



Diplomado en

**FULLSTACK**  
DEVELOPER

Diplomado en

# FULLSTACK DEVELOPER

Aprenderás a crear tus sitios web partiendo desde el inicio hasta el final. Te sumergirás en las mejores prácticas del desarrollo, trabajando de la parte del front-end Angular y de la parte del desarrollo backend con lenguaje de programación .NET. Aprenderás a trabajar con Bases de datos SQL dominando la gestión de los datos en forma eficiente, ágil y con una gran facilidad de escalabilidad.

## Información general



Duración  
120 horas



Horario  
martes, miércoles y jueves  
de 6:30 p.m. a 9:30 p.m.



Metodología y sedes  
Online.  
Sede Bogotá.

## Justificación

El ser desarrollador Full Stack, te permitirá crear e integrar el back end y front end de una aplicación web completa, accediendo a una base de datos relacional utilizando las mejores prácticas de la industria .NET, siendo capaz de controlar de punta a punta las soluciones de software requeridas en cualquier entorno laboral.



## Perfiles de ingreso

En este curso aprenderás a crear tus sitios web partiendo desde el inicio hasta el final. Te sumergirás en las mejores prácticas del desarrollo, trabajando de la parte del front-end Angular y de la parte del desarrollo backend con lenguaje de programación .NET. Aprenderás a trabajar con Bases de datos SQL dominando la gestión de los datos en forma eficiente, ágil y con una gran facilidad de escalabilidad.

Conocimientos Requeridos:

- HTML y CSS básico.
- JavaScript básico.

## Propósitos de formación

El Diplomado tiene como propósito socializar los conocimientos suficientes al estudiante relacionados con los lenguajes de programación tanto de backend y de frontend brindando las herramientas suficientes para desarrollar aplicaciones web completas.

En la actualidad se manejan muchos lenguajes de programación de software orientados a objetos y C# .NET es uno de los pilares de estos lenguajes, conforme a la continua evolución de los sistemas de información dentro de las compañías, C# .NET siendo un lenguaje de programación estructurado, brinda la posibilidad de realizar múltiples sistemas de información dependiendo de los requerimientos y necesidades del cliente.

## Competencias a desarrollar

Las competencias que se desarrollarán durante este curso serán las siguientes:

- Dominio de lenguajes de programación Angular (Front-end) y .NET (Backend).
- Capacidad de diseñar y producir componentes de software inteligente para ser integrado en aplicaciones funcionales y técnicas.
- Modificar otras que ya estén programadas con estos lenguajes.
- Entender muy bien cómo funciona un producto web de principio a fin, desde su diseño en mockup y despliegue.
- Dominio de SQL para la gestión de los datos en forma eficiente.
- Manejo de herramientas para gestionar requerimientos y ambientes de desarrollo.

## Plan de estudios

Módulo  
**01**

### Bases de Datos SQL

Módulo I:

#### Contextualización a las bases de datos

- ¿Qué son las bases de datos y para que sirven?
- Tipos de bases de datos.
- Motores de bases de datos.
- Bases de datos relacionales.
- Bases de datos no relacionales.
- Consolidación de conocimientos (examen básico).

Módulo II:

#### Lenguaje SQL

- Introducción al lenguaje SQL.
- Crear una base de datos.
- Cláusulas y funciones.
- Ejercicio Práctico.

Módulo III:

#### DML (Data Manipulation Language)

- Instrucción SELECT.
- Instrucción INSERT.
- Instrucción UPDATE.
- Instrucción DELETE.
- Ejercicio práctico.

Módulo IV:

#### DDL (Data Definition Language)

- Instrucción CREATE.
- Instrucción DROP.
- Instrucción ALTER.
- Instrucción TRUNCATE.
- Ejercicio práctico.

Módulo V:

#### Control de flujo de datos

- Procedimientos almacenados.
- Cómo crear los procedimientos almacenados.
- Realizar usando flujos de entrada - salida.
- Ejercicio práctico.

Módulo VI:

#### Proyecto

- Creación de una base de datos transaccional, para trabajar durante todo el curso.



Módulo  
**02**

## Programación Backend con lenguaje C# .NET

<p>Módulo I:</p> <p><b>Revisión de la Sintaxis de C#</b></p> <ul style="list-style-type: none"> <li>- Resumen de las aplicaciones de escritura usando C#.</li> <li>- Tipos de datos, operadores y expresiones.</li> <li>- Construcciones del lenguaje de programación C#.</li> <li>- Ejercicios prácticos.</li> </ul>	<p>Módulo II:</p> <p><b>Creación de métodos y control de excepciones</b></p> <ul style="list-style-type: none"> <li>- Tipos de métodos.</li> <li>- Crear e invocar los métodos.</li> <li>- Métodos sobrecargados.</li> <li>- Métodos con parámetros opcionales.</li> <li>- Control de excepciones.</li> <li>- Ejercicio práctico.</li> </ul>
<p>Módulo III:</p> <p><b>Creación de clases e implementación de interfaces</b></p> <ul style="list-style-type: none"> <li>- Definición de las clases en c#.</li> <li>- Creación de clases.</li> <li>- Definición e implementación de interfaces.</li> <li>- Tipos de colecciones.</li> <li>- Ejercicio práctico.</li> </ul>	<p>Módulo IV:</p> <p><b>Creación de una jerarquía de clases mediante Herencia</b></p> <ul style="list-style-type: none"> <li>- Definición de Herencia en c#.</li> <li>- Creación de jerarquías de clase.</li> <li>- Extendiendo clases mediante herencia.</li> <li>- Ejercicio práctico.</li> </ul>
<p>Módulo V:</p> <p><b>Lectura y escritura de datos locales</b></p> <ul style="list-style-type: none"> <li>- Lectura y escritura de archivos.</li> <li>- Serializar y deserializar datos.</li> <li>- Realizar usando flujos de entrada - salida.</li> <li>- Ejercicio práctico.</li> </ul>	<p>Módulo VI:</p> <p><b>Acceso a una base de datos</b></p> <ul style="list-style-type: none"> <li>- Creación y uso de modelos de datos (entidades y/o modelos).</li> <li>- Creación de métodos para acceder a la base de datos.</li> <li>- Consultar datos de una tabla.</li> <li>- Ejercicio práctico.</li> </ul>
<p>Módulo VII:</p> <p><b>Contextualización a las APIS Rest</b></p> <ul style="list-style-type: none"> <li>- ¿Qué es un api rest?</li> <li>- ¿Cómo funciona una api rest?</li> <li>- ¿Para qué sirve y cómo se comunica un api rest?</li> <li>- ¿Qué son los endpoint y para que sirven?</li> <li>- Consolidación de conocimientos (examen básico).</li> </ul>	<p>Módulo VIII:</p> <p><b>Concepto de Capas y MVC para la construcción de una aplicación</b></p> <ul style="list-style-type: none"> <li>- Definición de la arquitectura de software.</li> <li>- Contextualización de las capas para la construcción de software.</li> <li>- Responsabilidades de las capas.</li> <li>- Concepto de vista modelo y controlador.</li> <li>- Ejercicio práctico.</li> </ul>
<p>Módulo IX:</p> <p><b>Construyendo un Api REST con .NET</b></p> <ul style="list-style-type: none"> <li>- Construcción de solución de software para la API REST</li> <li>- Aplicando el concepto de capas para la construcción del software.</li> <li>- Aplicando el conocimiento de Interfaces.</li> <li>- Exponer un endpoint para consumir desde el front.</li> </ul>	<p>Módulo X:</p> <p><b>Control de versiones</b></p> <ul style="list-style-type: none"> <li>- ¿Qué es git y como funciona?</li> <li>- Versionando nuestro software.</li> </ul>

Módulo  
**03**

## Programación Front-end con Angular

<p>Módulo I:</p> <p><b>Instalaciones y primeros pasos</b></p> <ul style="list-style-type: none"> <li>- Instalación de Angular.</li> <li>- Repaso de HTML.</li> <li>- Repaso de CSS.</li> <li>- Repaso JavaScript.</li> <li>- Crear un proyecto en Visual Studio Code.</li> <li>- Consolidación de conocimientos (examen básico).</li> </ul>	<p>Módulo II:</p> <p><b>Componentes y plantillas</b></p> <ul style="list-style-type: none"> <li>- ¿Qué son los componentes y como funcionan?</li> <li>- Crear componente.</li> <li>- Utilización de múltiples componentes.</li> <li>- Ejercicio práctico.</li> </ul>
<p>Módulo III:</p> <p><b>TypeScript con Angular</b></p> <ul style="list-style-type: none"> <li>- ¿Qué es TypeScript y como funciona?</li> <li>- Constructores, métodos de los componentes.</li> <li>- Propiedades y tipos de datos.</li> <li>- Creación de clases, objetos y modelos de datos.</li> <li>- Ejercicio práctico.</li> </ul>	<p>Módulo IV:</p> <p><b>Directivas, binding, Eventos</b></p> <ul style="list-style-type: none"> <li>- ¿Qué son las directivas en Angular?</li> <li>- Directiva ngLif y evento clic.</li> <li>- Directiva ngClass.</li> <li>- Directiva ngFor.</li> <li>- Ejercicio práctico.</li> </ul>
<p>Módulo V:</p> <p><b>Routing, configuración y navegación</b></p> <ul style="list-style-type: none"> <li>- ¿Cómo configurar el routing?</li> <li>- Redirecciones con el ROUTER.</li> <li>- Menú de navegación.</li> <li>- Rutas y parámetros por la URL.</li> <li>- Ejercicio práctico.</li> </ul>	<p>Módulo VI:</p> <p><b>Servicios</b></p> <ul style="list-style-type: none"> <li>- ¿Qué son los servicios?</li> <li>- Crear y utilizar un servicio.</li> <li>- Ejercicio práctico.</li> </ul>
<p>Módulo VII:</p> <p><b>Pipes y filtros</b></p> <ul style="list-style-type: none"> <li>- ¿Qué son los pipes y como se utilizan?</li> <li>- Pipes para fechas.</li> <li>- Filtros para transformación de caracteres.</li> <li>- Pipes personalizados.</li> <li>- Ejercicio práctico</li> </ul>	<p>Módulo VIII:</p> <p><b>Servicios, Http y manejo de Ajax</b></p> <ul style="list-style-type: none"> <li>- Crear servicio para hacer peticiones HTTP.</li> <li>- Operator map.</li> <li>- rxjs/Observable.</li> <li>- Peticiones AJAX en Angular (HTTP, Llamadas a servicios REST).</li> <li>- Práctica - Consumir API Rest construida.</li> </ul>
<p>Módulo IX:</p> <p><b>Proyecto - Crear web app para mostrar la información del API</b></p> <ul style="list-style-type: none"> <li>- Creación y maquetación de (HTML, CSS, JavaScript)</li> <li>- Aplicando el concepto de capas para la construcción del software.</li> <li>- Página home.</li> <li>- Página para operaciones CRUD (operando con API desarrollada en el modulo Backend)</li> <li>- Configurar el ROUTER.</li> <li>- Página de error 404.</li> </ul>	

## Requisitos para ingreso

- Formulario de inscripción diligenciado y con firma.
- 2 fotografías tipo documento 3×4 fondo blanco.\*
- Fotocopia de la cédula de ciudadanía ampliada al 150% o equivalente según el país de origen del estudiante, pasaporte y visa.

\* Las fotografías deben ser en alta resolución, no tener más de 6 meses de antigüedad, centrada y enfocada, la cara debe aparecer mirando directamente a la cámara, no de perfil ni mirando por encima del hombro, y no debe haber sombras sobre la cara ni sobre el fondo. No se aceptarán fotos con gafas de fantasía ni con reflejos en los cristales, ni con sombrero, gorro, pañuelo o visera. En caso de traer la documentación por medio físico, se debe tener en cuenta que las fotografías deben ser impresas en papel de calidad fotográfica (no papel común).

## Descuentos y Entidades FINANCIERAS

Para mayor información sobre los **DESCUENTOS** que ofrece la Universidad a sus aspirantes y egresados:  
[http://www.unipiloto.edu.co/descargas/DESCUENTOS\\_EC.pdf](http://www.unipiloto.edu.co/descargas/DESCUENTOS_EC.pdf)

Aplican convenios con **ENTIDADES FINANCIERAS** vigentes. para mayor información:  
[http://www.unipiloto.edu.co/descargas/ENTIDADES-FINANCIERAS\\_2019.pdf](http://www.unipiloto.edu.co/descargas/ENTIDADES-FINANCIERAS_2019.pdf)

La información estará sujeta a cambios según disposiciones de cada entidad.

## Contáctenos

Posgrados y Educación Continuada  
Universidad Piloto de Colombia  
[postgrados@unipiloto.edu.co](mailto:postgrados@unipiloto.edu.co)  
[www.unipiloto.edu.co](http://www.unipiloto.edu.co)

**PBX: 580 0968**  
Calle 45 A No. 9 - 17  
Bogotá - Colombia.

"La Universidad se reserva el derecho de apertura o aplazamiento de los programas en caso de no contar con el número mínimo de inscritos. El grupo docente estará sujeto a cambios según disponibilidad de su agenda al igual que el cronograma y horarios de actividades académicas"

VIGILADA MINEDUCACIÓN

Institución de educación superior sujeta a la inspección y vigilancia del Ministerio de Educación Nacional de Colombia. Reconocimiento de personería jurídica como institución de educación superior con Resolución No. 3681 del 27 de noviembre de 1962 del Ministerio de Educación Nacional de Colombia. Código institución: 1815.

Vigencia desde 2023