

Programador Jr. de Aplicaciones ASP.NET MVC

Manual de estudiante



Miguel Muñoz Serafín

Programador Jr. de aplicaciones ASP.NET MVC

Manual de estudiante

Primera edición

Febrero de 2018

Soporte técnico:

soporte@mail.ticapacitacion.com

<https://ticapacitacion.com/curso/mvcjr>

Contenido

Acerca del curso	7
Audiencia.....	7
Objetivos	7
Requerimientos	8
Contenido del curso	8
Módulo 1: Explorando ASP.NET MVC	12
Lección 1 Visión general de las tecnologías Web Microsoft	14
Introducción a las tecnologías Web de Microsoft.....	15
Visión general de ASP.NET	19
Tecnologías Web del lado del Cliente	23
Internet Information Server	26
Microsoft Azure.....	29
Lección 2 Visión general de ASP.NET	31
Aplicaciones Web Pages.....	32
Aplicaciones Web Forms	34
Aplicaciones MVC.....	36
Características compartidas	39
Lección 3 Introducción a ASP.NET MVC	42
Modelos, Vistas y Controladores	43
Explorando una aplicación MVC.....	45
Características importantes de ASP.NET MVC	46
Módulo 2: Diseñando aplicaciones Web ASP.NET MVC	48
Lección 1 Planeación en la fase de diseño del proyecto	50
Metodologías de desarrollo de Proyectos	51
Recopilación de Requerimientos	55
Planificando el Diseño de la Base de Datos.....	58
Planificación de Aplicaciones Distribuidas	61
Planificando la Administración de Estado	64
Planificando la Globalización y Localización	68

Lección 2 Diseñando Modelos, Controladores y Vistas	70
Diseñando Modelos	71
Diseñando Controladores.....	75
Diseñando Vistas	77
Módulo 3: Desarrollando Modelos ASP.NET MVC	79
Lección 1 Creando Modelos MVC	81
Desarrollo de Modelos	82
Uso de Anotaciones de Datos de Visualización y Edición sobre las propiedades	85
Validando la entrada de usuario con Anotaciones de Datos	86
¿Qué son los Enlazadores de Modelo (Model Binders)?	88
Extensibilidad de Modelo.....	91
Creando una Anotación de Datos de Validación personalizada	92
Creando Enlazadores de Modelo personalizados	94
Lección 2 Trabajando con datos.....	96
Conectando a una base de datos	97
Conectando una aplicación Web MVC a una base de datos.....	99
Entity Framework.....	100
Agregando un Contexto de Entity Framework.....	101
LINQ to Entities	103
Utilizando LINQ to Entities	104
Acceso a datos en Modelos y Repositorios.....	105
¿Cómo separar las clases Modelo y Repositorio?	106
Módulo 4: Desarrollando Controladores ASP.NET MVC.....	108
Lección 1 Planeación en la fase de diseño del proyecto	110
Respondiendo a las peticiones del Usuario	111
Escribiendo Acciones del Controlador	113
Acciones que responden a peticiones HTTP POST	115
Tipos derivados de ActionResult	117
Acciones hijas (Child Actions).....	119
Utilizando Parámetros.....	120

Pasando información a las Vistas	121
¿Qué son las Fábricas de Controladores?	123
Creando una Fábrica personalizada de Controladores	125
Lección 2 Creando Filtros de Acción.....	127
¿Qué son los Filtros?	128
Creando y Utilizando Filtros de Acción	130
Módulo 5: Desarrollando Vistas ASP.NET MVC	132
Lección 1 Creando Vistas con sintaxis Razor	134
Agregando Vistas.....	135
Diferenciando el código del servidor y el código HTML.....	137
Características de la Sintaxis Razor	139
Enlazando Vistas a Modelos.....	141
Generando código HTML accesible.....	144
Motores de Vistas alternativos	146
Creación de Motores de Vista Personalizados.....	148
Lección 2 Utilizando HTML Helpers.....	149
Utilizando Helpers Action.....	150
Utilizando Helpers Display	152
El Helper BeginForm.....	153
Utilizando Formularios para subir archivos.....	154
Utilizando Helpers de edición	155
Utilizando Helpers de Validación	157
Codificación HTML.....	159
Creación de Helpers HTML personalizados.....	160
Lección 3 Reutilizando código en Vistas.....	162
Creando Vistas Parciales	163
Utilizando Vistas Parciales.....	165
Módulo 6: Probando, Depurando y Solucionando problemas en aplicaciones Web ASP.NET MVC	166
Lección 1 Realizando pruebas unitarias de los componentes MVC.....	168
¿Por qué realizar Pruebas Unitarias?	169

Principios del Desarrollo Guiado por Pruebas	172
Creando Componentes MVC débilmente acoplados (Loosely-Coupled)	174
Laboratorio: Creando Pruebas Unitarias para componentes MVC.....	176
Ejercicio 1: Creando Pruebas Unitarias	178
Tarea 1: Crear un proyecto de pruebas	178
Tarea 2. Configurar el proyecto de pruebas	180
Tarea 3: Preparar la aplicación Web	183
Tarea 4: Escribir las pruebas.....	184
Tarea 5: Implementar la funcionalidad para aprobar las pruebas.....	194
Tarea 6. Probar clases Modelo y de Lógica de Negocio	198
Ejercicio 2: Creando Interfaces Repositorio	204
Tarea 1. Crear un Repositorio	204
Tarea 3: Refactorizar el Controlador para utilizar el Repositorio	212
Tarea 4: Implementar el Repositorio Doble de Prueba	223
Tarea 5: Refactorizar las Pruebas Unitarias para utilizar el Repositorio de simulación.....	229
Tarea 6: Agregar pruebas adicionales	236
Resumen.....	243
Lección 2 Implementando una estrategia para el manejo de Excepciones	244
Disparando y atrapando excepciones	245
Configurando el manejo de Excepciones	250
Utilizando Visual Studio IntelliTrace en MVC.....	252
Registro de Excepciones.....	255
Monitoreo de salud en ASP.NET (ASP.NET Health Monitoring).....	257
Laboratorio: Implementando una estrategia para el manejo de Excepciones	260
Ejercicio 1: Disparando y atrapando Excepciones.....	261
Tarea 1: Disparar una excepción personalizada.....	261
Tarea 2: Atrapar excepciones con bloques Try/Catch	265
Tarea 3: Atrapar excepciones de Controladores con el método OnException	270
Tarea 4: Atrapar excepciones de Controladores con la anotación HandleError.....	275
Tarea 5: Atrapar excepciones adicionales.....	283

Resumen.....	286
--------------	-----

Acerca del curso

Este entrenamiento, describe las distintas tecnologías disponibles para desarrollar aplicaciones Web ASP.NET, enfocándose principalmente en el desarrollo de aplicaciones Web utilizando el patrón Model-View-Controller (MVC). Se compara ASP.NET MVC con ASP.NET Web Forms para que los participantes puedan decidir en qué escenarios deben o pueden ser utilizadas esas tecnologías.

El entrenamiento describe la estructura de una aplicación Web ASP.NET MVC para poder diseñar y crear Modelos, Controladores y Vistas de aplicaciones Web que accedan a bases de datos utilizando Entity Framework.

El entrenamiento incluye una introducción al desarrollo de Pruebas Unitarias para los componentes MVC y describe la forma de implementar una estrategia para el manejo de excepciones en aplicaciones Web ASP.NET MVC.

Audiencia

Este curso está diseñado para personas con un perfil de desarrolladores web y que utilizan ambientes de desarrollo individual o en equipo. Los candidatos a este curso son personas Interesadas en desarrollar aplicaciones web básicas y que desean administrar comprensivamente la presentación visual de HTML separando la interfaz de usuario, el acceso a datos y la lógica de la aplicación.

Para un buen aprovechamiento de este entrenamiento, es recomendable que los participantes cuenten con un mínimo de 3 meses de experiencia en el uso del .NET Framework, el lenguaje de programación C# y Visual Studio.

Objetivos

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

- Describir las distintas tecnologías Web de Microsoft y seleccionar la apropiada para utilizarla en el desarrollo de una aplicación determinada.
- Diseñar la arquitectura y la implementación de una aplicación Web que cumpla con un conjunto de requerimientos funcionales, requerimientos de interfaz de usuario y que se adapte a modelos de negocios.
- Crear Modelos MVC y escribir código que implemente la lógica de negocios con los métodos, propiedades y eventos del Modelo.
- Agregar Controladores a una aplicación MVC para manejar la interacción de usuario, actualizar los Modelos y seleccionar y devolver las Vistas apropiadas.
- Crear Vistas en una aplicación MVC que muestren y editen datos además de interactuar con Modelos y Controladores.

- Ejecutar pruebas unitarias y herramientas de depuración en una aplicación web con Visual Studio para encontrar y solucionar problemas.

Requerimientos

Para poder practicar los conceptos y realizar los ejercicios del curso, se recomienda contar con lo siguiente:

- Sistema Operativo Windows 8 o posteriores. Puede descargarse la versión de evaluación en <https://technet.microsoft.com/es-mx/evalcenter/hh699156.aspx>
- Microsoft WebMatrix 3. Puede descargarse gratuitamente en <http://www.microsoft.com/web/webmatrix/>
- Visual Studio 2013 o posteriores. Puede utilizarse la versión gratuita Visual Studio Community que puede descargarse en <http://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx>
- SQL Server 2012 o posteriores. Puede descargarse la versión gratuita en <http://www.microsoft.com/en-us/download/details.aspx?id=42299>

Contenido del curso

El contenido de este entrenamiento consta de 6 módulos.

Módulo 1. Explorando ASP.NET MVC

En este módulo se presenta un resumen del conjunto de tecnologías Web de Microsoft para poder seleccionar la tecnología apropiada para utilizarla en el desarrollo de una aplicación determinada. Se describen los tres modelos de programación disponibles en ASP.NET tales como Páginas Web, Web Forms y MVC. En este módulo, también se presenta una introducción a ASP.NET MVC incluyendo nuevas características y configuración.

Al finalizar este módulo, los participantes podrán:

- Describir el papel de ASP.NET en el Stack de tecnologías Web y como utilizar ASP.NET para construir aplicaciones Web.
- Describir los tres modelos de programación disponibles en ASP.NET: Web Pages, Web Forms y MVC para seleccionar el modelo apropiado para un proyecto específico.
- Distinguir entre un Modelo MVC, Controlador MVC y una Vista MVC.

Los temas que forman parte de este módulo son:

- Lección 1: Visión general de las tecnologías Web de Microsoft.
- Lección 2: Visión general de ASP.NET.
- Lección 3: Introducción a ASP.NET MVC.

Módulo 2: Diseñando aplicaciones Web ASP.NET MVC

En este módulo, se describe el proceso de diseño típico que los arquitectos deben completar cuando planean el desarrollo de una aplicación y MVC ha sido seleccionado como el modelo de programación más apropiado. Se describe el detalle de una aplicación Web MVC tal como su arquitectura, Controladores, Vistas, Modelos y Rutas.

Al finalizar este módulo, los participantes podrán:

- Planear la arquitectura de una aplicación Web MVC y considerar aspectos tales como el manejo de estado.
- Planear los Modelos, Controladores y Vistas que son requeridos para implementar un conjunto de requerimientos funcionales proporcionado.

Los temas que forman parte de este módulo son:

- Lección 1: Planeación en la fase de diseño del proyecto.
- Lección 2: Diseñando Modelos, Controladores y Vistas.

Módulo 3. Desarrollando Modelos ASP.NET MVC

En este módulo se muestra la forma de crear Modelos en una aplicación Web MVC que implemente la lógica de negocio necesaria para satisfacer los requerimientos de funcionalidad de la aplicación. Este módulo también describe la forma de implementar una conexión a una base de datos o a un almacenamiento alternativo de datos utilizando Entity Framework y LINQ.

Al finalizar este módulo, los participantes podrán:

- Agregar un Modelo a una aplicación MVC y escribir el código en el Modelo para implementar la lógica de negocios.
- Crear una nueva base de datos SQL Database en Windows Azure para almacenar los datos de la aplicación Web.

Los temas que forman parte de este módulo son:

- Lección 1: Creando modelos MVC.
- Lección 2: Trabajando con datos.

Módulo 4. Desarrollando Controladores ASP.NET MVC

En este módulo se muestra la forma de crear Controladores en una aplicación Web MVC y la forma de implementar Acciones que respondan a la entrada de los usuarios y de otros eventos. Se muestra la relación entre el Controlador y el Modelo, así como la forma de implementar Acciones del Controlador que definan las Vistas utilizadas para mostrar o editar datos. En este módulo también se

describe la forma de escribir Filtros de Acción (Action Filters) para ejecutar código antes o después de las Acciones en el Controlador. Se definen las situaciones en que los Filtros de Acción son útiles.

Al finalizar este módulo, los participantes podrán:

- Agregar un Controlador a una aplicación Web que responda a las acciones del usuario especificadas en el diseño del proyecto.
- Escribir código en Filtros de Acción que se ejecute antes o después de una Acción del Controlador.

Los temas que forman parte de este módulo son:

- Lección 1: Creando Controladores y Acciones.
- Lección 2: Creando Filtros de Acción.

Módulo 5. Desarrollando Vistas ASP.NET MVC

En este módulo se describe el papel de las Vistas en las aplicaciones Web ASP.NET MVC, así como su interacción con los Modelos y Controladores para poder mostrar datos y permitir su edición. Se describe la sintaxis de una Vista Razor que define la estructura y funcionalidad de la presentación de datos. Se describe la funcionalidad de los principales HTML Helpers, la reutilización de código mediante la definición de Vistas parciales y los Razor Helpers.

Al finalizar este módulo, los participantes podrán:

- Crear una Vista MVC y agregar código Razor para mostrar datos a los usuarios.
- Utilizar HTML Helpers en una Vista para generar el código de los controles, enlaces y otros elementos HTML.
- Reutilizar Código Razor en múltiples ubicaciones de una aplicación.

Los temas que forman parte de este módulo son:

- Lección 1: Creando Vistas con sintaxis Razor.
- Lección 2: Utilizando HTML Helpers.
- Lección 3: Reutilizando código en Vistas.

Módulo 6. Probando, Depurando y Solucionando problemas en aplicaciones Web ASP.NET MVC

En este módulo se describe la forma de realizar Pruebas Unitarias de cada uno de los componentes de una aplicación MVC y la forma de implementar una estrategia de manejo de Excepciones para la aplicación.

Se describe también la forma de mejorar la calidad de una aplicación localizando y corrigiendo errores de código y otros resultados inesperados a través de las herramientas de depuración disponibles en Visual Studio.

Al finalizar este módulo, los participantes podrán:

- Ejecutar pruebas unitarias sobre los componentes MVC tales como las clases del Modelo.
- Localizar errores potenciales en los componentes MVC.
- Construir una aplicación MVC que maneje Excepciones de una forma robusta y eficiente.

Los temas que forman parte de este módulo son:

- Lección 1: Realizando pruebas unitarias de los componentes MVC.
- Lección 2: Implementando una estrategia para el manejo de Excepciones.

Programador Jr. de aplicaciones ASP.NET MVC

Módulo 1: Explorando ASP.NET MVC

Acerca del módulo

Microsoft ASP.NET MVC y las otras tecnologías Web del .NET Framework ayudan a crear y alojar aplicaciones web dinámicas, potentes y extensibles.

ASP.NET MVC soporta desarrollo ágil, basado en pruebas y los últimos estándares Web tal como HTML5.

Para crear aplicaciones Web robustas, es necesario estar familiarizados con las tecnologías y los productos de la pila Web de Microsoft.

Necesitamos también saber cómo funcionan las Aplicaciones ASP.NET con IIS, Visual Studio, SQL Server, Microsoft Azure y Microsoft Azure SQL DataBase para ofrecer páginas Web atractivas a los visitantes del sitio.

Para elegir un lenguaje de programación que mejor se ajuste a un conjunto de requisitos de negocio, necesitamos saber cómo las aplicaciones del Modelo-Vista-Controlador (MVC) difieren de los otros modelos de programación ASP.NET: Páginas Web y Formularios Web.

Objetivos

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

- Describir el papel de ASP.NET en la pila de tecnologías web de Microsoft y cómo utilizar ASP.NET para construir aplicaciones Web.
- Describir los tres modelos de programación disponibles en ASP.NET: Páginas Web, Formularios Web y MVC.
- Seleccionar un modelo apropiado para un proyecto determinado.
- Distinguir entre un Modelo MVC, un Controlador MVC y una Vista MVC.

Los temas que se cubren en este módulo son:

- Lección 1: Visión general de las tecnologías Web Microsoft
- Lección 2: Visión general de ASP.NET
- Lección 3: Introducción a ASP.NET MVC

Lección 1

Visión general de las tecnologías Web Microsoft

Antes de empezar a trabajar con ASP.NET MVC, es necesario conocer donde se ubica Microsoft ASP.NET en el contexto de la pila de tecnologías Web de Microsoft. Debemos conocer cómo se alojan los sitios Web ASP.NET en Internet Information Server (IIS) o Microsoft Azure. Debemos conocer la forma en que las aplicaciones Web ejecutan código del lado del servidor en los servidores Web y el código del lado del cliente en los navegadores Web para ayudar a proporcionar contenido atractivo y rico en interfaz de usuario.

Objetivos de la lección

Al finalizar esta lección, seremos capaces de:

- Proporcionar una visión general de las tecnologías Web de Microsoft.
- Proporcionar una visión general de ASP.NET.
- Proporcionar una visión general de las tecnologías Web del lado del cliente, tales como, AJAX y bibliotecas JavaScript.
- Describir el papel de IIS en la pila de tecnologías Web de Microsoft.
- Describir el papel de Microsoft Azure en la pila de tecnologías Web de Microsoft.

Introducción a las tecnologías Web de Microsoft

Microsoft ofrece una amplia gama de tecnologías que se pueden utilizar para crear aplicaciones Web enriquecidas y publicarlas en Intranets y en Internet.

Además de la publicación de las aplicaciones Web, podemos utilizar estas tecnologías para desarrollar y alojar páginas Web que ejecuten código tanto en el servidor Web como en el navegador Web del usuario.

Las tecnologías que forman parte de la pila Web de Microsoft pueden agruparse en 3 categorías: herramientas de desarrollo, tecnologías de hospedaje y tecnologías de ejecución de código.

Herramientas de desarrollo

Podemos crear sitios web simples con texto e imágenes usando un editor de texto como el Bloc de notas, sin embargo, la mayoría de los sitios Web requieren que sean realizadas acciones complejas del lado del servidor, tales como operaciones de base de datos, envío de correo electrónico, cálculos complejos o procesamiento de gráficos. Para crear ese tipo de sitios web complejos, altamente funcionales y atractivos de forma rápida y sencilla, Microsoft ofrece herramientas de desarrollo:

- **WebMatrix.** Podemos utilizar WebMatrix para crear páginas HTML estáticas y páginas dinámicas con ASP.NET, PHP y Node.js. WebMatrix es una herramienta de desarrollo gratuito que puede instalarse a través del **Microsoft Web Platform Installer** el cual puede descargarse desde el sitio Web de Microsoft. WebMatrix permite desarrollar sitios web personalizados basados en aplicaciones Web populares como Orchard, Umbraco CMS y WordPress. Orchard está desarrollado con ASP.NET mientras que WordPress está escrito con PHP. Con WebMatrix, podemos crear aplicaciones ASP.NET Web Pages pero no podemos crear aplicaciones ASP.NET Web Forms o aplicaciones ASP.NET MVC.
- **Microsoft Visual Studio.** Podemos utilizar Visual Studio, un ambiente de desarrollo integrado o IDE para crear aplicaciones personalizadas basadas en las tecnologías de Microsoft, independientemente de si estas aplicaciones se ejecutan en la web, en equipos de escritorio, en dispositivos móviles o mediante el uso de Servicios de nube de Microsoft. Visual Studio ofrece muchas facilidades para diseñar, codificar y depurar cualquier aplicación web ASP.NET, incluyendo aplicaciones ASP.NET MVC.
- **Microsoft Visual Studio Express for Web.** Podemos utilizar Visual Studio Express para Web para crear aplicaciones Web Forms o aplicaciones Web MVC. Esta es una herramienta gratuita que no incluye todas las características de las ediciones de Visual Studio, sin embargo, se puede utilizar para crear sitios web MVC completamente funcionales.

- **Microsoft Visual Studio Community.** Visual Studio Community es una herramienta gratuita que incluye toda la funcionalidad de Visual Studio Professional, diseñada y optimizada para desarrolladores individuales, estudiantes, contribuidores de Open Source y equipos pequeños de trabajo. Se puede utilizar Visual Studio Community para crear aplicaciones gratuitas o pagadas. Visual Studio Community reemplaza a la versión Express de Visual Studio.

Tecnologías de hospedaje

Independientemente de la herramienta que utilicemos para construir una aplicación Web, debemos utilizar un servidor web para alojar la aplicación.

Cuando los usuarios visitan nuestro sitio web, el servidor responde generando código HTML y lo entrega al navegador del usuario para su visualización.

El servidor de hospedaje puede consultar una base de datos antes de que genere el código HTML y puede realizar otras acciones, como el envío de mensajes de correo electrónico o guardar los archivos subidos por el usuario.

Es necesario probar la funcionalidad de las acciones de los usuarios en un servidor Web.

Al crear una aplicación Web utilizando Visual Studio, es posible utilizar el servidor Web IIS Express incorporado a Visual Studio para ejecutar la aplicación, sin embargo, IIS Express no está diseñado para hospedar aplicaciones Web en producción, por lo tanto, cuando se termina la construcción de la aplicación Web y está lista para que los usuarios accedan a ella en una Intranet o en Internet, se debe utilizar un servidor Web completamente funcional.

De las tecnologías de hospedaje del Stack Web de Microsoft tenemos:

- **Microsoft Internet Information Server (IIS).** IIS es una tecnología avanzada de alojamiento web. Podemos instalar servidores IIS en nuestra red local o en la red perimetral o utilizar servidores IIS alojados por un Proveedor de Servicios de Internet (ISP). IIS puede alojar cualquier sitio web ASP.NET, PHP, o Node.js. Podemos escalar IIS para alojar sitios web grandes y concurridos mediante la configuración de granjas de servidores que contengan múltiples servidores IIS, todos dando servicio al mismo contenido.
- **Microsoft Azure.** Microsoft Azure es una plataforma en la nube que ofrece servicios sobre demanda para construir, implementar, hospedar y administrar aplicaciones Web a través de centros de datos administrados por Microsoft. Cuando utilizamos los servicios de Microsoft Azure, debemos pagar sólo por los recursos que el sitio Web consume. Además, no tenemos por qué preocuparnos por la construcción de una infraestructura escalable ya que Microsoft Azure agrega recursos de forma automática a medida que el sitio Web crece.

Es importante mencionar que hay muchas otras tecnologías que no son de Microsoft y que podemos utilizar para alojar sitios web. El más popular, por ejemplo, es Apache. Apache se ejecuta con mayor frecuencia en los servidores Linux y a menudo se combina con MySQL para gestionar bases de datos y PHP como un Framework Web basado en servidor. Esta configuración se conoce a menudo como la pila Web Linux-Apache-MySQL-PHP (LAMP).

Microsoft Azure y IIS pueden alojar sitios Web escritos en ASP.NET, PHP, o Node.js. Los servidores Web Apache se pueden utilizar para hospedar una serie de tecnologías Web del lado del servidor, incluyendo PHP, Node.js y Python, pero los servidores Web Apache no se pueden utilizar para alojar sitios Web ASP.NET aunque de manera NO oficial podemos encontrar foros donde los usuarios comentan que han podido hospedar aplicaciones ASP.NET sobre Apache modificando algunos archivos de configuración de Apache (Apache.exe.config y httpd.default.conf). Para más información, puede consultarse el siguiente enlace: <http://forums.asp.net/t/962964.aspx>.

La mayoría de los sitios Web requieren alojar una base de datos para gestionar datos, como los detalles del producto, información del usuario o temas de discusión de un foro. Podemos seleccionar alguna de las siguientes tecnologías Microsoft para administrar los datos:

- **Microsoft SQL Server.** Microsoft SQL Server es una tecnología de Microsoft para alojar y administrar datos. SQL Server es un servidor de base de datos de alta calidad que se puede utilizar para almacenar cualquier base de datos desde las más simples hasta las más complejas. SQL Server puede escalar para soportar bases de datos de gran tamaño y un gran número de usuarios. Podemos construir clústeres de SQL Server para asegurar la mejor disponibilidad y fiabilidad. Muchas de las organizaciones más grandes del mundo confían en SQL Server para alojar datos.
- **Microsoft Azure SQL Database.** Base de datos SQL de Microsoft Azure es una plataforma de base de datos en la nube y una parte de Microsoft Azure. Usando SQL Database, podemos desplegar una base de datos y pagar sólo por las operaciones y el almacenamiento de los datos que utilicemos. No debemos preocuparnos por la administración de nuestra infraestructura de base de datos ya que la base de datos escala de forma automática según crezca nuestro sitio Web.

Tecnologías de ejecución de código

El código que escribimos en una aplicación Web con una herramienta de desarrollo, debe ejecutarse en uno de dos lugares:

- **En el servidor Web.** El código que se ejecuta en el servidor Web tiene acceso completo a los recursos del servidor Web y a cualquier base de datos asociadas a él. Puede acceder a la base de datos con rapidez, enviar mensajes de correo electrónico y generar páginas Web.

- **En el navegador Web del Usuario.** El código que se ejecuta en el navegador Web del usuario responde rápidamente a las acciones del usuario, tales como clics del ratón, pero está más limitado en lo que puede hacer sin interactuar con el servidor Web.

Es necesario utilizar diferentes tecnologías para ejecutar código en el servidor y el código en el cliente.

Ejecución del lado del Servidor

Microsoft ASP.NET es un entorno Web del lado del servidor que ejecuta código .NET del lado del servidor, código que podemos escribir en Visual Studio o WebMatrix. El código puede acceder a una base de datos, generar las páginas HTML y entregarlas al navegador Web. El modelo de programación MVC es una parte de ASP.NET. Otras tecnologías No Microsoft del lado del servidor son PHP y Node.js.

Ejecución del lado del Cliente

La mayoría de los navegadores Web pueden ejecutar código escrito en el lenguaje JavaScript. Este código es enviado al navegador como texto dentro de la página HTML generada o en un archivo .js separado. Debido a que JavaScript se ejecuta de forma local en el navegador, puede responder rápidamente a las acciones del usuario tales como al hacer clic, apuntar o arrastrar el puntero del mouse.

Muchas bibliotecas JavaScript están disponibles para acelerar el desarrollo del código del lado del cliente. Por ejemplo, la popular biblioteca jQuery hace que sea fácil de acceder a elementos de la página y manipularlos mediante el cambio de su estilo o contenido. Al utilizar estas bibliotecas, podemos escribir funciones con pocas líneas de código que de otro modo requerirían cientos de líneas de nuestro propio código JavaScript.

Las aplicaciones ASP.NET también pueden utilizar la tecnología **Asynchronous JavaScript and XML (AJAX)** en la computadora del cliente. Podemos utilizar AJAX para actualizar una pequeña sección de una página HTML en lugar de volver a cargar la totalidad de la página desde el servidor. Las actualizaciones parciales de página ayudan a crear páginas Web más responsivas y atractivas.

Visión general de ASP.NET

Podemos utilizar ASP.NET para desarrollar sitios Web con acceso a bases de datos, altamente funcionales, dinámicos y escalables que utilicen código en el lado del cliente y código en el lado del servidor. Se pueden crear diferentes tipos de sitios Web con ASP.NET, por ejemplo, portales Web, sitios de compras en línea, blogs y wikis entre otros.

Modelos de programación

Cuando utilizamos ASP.NET para construir una aplicación, no estamos limitados a un solo estilo de programación, por el contrario, podemos elegir entre tres modelos de programación diferentes. Cada modelo de programación tiene una estructura particular en el entorno de desarrollo y almacena el código en diferentes lugares de la jerarquía Web.

- **Web Pages.** El modelo de programación ASP.NET Web Pages es un modelo de programación simple y fácil de aprender y es adecuado para sitios simples de acceso a datos.

Crear una aplicación Web con este modelo es tan sencillo como crear una página HTML y agregar el código que deseamos que el motor ASP.NET ejecute del lado del servidor para procesar los datos de una base de datos, responder a un envío (post) de un formulario o realizar otras acciones.

El código del lado del servidor puede ser escrito con el lenguaje de programación C# o con VB.NET. Si el código está en C#, el nombre del archivo debe tener la extensión CSHTML y si el código es Visual Basic, la extensión debe ser VBHTML.

El modelo de desarrollo Web Pages es similar al de las páginas ASP anterior a .NET donde se mezclaba en un solo archivo código HTML con código para ejecutarse del lado del servidor con Visual Basic Scripting Edition.

- **Web Forms.** Cuando construimos un sitio Web mediante el uso de ASP.NET Web Forms, empleamos un modelo de programación con controles ricos del lado del servidor y un ciclo de vida de página similar al desarrollo de aplicaciones de escritorio. Los controles integrados incluyen botones, cuadros de texto y Grids para mostrar datos en forma tabular. También podemos agregar controles de terceros o crear controles personalizados. Para responder a las acciones del usuario, podemos asociar a los controles del lado del servidor, controladores de eventos conteniendo código. Por ejemplo, para responder a un clic en un botón podríamos escribir código en el controlador del evento **Click** del botón.

- **MVC.** Cuando construimos un sitio mediante ASP.NET MVC, separamos el código del lado del servidor en tres partes:
 - **Modelo:** Un modelo MVC define un conjunto de clases que representan los tipos de objetos que la aplicación Web administra. Por ejemplo, el modelo de un sitio de comercio electrónico podría incluir una clase modelo Producto que defina propiedades tales como la descripción, número de catálogo, precio y otros. Los Modelos suelen incluir lógica de acceso a datos que lee datos desde una base de datos y escribe datos a esa base de datos.
 - **Vistas:** Una vista MVC es un componente que construye las páginas Web que conforman la interfaz de usuario de la aplicación Web. Los controladores suelen pasar una instancia de una clase del Modelo a una Vista. La Vista muestra las propiedades de la clase Modelo. Por ejemplo, si el controlador pasa un objeto Producto, la vista puede mostrar el nombre del producto, una imagen y su precio.
 - **Controladores:** Un controlador MVC es una clase que se encarga de la interacción del usuario, crea y modifica clases del Modelo y selecciona las Vistas correspondientes. Por ejemplo, cuando un usuario solicita detalles completos acerca de un producto en particular, el controlador crea una nueva instancia de la clase modelo Producto y la pasa a la Vista de detalles para que la muestre al usuario.

Esta separación de código de Modelo, Vista y Controlador garantiza que las aplicaciones MVC tengan una estructura lógica, incluso para los sitios más complejos. También mejora la capacidad de prueba de la aplicación. En última instancia, ASP.NET MVC proporciona un mayor control sobre el código HTML en comparación con Web Pages y Web Forms.



¿Cuál de los tres modelos de programación proporciona el máximo control sobre el código HTML y JavaScript que se envía al navegador?

MVC y Web Pages proporcionan un alto nivel de control sobre el código HTML y JavaScript mediante el uso de las Vistas.

La API ASP.NET

Sea cual sea el modelo de programación que elijamos, tenemos acceso a las clases de la API ASP.NET. Estas clases están incluidas en el .NET Framework en espacios de nombre dentro del espacio de