

# Programación Orientada a Objetos con C#

## **Nro. Horas: 10**

En las últimas décadas, el paradigma de Orientación a Objetos es escogido para el desarrollo de muchos de los sistemas de información que existen hoy en día. Es necesario una comprensión de este paradigma, para aplicarlos en la práctica, y así conseguir sus mayores beneficios como: producir sistemas bien estructurados, buena actualización y alta productividad.

En este curso, abordaremos elementos conceptuales de la POO, demostrando una implementación práctica usando el lenguaje de programación C#, paso a paso con Visual Studio. Veremos conceptos como asociaciones, herencia y polimorfismo, acceso a base de datos SQL Server usando ADO.NET y Entity Framework.

## **¿Qué aprenderás?:**

Aprenderemos a aplicar los conceptos de POO utilizando el lenguaje de programación C#:

- Introducción a Programación Orientada a Objetos (POO)
- Iniciando la implementación de POO
- Asociaciones e inicialización de Objetos
- Colecciones, agregaciones, identidad y recuperación de objetos
- Herencia, polimorfismo y excepciones
- Solución dividida en capas
- Acceso a base de datos con ADO.NET
- Entity Framework

## **Prerrequisitos:**

- Fundamentos de programación

## **Contenidos:**

### **Introducción a Programación Orientada a Objetos**

1. Algunos componentes de OO
2. Características de OO
3. Escogiendo la plataforma del lenguaje
4. Conclusiones

### **Iniciando la implementación de POO**

1. La primera clase
2. Tipos de datos
3. Una aplicación de prueba e inicio con Visual Studio
4. Ejecución de la clase de prueba
5. Otros ejemplos de clases
6. Conclusiones

### **Asociaciones e inicialización de Objetos**

1. Identificación e implementación de asociaciones
2. Test de asociaciones
3. Inicialización de objetos
4. Composición como asociación
5. Otros ejemplos de asociaciones
6. Conclusiones

### **Colecciones, agregaciones, identidad y recuperación de objetos**

1. Uso de colecciones
2. Identidad de un objeto
3. Agregación como asociación
4. Recuperación de objetos de una colección por medio de LINQ
5. Otros ejemplos con colecciones
6. Conclusiones

### **Herencia, polimorfismo y excepciones**

1. Herencia por extensión
2. Herencia por implementación, con interfaces y polimorfismo
3. Excepciones
4. Sobrecarga y sobrescritura de métodos
5. Otro ejemplo de herencia por extensión
6. Otro ejemplo de herencia por implementación
7. Conclusiones

## **Solución dividida en capas**

1. Contextualización sobre capas
2. Los proyectos que representan las capas
3. Implementación de interacción con el usuario
4. Encapsulamiento
5. Conclusiones

## **Acceso a base de datos**

1. Introducción a ADO.NET y creación de la base de datos con Visual Studio
2. Realizando operaciones relacionadas de CRUD con una tabla de BD
3. Conclusiones

## **Entity Framework como ORM**

1. Creación del proyecto para aplicación de EF Core
2. Recuperación de datos con EF Core
3. Guardando datos con el EF Core
4. Asociaciones con el EF Core
5. Datos de prueba para nuestra BD
6. Ejemplo final
7. Conclusiones