

# C#, desarrollo en .NET con Visual Studio

Curso práctico de 5 días - 35h

Ref.: PSH - Precio 2025: 2 120€ sin IVA

Este curso presenta .NET Framework y .Net Core. Los ejercicios prácticos muestran cómo programar en modo orientado a objetos, utilizando Visual Studio y el lenguaje C#, para explotar las clases que ofrece Microsoft. También se trata el acceso a datos con ADO.NET y LINQ y los servicios web ASP.NET.

## OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Descubra las principales tecnologías del marco .NET

Dominar la sintaxis del lenguaje C#

Implementación de la programación orientada a objetos con C#

Uso del entorno de desarrollo integrado Visual Studio

Al principio del curso, se ilustran los aspectos sintácticos del lenguaje C# y las clases básicas mediante ejemplos sencillos.

## PROGRAMA

última actualización: 07/2024

### 1) La plataforma .NET

- Principio y arquitectura de la plataforma .NET.
- Arquitectura .NET Framework: CLR, BCL y CLS (soporte multilingüe).
- Visión general de los distintos tipos de aplicaciones en un entorno multiobjetivo.
- Estructura de una aplicación .NET: noción de espacio de nombres.
- Herramientas y entorno de desarrollo.
- Lenguaje CIL: principio del lenguaje intermedio, principio de la compilación JIT (Just In Time Compiler).
- Noción de montaje, metadatos y despliegue.
- Ensamblado privado y ensamblado compartido: firma, papel del GAC (Global Assembly Cache), implementación en el GAC.
- Evolución de .NET Core, código abierto y multiplataforma.

*Trabajo práctico* : Ejemplo de programa mínimo en C#. Ejecución en modo gestionado.

*Utilización del entorno de desarrollo Visual Studio.NET para escribir el primer programa.*

### 2) Sintaxis básica: datos, expresiones e instrucciones

- Variables y expresiones: declaración, constantes, operadores, tipos anónimos y dinámicos.
- Tipos de datos: Common Type System, papel de la clase base System.Object y transtyping.
- ¿Cuáles son las diferencias entre los tipos de valor y los tipos de referencia?
- Sintaxis básica: manipulación de matrices, instrucciones de control de flujo, etc.
- Nuevas funciones en C# 6 (null-conditional, propiedades automáticas mejoradas, funciones "con cuerpo de expresión", etc.).
- Nuevas funciones de C# 7 (legibilidad de constantes, variables "out", tuplas, concordancia de patrones, retorno de referencias, etc.).

*Trabajo práctico* : Escribir programas en C# utilizando algoritmos tradicionales.

### PARTICIPANTES

Desarrolladores que deseen adquirir las técnicas para desarrollar todo tipo de aplicaciones .NET con Visual Studio.

### REQUISITOS PREVIOS

Buenos conocimientos de programación. Conocimientos básicos de conceptos orientados a objetos. Se requiere experiencia en el desarrollo de software utilizando un lenguaje de tipo C/C++ o Java.

### COMPETENCIAS DEL FORMADOR

Los expertos que imparten la formación son especialistas en las materias tratadas. Han sido validados por nuestros equipos pedagógicos, tanto en el plano de los conocimientos profesionales como en el de la pedagogía, para cada curso que imparten. Cuentan al menos con entre cinco y diez años de experiencia en su área y ocupan o han ocupado puestos de responsabilidad en empresas.

### MODALIDADES DE EVALUACIÓN

El formador evalúa los progresos pedagógicos del participante a lo largo de toda la formación mediante preguntas de opción múltiple, escenificaciones de situaciones, trabajos prácticos, etc. El participante también completará una prueba de posicionamiento previo y posterior para validar las competencias adquiridas.

### MEDIOS PEDAGÓGICOS Y TÉCNICOS

- Los medios pedagógicos y los métodos de enseñanza utilizados son principalmente: ayudas audiovisuales, documentación y soporte de cursos, ejercicios prácticos de aplicación y ejercicios corregidos para los cursillos prácticos, estudios de casos o presentación de casos reales para los seminarios de formación.
- Al final de cada cursillo o seminario, ORSYS facilita a los participantes un cuestionario de evaluación del curso que analizarán luego nuestros equipos pedagógicos.
- Al final de la formación se entrega una hoja de presencia por cada media jornada de presencia, así como un certificado de fin de formación si el alumno ha asistido a la totalidad de la sesión.

### MODALIDADES Y PLAZOS DE ACCESO

La inscripción debe estar finalizada 24 horas antes del inicio de la formación.

### ACCESIBILIDAD DE LAS PERSONAS CON DISCAPACIDAD

¿Tiene alguna necesidad específica de accesibilidad? Póngase en contacto con la Sra. FOSSE, interlocutora sobre discapacidad, en la siguiente dirección [psh-accueil@orsys.fr](mailto:psh-accueil@orsys.fr) para estudiar de la mejor forma posible su solicitud y su viabilidad.

### 3) Gestión de excepciones

- Cómo la filosofía de las excepciones debería cambiar tu forma de programar.
- Activar errores personalizados con "throw".
- Uso de excepciones para la gestión centralizada de errores.

*Trabajo práctico* : Tratamiento de errores de entrada mediante excepciones.

### 4) Programación orientada a objetos

- Clases y objetos.
- Modelización del mundo real basada en objetos.
- Conceptos de campo, método y propiedad.
- Patrimonio.
- Polimorfismo.
- Interfaces para herencia múltiple.

### 5) Clases y objetos en C#

- Definición de clases. Definición de objetos.
- Definición del contenido de la clase: métodos y propiedades.
- Visibilidad de los miembros de la clase: propiedades y métodos. Utilización de espacios de nombres.
- Ciclo de vida de los objetos: constructor, destructor. Gestión de memoria con Garbage Collector.
- El mecanismo de sobrecarga (constructores, métodos y operadores).
- Derivación y herencia de clases: principio de derivación, control de acceso durante la derivación.
- El principio del polimorfismo.
- Principio de las interfaces (definición e implementación, utilización para la herencia múltiple).
- Manejo de atributos: el principio de los metadatos. Atributos de clase, método y campo.
- Regiones de código y clases parciales. Clases genéricas.
- Delegados, covarianza, contravarianza y eventos. Varianza aplicada a interfaces.
- Clases empresariales similares a las del marco .NET: propiedades, indexadores y enumeradores.
- Generación de documentación.

*Trabajo práctico* : Escritura de clases básicas. Manipulación de métodos y propiedades en las clases. Derivación de clases. Ejemplo de implementación de una interfaz e implementación del polimorfismo por herencia o con una interfaz.

### 6) Objetos y clases base de .NET Framework

- ¿Qué es un marco? Cómo funciona. Jerarquía de clases.
- Tratamiento de fechas y duraciones. Procesamiento de cadenas con StringBuilder y expresiones regulares.
- Clases esenciales: manipulación del sistema de archivos, Matemáticas, Aleatorio, etc.
- Los diferentes tipos de colecciones, diccionarios y tablas hash.
- Colecciones genéricas y bases Linq con LINQ To Object sobre colecciones estándar.

*Trabajo práctico* : Utilizar expresiones regulares y una tabla hash. Hacer compatible con LINQ una colección creada previamente y utilizar LINQ To Objects para consultarla.

### 7) Los distintos tipos de aplicaciones desarrolladas en .NET

- Principio y diseño de una biblioteca de clases reutilizables.
- Principio y ejemplo de aplicaciones WPF.
- Principio y ejemplo de los servicios web ASP.NET.
- Introducción al acceso a datos con ADO.NET o Entity Framework y LINQ.
- Principio y ejemplo de aplicaciones Web Asp.Net MVC.

*Trabajo práctico* : Creación de un formulario Windows para consultar una base de datos. Demostración de Asp.Net MVC con un controlador y una página Razor, mostrando datos. Demostración de un servicio Web sencillo.

## FECHAS

---

Contacto