



MF0227_3 PROGRAMACION ORIENTADA A OBJETOS



Presencial



250 horas



100%
Subvencionado

Este curso se ajusta a lo expuesto en el itinerario de aprendizaje perteneciente al Módulo Formativo MF0227_3 Programación Orientada a Objetos, regulado en el Real Decreto 628/2013, de 2 de Agosto, que permitirá al alumnado adquirir conocimientos sobre los principios de la programación orientada a objetos, el modelo de programación web y bases de datos y el ciclo de vida del desarrollo de aplicaciones.

El curso se impartirá en **modalidad presencial en:**

Sevilla. C/Rafael Belmonte García, 4 41010.



Contenidos:



UNIDAD FORMATIVA 1.

PRINCIPIOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

1. Recorrido por el ciclo de desarrollo del software en el contexto de la programación orientada a objetos: desde el análisis hasta la programación.
2. Exploración del proceso de creación de software: importancia de la modularidad.
3. Diferenciación del concepto de módulo en el marco de la programación orientada a objetos.
4. Reconocimiento de objetos como representaciones abstractas de entidades del mundo real.

UNIDAD DIDÁCTICA 2. CLASES Y OBJETOS

1. Comprensión del concepto de clase, incluyendo sus atributos, métodos y el principio de encapsulación.
2. Análisis de los objetos en términos de estado, comportamiento e identidad.
3. Uso de objetos como instancias de una clase, así como la referencia a la instancia actual (this, self, current).
4. Definición de programa en el ámbito de la programación orientada a objetos: POO = Objetos + Mensajes.

UNIDAD DIDÁCTICA 3. GENERALIZACIÓN/ESPECIALIZACIÓN: HERENCIA

1. Explicación del concepto de herencia: tanto simple como múltiple.
2. Diferenciación de la herencia múltiple.
3. Creación de objetos dentro del contexto de la herencia.
4. Clasificación jerárquica de las clases.

UNIDAD DIDÁCTICA 4. RELACIONES ENTRE CLASES

1. Diferenciación entre agregación y composición.
2. Distinguir entre generalización y especialización.
3. Identificación de diferentes tipos de asociaciones entre clases.

UNIDAD DIDÁCTICA 5. ANÁLISIS DEL POLIMORFISMO

1. Definición del concepto de polimorfismo.
2. Diversos tipos de polimorfismo.

UNIDAD DIDÁCTICA 6. TÉCNICAS DE PROGRAMACIÓN ESTRUCTURADA

1. Identificación de los elementos fundamentales: constantes, variables, operadores y expresiones.
2. Estudio de las estructuras de control: secuencia, condiciones y bucles.
3. Diferenciación entre funciones y procedimientos.
4. Ejemplificación de llamadas a funciones y procedimientos.
5. Uso de funciones y procedimientos definidos dentro de las clases.

UNIDAD DIDÁCTICA 7. ESTRUCTURA DE LA INFORMACIÓN

1. Clasificación de datos simples: numéricos (enteros y reales), booleanos, caracteres, cadenas de texto y punteros o referencias de memoria.
2. Introducción a los datos estructurados: Arrays.
3. Gestión de memoria y sus mecanismos asociados.



UNIDAD DIDÁCTICA 8. LENGUAJES DE PROGRAMACIÓN ORIENTADOS A OBJETOS

1. Exploración del lenguaje de programación en el contexto de la programación orientada a objetos y sus paradigmas.
2. Comparativa entre los lenguajes de programación orientados a objetos más utilizados y sus características clave.
3. Uso de librerías de clases en proyectos de desarrollo.

UNIDAD DIDÁCTICA 9. IMPLEMENTACIÓN DEL PARADIGMA UTILIZANDO UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

1. Selección del lenguaje adecuado para el desarrollo.
2. Clasificación de tipos de aplicaciones que se pueden desarrollar.
3. Herramientas de desarrollo recomendadas.
4. Tipos de datos y elementos básicos del lenguaje elegido; instrucciones y su uso.
5. Estudio y aplicación de clases básicas incluidas en las librerías de clases.
6. Definición de nuevas clases.
7. Gestión de eventos y su implementación.
8. Uso de hilos en programación.
9. Definición y análisis de la programación en red.
10. Acceso a bases de datos desde aplicaciones, incluyendo librerías de clases asociadas.

UNIDAD FORMATIVA 2. MODELO DE PROGRAMACIÓN WEB Y BASES DE DATOS

UNIDAD DIDÁCTICA 1. INTRODUCCIÓN AL DESARROLLO DE APLICACIONES EN EL MODELO DE PROGRAMACIÓN WEB

1. Análisis de la arquitectura web: cliente ligero, servidor web, servidor de aplicaciones y servidor de datos.
2. Enumeración de protocolos y tecnologías comunes en el desarrollo web.
3. Revisión de los modelos de programación que son estándares en la industria.
4. Uso de componentes orientados a objetos como base para el desarrollo de aplicaciones en un entorno web.

UNIDAD DIDÁCTICA 2. ARQUITECTURA MULTICAPA (N-TIER).

1. Estudio de la arquitectura multicapa.
2. Diferenciación y análisis del modelo de tres capas en la web: presentación, aplicación y datos.
3. Diseño de arquitecturas de aplicación basadas en el modelo multicapa.
4. Comprensión del concepto de lógica de negocio y la función de la capa lógica.

UNIDAD DIDÁCTICA 3. LA CAPA DE PRESENTACIÓN

1. Descripción de la capa de presentación: introducción al lenguaje de hipertexto.
2. Análisis de la capa de presentación avanzada: lenguajes de scripting y lenguajes de hipertexto dinámico.
3. Revisión de lenguajes orientados a la creación de la capa de presentación y la ejecución de solicitudes desde clientes ligeros web (JSP, Servlets, ASP, PHP).

UNIDAD DIDÁCTICA 4. DISEÑO DE BASES DE DATOS RELACIONALES

1. Definición clara de qué son las bases de datos relacionales.
2. Proceso de diseño de bases de datos en varios niveles.



3. Análisis de diferentes tipos de relaciones y su implementación en bases de datos.

4. Descripción del lenguaje utilizado para acceder a bases de datos.
5. Comparación entre el modelo relacional y el modelo orientado a objetos.
6. Nociones sobre cómo se almacenan objetos en bases de datos relacionales.

UNIDAD DIDÁCTICA 5. ACCESO A BASES DE DATOS RELACIONALES: CAPA DE ACCESO A DATOS

1. Análisis del API para acceder a la base de datos.
2. Descripción del nivel controlador.
3. Interfaz de acceso a la base de datos (driver) y su funcionamiento.
4. Análisis del nivel de aplicación.
5. Establecimiento de conexión con una base de datos.
6. Operaciones sobre la base de datos: sentencias del lenguaje de acceso, objetos para ejecutar consultas y manipular resultados.
7. Integración de tipos de datos del lenguaje de acceso a bases de datos dentro del lenguaje de programación de la aplicación.
8. Uso de procedimientos almacenados.
9. Gestión de transacciones distribuidas.

UNIDAD DIDÁCTICA 6. LENGUAJES DE DEFINICIÓN DE DATOS

1. Conceptos básicos y estándares en lenguajes de definición de datos.
2. Uso del lenguaje de definición de datos (DDL SQL) en sistemas de gestión de bases de datos actuales.
3. Diferenciación de elementos del estándar SQL-92 frente a otros elementos en bases de datos comerciales.
4. Instrucciones para crear elementos: CREATE.
5. Instrucciones para modificar elementos: ALTER.
6. Instrucciones para eliminar elementos: DROP, TRUNCATE.

UNIDAD DIDÁCTICA 7. MANIPULACIÓN DE LOS DATOS

1. Uso del lenguaje de manipulación de datos (DML SQL).
2. Realización de consultas de datos: SELECT.
3. Inserción de datos: INSERT.
4. Modificación de datos: UPDATE.
5. Eliminación de datos: DELETE.
6. Agregación de conjuntos de datos mediante consultas: JOIN, UNION.
7. Uso de subconsultas.

UNIDAD FORMATIVA 3. EL CICLO DE VIDA DEL DESARROLLO DE APLICACIONES

UNIDAD DIDÁCTICA 1. PROCESO DE INGENIERÍA DEL SOFTWARE

1. Diferenciación de las etapas del proceso de ingeniería de software: especificación, diseño, construcción y pruebas unitarias, validación, implementación y mantenimiento.
2. Revisión de modelos de proceso de ingeniería: modelo en cascada, desarrollo evolutivo, desarrollos formales, entre otros.
3. Identificación de requisitos: concepto, evolución y trazabilidad.
4. Análisis de metodologías de desarrollo centradas en objetos.



5. Resolución de un caso práctico utilizando UML en metodologías de desarrollo.

6. Definición de herramientas CASE y su importancia en el desarrollo de software.

UNIDAD DIDÁCTICA 2. PLANIFICACIÓN Y SEGUIMIENTO

1. Realización de estimaciones para la planificación del proyecto.
2. Creación de planificaciones utilizando modelos diagramáticos, como el diagrama de Gantt.
3. Análisis del proceso de seguimiento mediante reuniones e informes.

UNIDAD DIDÁCTICA 3. DIAGRAMADO

1. Identificación de los principios básicos de UML y su aplicación.
2. Uso de diagramas de casos de uso.

UNIDAD DIDÁCTICA 4. DESARROLLO DE LA GUI

1. Estudio del modelo basado en componentes y eventos.
2. Diferenciación de los elementos que componen la interfaz gráfica de usuario (GUI).
3. Presentación del diseño centrado en el usuario, incluyendo nociones de usabilidad.
4. Utilización de herramientas para el diseño de interfaces gráficas.

UNIDAD DIDÁCTICA 5. CALIDAD EN EL DESARROLLO DEL SOFTWARE

1. Identificación de criterios de calidad en el desarrollo de software.
2. Análisis de métricas y estándares de calidad aplicables.

UNIDAD DIDÁCTICA 6. PRUEBAS

1. Diferenciación de los tipos de pruebas disponibles.
2. Análisis de pruebas de defectos, incluyendo pruebas de caja negra, pruebas estructurales, pruebas de trayectorias, pruebas de integración y pruebas de interfaces.

UNIDAD DIDÁCTICA 7. EXCEPCIONES

1. Definición de excepciones, fuentes de errores, tratamiento y prevención de fallos, así como excepciones definidas y lanzadas por el programador.
2. Uso de excepciones tratadas como objetos en programación.

UNIDAD DIDÁCTICA 8. DOCUMENTACIÓN

1. Cómo elaborar un documento técnico.
2. Estructura recomendada para el documento.
3. Generación automática de documentación a partir del código.



Requisitos para realizar el Curso

El curso está 100% subvencionado (**gratuito**) para trabajadores desempleados y trabajadores ocupados por cuenta propia o ajena.

Los **requisitos de acceso** son: Cumplir al menos uno de los siguientes requisitos:

- Título de Bachiller.
- Certificado de Profesionalidad de nivel III o Certificado de Profesionalidad de nivel II de la misma familia y área profesional
- Prueba de acceso a FP Superior superada.
- Prueba de acceso a la universidad para mayores de 25 años y/o de 45 años.

¡Contacta con nosotros y te informaremos!



955 12 70 10



671 05 52 88



formacionsevilla@corenetworks.es