICAS*

| **Autor** | **Año** | **Título del Documento** | **Ciudad** | **País** | **Editorial** |
| --- | --- | --- | --- | --- | --- |
| Paul deitel | 2016 9786073238021 | *Cómo programar en java, 10a edición.* | CDMX | México | Pearson Educacion |
| Paul deitel | 2015 9786073227391 | *Cómo programar en C++, 12a edición.* | 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 | *Python 3 object-oriented programming - second edition.* | California | Estados Unidos | Packt Publishing |
| Julie Anderson, Hervé Franceschi | 2016 9781284045314 | *Java illuminated an active learning approach 4ta edición.* | N/D | Estados Unidos | Library of Congress Cataloging |
| Boyarsky, Jeanne | 2016 9781119272090 | *OCA/OCP JAVA SE 8 programmer certification kit.* | California | Estados Unidos | Sybex |

CIZG