

Introducción a ASP.NET Core

Manual de estudiante



Yazmín Nabor López
y
Miguel Muñoz Serafín





Introducción a ASP.NET Core

Manual de estudiante

Primera edición

Marzo de 2018



Contenido

Introducción.....	4
Acerca del módulo.....	5
Audiencia.....	5
Objetivos	5
Requerimientos	5
Contenido del módulo.....	6
Lección 1 Introducción a ASP.NET Core	7
¿Qué es .NET Core?	8
¿Qué es ASP.NET Core?	14
Elegir entre ASP.NET y ASP.NET Core.....	16
Ejercicio 1: Preparando el entorno de desarrollo.	18
Ejercicio 2: Creando tu primera aplicación .NET Core.....	22
Ejercicio 3: Creando tu primera aplicación ASP.NET Core desde línea de comandos	25
Ejercicio 4: Creando tu primera aplicación ASP.NET Core desde Visual Studio.....	29
Ejercicio 5: Explorando el código.	31
Lección 2 Creando una aplicación web UI con ASP.NET Core MVC	38
¿Qué es ASP.NET Core MVC?	39
Ejercicio 1: Creando una aplicación web UI ASP.NET Core MVC	41
Ejercicio 2: Configurando la clase Startup.....	43
Ejercicio 3: Agregando un Controlador	45
Ejercicio 4: Agregando una Vista.....	47
Ejercicio 5: Agregando un Modelo	50
Ejercicio 6: Implementando un repositorio utilizando Inyección de Dependencias.....	52
Ejercicio 7: Implementando funcionalidad en el Controlador y las Vistas.....	59
Lección 3 Creando una aplicación web UI Razor Pages con ASP.NET Core	64
¿Qué es Razor Pages?.....	65
Ejercicio 1: Creando una aplicación web UI Razor Pages.	66
Ejercicio 2: Agregando el Modelo y el Servicio de la aplicación.....	69
Ejercicio 3: Agregando Razor Pages	74



Lección 4 Creando una API web con ASP.NET Core	86
Introducción a Web API.....	87
Ejercicio 1: Creando una API web.....	88
Ejercicio 2: Explorando el código.....	90
Ejercicio 3: Agregando el Modelo y el Servicio de la aplicación.....	92
Ejercicio 4: Implementando funcionalidad en el Controlador	97
Resumen.....	101

Introducción

Hola, ¿qué tal? mi nombre es Miguel Muñoz Serafín responsable del entrenamiento “**Introducción a ASP.NET Core**”.

En la actualidad, una de las tecnologías de desarrollo de aplicaciones Web que ha tomado un gran impacto es **ASP.NET Core**. Algunos de los factores que podemos destacar de ASP.NET Core es que además de ser un Framework de código abierto, nos proporciona herramientas para el desarrollo de aplicaciones multiplataforma, lo que significa que ahora podemos desarrollar aplicaciones Web y hospedarlas en servidores Windows, Linux o macOS.

Este entrenamiento surge de la necesidad de crear una guía introductoria de aprendizaje sobre ASP.NET Core. A lo largo de este entrenamiento se presentan conceptos teóricos y una serie de ejercicios con la finalidad de que el participante se introduzca en el desarrollo de aplicaciones web utilizando el framework ASP.NET Core y Visual Studio de una manera sencilla y práctica. La información de este entrenamiento ha sido tomada de sitios oficiales de Microsoft, así como de la interacción con los grupos de desarrollo de ASP.NET.

Hemos desarrollado el contenido de este entrenamiento con colaboración especial de Yazmín Nabor López quien se ha dado a la tarea de diseñar el material plasmándolo en una serie de ejercicios paso a paso para facilitar el aprendizaje.

Yazmin es licenciada en Ciencias de la Computación, en el año 2004 se inicia profesionalmente en el desarrollo de software participando en múltiples proyectos desarrollados bajo la plataforma .NET. Desde el año 2014 es parte del equipo de TI Capacitación participando principalmente en el área de desarrollo de software y generación de contenido.

¡Reciban una bienvenida a este entrenamiento!

Miguel Muñoz Serafín



<https://mvp.microsoft.com/es-es/PublicProfile/21053>

Acerca del módulo

ASP.NET Core es un rediseño de *ASP.NET* con cambios arquitectónicos que dan como resultado un framework más compacto y modular. Al ser ASP.NET Core un framework de código abierto, multiplataforma y de alto rendimiento, nos permite construir aplicaciones web modernas, conectadas a internet y basadas en la nube.

En este módulo, se presenta una introducción a **.NET Core**, **ASP.NET Core** y al desarrollo de aplicaciones web con ASP.NET Core. Se describen los pasos a seguir para preparar el ambiente de desarrollo con Visual Studio para la creación de aplicaciones web con ASP.NET Core. Se incluyen ejercicios para crear aplicaciones web UI MVC, web UI Razor Pages y web API utilizando C# y ASP.NET Core.

Audiencia

Este módulo está dirigido a desarrolladores con experiencia en el desarrollo de aplicaciones web con ASP.NET MVC y C#.

Objetivos

Al finalizar este módulo, los participantes contarán con las habilidades y conocimientos para:

- Crear una aplicación Web MVC básica con ASP.NET Core.
- Crear una aplicación Web Razor Pages básica con ASP.NET Core.
- Crear una aplicación Web API básica con ASP.NET Core.

Requerimientos

Para poder realizar los ejercicios de este módulo, es necesario contar con lo siguiente:

- Un equipo de desarrollo con Visual Studio 2017. Puede utilizarse la versión gratuita **Visual Studio Community** que puede descargarse desde el siguiente enlace:

<https://www.visualstudio.com/community/>

Los ejemplos descritos en este módulo fueron diseñados con Visual Studio Enterprise 2017 sobre una máquina con Windows 10 Pro.

Contenido del módulo

El contenido de este entrenamiento consta de 4 lecciones.

Lección 1: Introducción a ASP.NET Core.

En esta lección se proporciona una introducción a .NET Core y a ASP.NET Core. Se describen también los pasos a seguir para preparar el ambiente de desarrollo con Visual Studio para la creación de aplicaciones web con ASP.NET Core. Se proporcionan ejercicios para crear una aplicación *¡Hello World!* tanto con .NET Core como con ASP.NET Core utilizando la herramienta **dotnet** y Visual Studio. Adicionalmente, se explica el propósito de la clase **Startup**.

Lección 2: Creando una aplicación web UI MVC con ASP.NET Core.

En esta lección se presenta una introducción a ASP.NET Core MVC y se proporcionan los pasos para crear una aplicación web UI con ASP.NET Core MVC y C# desde Visual Studio. Se explica el uso de servicios aplicando la técnica de **Inyección de Dependencias (DI)**.

Lección 3: Creando una aplicación web UI Razor Pages con ASP.NET Core.

En esta lección se proporcionan los pasos para crear una aplicación web UI Razor Pages con ASP.NET Core y C# desde Visual Studio. Se proporciona una introducción a Razor Pages, la nueva característica de ASP.NET Core 2.0 que hace que la codificación de escenarios centrados en páginas sea más fácil y productiva.

Lección 4: Creando una API web con ASP.NET Core.

En esta lección se presenta una introducción al desarrollo de aplicaciones API web con ASP.NET Core MVC y C# desde Visual Studio. A lo largo de esta lección se desarrollará una API web utilizando la nueva plantilla Web API.



Lección 1

Introducción a ASP.NET Core

En esta lección se proporciona una introducción a .NET Core y a ASP.NET Core. Se describen también los pasos a seguir para preparar el ambiente de desarrollo con Visual Studio para la creación de aplicaciones web con ASP.NET Core. Se proporcionan ejercicios para crear una aplicación *¡Hello World!* tanto con .NET Core como con ASP.NET Core y se explica el propósito de la clase **Startup**.

Objetivos de la lección

Al finalizar esta lección, los participantes podrán:

- Definir qué es .NET Core.
- Definir qué es ASP.NET Core.
- Configurar su ambiente de desarrollo Visual Studio para comenzar a desarrollar aplicaciones web con ASP.NET Core.

¿Qué es .NET Core?

.NET Core es una implementación modular, multiplataforma y de código abierto del **.NET Standard** diseñada para que sea portátil entre plataformas a fin de permitir la reutilización del código al máximo y su uso compartido. Contiene muchas de las mismas APIs que contiene el **.NET Framework** (pero **.NET Core** es un conjunto más pequeño) e incluye componentes de tiempo de ejecución (runtime), framework, compilador y herramientas que soportan una variedad de sistemas operativos y procesadores. La implementación de .NET Core fue orientada principalmente para las cargas de trabajo de ASP.NET Core, pero también por la necesidad y el deseo de tener una implementación más moderna.

.NET Core es una plataforma de desarrollo de propósito general mantenida por Microsoft y la comunidad .NET en *GitHub*. Es multiplataforma, compatible con *Windows*, *macOS* y *Linux* y puede utilizarse en dispositivos, en la nube y escenarios integrados/IoT.

.NET Core es portátil entre plataformas porque, aunque se trata de un subconjunto de la versión completa del .NET Framework, proporciona una funcionalidad clave para implementar las características que una aplicación necesita y reutilizar ese código independientemente del destino de la plataforma.

Antes, las distintas versiones de .NET para diferentes plataformas carecían de funcionalidad compartida para las tareas clave, como por ejemplo la lectura de archivos locales. Las plataformas Microsoft que pueden establecerse como destino de una aplicación o biblioteca .NET Core incluyen Windows de escritorio tradicional, así como dispositivos y teléfonos Windows. Cuando se usan herramientas como *Xamarin*, .NET Core es portátil para dispositivos iOS y Android. Además, .NET Core está disponible para los sistemas operativos Mac y Linux para permitir que las aplicaciones web se ejecuten en estos sistemas.

.NET Core es modular ya que se publica a través de *NuGet* en paquetes de ensamblado más reducidos. En lugar de un ensamblado grande que contenga la mayor parte de la funcionalidad básica, .NET Core está disponible como paquetes más pequeños centrados en características específicas. Esto nos permite un modelo de desarrollo más ágil y nos ofrece la posibilidad de elegir las funcionalidades que necesitamos para nuestras aplicaciones y bibliotecas.

Las siguientes son las principales características de .NET Core:

- **Multiplataforma:** .NET Core proporciona funcionalidad clave para implementar las características que necesita una aplicación y reutilizar ese código independientemente de la plataforma destino. Actualmente soporta tres sistemas operativos principales: Windows, Linux y macOS. Podemos escribir aplicaciones y bibliotecas que se ejecuten sin modificaciones a través de los sistemas operativos soportados. Los sistemas operativos soportados, CPUs y escenarios de aplicación crecerán con el tiempo, proporcionados por Microsoft, otras empresas y particulares.