

***CURSO de ESPECIALIZACIÓN en:***  
**TECNOLOGÍAS .NET**

*“Conviértete en un Desarrollador .NET full-stack”*

<b>MÓDULOS</b>	Programación en C# con Visual Studio	3 semanas
	ASP.NET Core	3 semanas
	.NET MAUI	3 semanas
	.NET AVANZADO	3 semanas
<b>REQUISITOS</b>	Conocimientos en programación y Nociones de Bases de Datos (SQL Server)	
<b>DURACIÓN</b>	<b>3 meses de Lunes a Viernes</b>	

## 1. Programación en C# con Visual Studio

**OBJETIVO:** Brindar a los participantes los conocimientos y habilidades necesarias para desarrollar aplicaciones utilizando el lenguaje de programación C# en el entorno de desarrollo Visual Studio, mediante la comprensión de los conceptos fundamentales de la programación orientada a objetos y la aplicación práctica de herramientas y técnicas modernas de desarrollo.

### CONTENIDO

1. Conociendo la interface de Visual Studio
2. Programación estructurada
  - a) Tipos de datos y variables
  - b) Parse
  - c) Constantes
  - d) Métodos
  - e) Rescatar valores de consola
  - f) Manejo de errores
  - g) Operadores
  - h) Condicionales
  - i) Método TryParse
  - j) Switch
  - k) Condicionales simples, compuestos, anidados

- l) Bucles (loops)
- m) break & continue
- 3. Programación orientada a Objetos
  - a) Clases y objetos
  - b) Constructores
  - c) Múltiples constructores
  - d) Propiedades
- 4. Sobrecarga de operadores
- 5. Gráficos
- 6. Manejo de archivos

## 2. ASP.NET Core

**OBJETIVO:** Formar a los participantes en el desarrollo de aplicaciones web dinámicas y robustas utilizando ASP.NET Core .NET 8 (LTS), a través del uso del patrón MVC, la integración con bases de datos y la implementación de buenas prácticas de diseño, seguridad y despliegue en entornos reales.

### CONTENIDO

- 1. Fundamentos de desarrollo de aplicaciones Web con .NET
  - a) ASP.NET MVC
    - i. Modelo MVC
    - ii. Ciclo de vida de aplicaciones Web y peticiones
    - iii. Métodos de petición
  - b) Partes de una Aplicación Web MVC
    - i. Controladores, action methods
    - ii. Filtros
    - iii. Selectores
    - iv. ActionVerbs
    - v. Validación y tratamiento de errores
    - vi. Lectura y escritura como acciones
    - vii. Vistas con HtmlHelpers y Bootstrap
    - viii. Binding de modelos
- 2. Razor & Blazor
- 3. Uso de NuGet
  - a) Descarga de librerías
  - b) Actualización de librerías

4. Acceso a datos
  - a) Entity Framework
    - i. Capa desconectada
    - ii. Capa conectada
    - iii. Vinculación con modelos
  - b) Acceso manual a datos
    - i. Data Adapter
    - ii. Consultas parametrizadas
  - c) Seguridad en las aplicaciones Web
  - d) Autenticación y autorización de usuarios
    - i. Conceptos preliminares
    - ii. Cuentas de usuarios individuales

### 3. .NET MAUI

**OBJETIVO:** Al finalizar el curso, los participantes podrán desarrollar aplicaciones multiplataforma utilizando .NET MAUI, entendiendo su arquitectura, el uso de C# y XAML para interfaces de usuario, y la implementación de APIs para operaciones CRUD. Además, aprenderán a crear un servicio de datos que optimice el rendimiento en un entorno Android.

#### CONTENIDO

1. ¿Qué es .NET MAUI?
  - a) Cómo trabaja
  - b) Conceptos de UI
  - c) C# y XAML
2. Nuestra aplicación
  - a) Su arquitectura
  - b) Usando API DbContext
  - c) API end points
    - i. Read
    - ii. Create
    - iii. Update
    - iv. Delete
3. Data Service
  - a) Como interface

- b) La implementación
  - c) Métodos de petición
4. Entorno Android

## 4. .NET avanzado

**OBJETIVO:** Capacitar a los participantes en el desarrollo de APIs REST y arquitecturas de Microservicios con .NET 8 (LTS), utilizando herramientas como Entity Framework, Docker, Kubernetes y certificados personalizados con Kestrel, para crear aplicaciones backend modernas, seguras y escalables.

### CONTENIDO

1. MVC Rest API con .NET 8
  - a) Creación de end-points
  - b) EF & SQL Server
  - c) Data Transfer Object
  - d) End-points de la API mediante métodos PUT, PATCH, DELETE
2. Microservicios .NET
  - a) Introducción
  - b) Data layers
  - c) Docker & kubernetes
  - d) API multirecurso
  - e) Mensajes asíncronos
3. Deploy de .NET Core API en Docker
4. Kestrel para certificados personalizados
  - a) HTTPS, SSL y TLS
  - b) Localhost
  - c) Dominio personalizado
5. Paquetes NuGet personalizados