

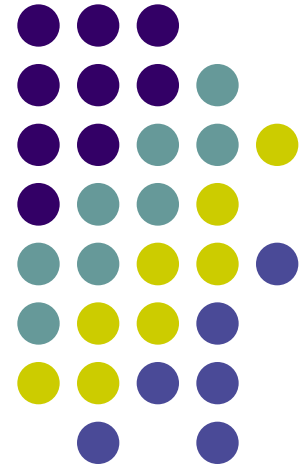
Desarrollo de Aplicaciones Web

# Desarrollo Web en Entornos de Servidor



Tema 2

## Inserción de controles de servidor en páginas Web



***Vicente J. Aracil Miralles***

[vj.aracilmiralles@edu.gva.es](mailto:vj.aracilmiralles@edu.gva.es)

03/09/2024

## Tema 2

# Inserción de controles de servidor en páginas Web



## Objetivos

- Aprender a generar código de forma dinámica para ser mostrado en el cliente Web
- Conocer la sintaxis y las etiquetas de los controles de servidor insertados en páginas Web ejecutadas en el servidor Web
- Conocer los aspectos declarativos e interactivos de las páginas Web basadas en lenguaje de marcas y ejecutadas por el servidor Web
- Comprender las técnicas y herramientas de desarrollo de la interfaz de usuario
- Reconocer la importancia del desarrollo visual de la capa de presentación de una aplicación Web



## Tema 2

### Inserción de controles de servidor

#### Contenidos

- 2.1 Páginas web de ASP.NET
- 2.2 Sintaxis general de las páginas Web de ASP.NET
- 2.3 Controles de servidor Web
- 2.4 Acceso a documentación en línea sobre ASP.NET



### Concepto y características básicas

- Las Páginas de ASP.NET o también denominadas Web Forms:
  - **Definen la interfaz de usuario** utilizada para presentar información al usuario en cualquier navegador web (capa de presentación)
  - **Implementan la lógica del procesamiento asociado** que se ejecuta en servidor Web y que se especifica mediante código lógico (capa de lógica del negocio)
- Características básicas
  - Se basan en la tecnología ASP.NET de Microsoft
  - La ejecución del código que contiene una página Web de ASP.NET en el servidor Web, **genera de forma dinámica una página HTML de respuesta** que se envía al cliente como resultado del procesamiento realizado en el servidor Web
  - Son compatibles con cualquier navegador Web que se ejecute en cualquier dispositivo
  - Admiten cualquier lenguaje de programación compatible con *Common Language Runtime* (CLR) de .NET para expresar la capa de lógica del negocio
  - Pueden utilizar todas las funcionalidades del modelo de desarrollo .NET
  - Son flexibles, al poder incorporar controles creados por el desarrollador y otros fabricantes



### Programación de páginas de ASP.NET

- La programación de páginas Web de ASP.NET se divide en dos partes:
  - **El componente visual.** Se compone de código de marcado que puede ser estático basado en HTML y ejecutado por el navegador, o basado en controles de ASP.NET y ejecutado en el servidor, o también puede combinar ambos.
    - Una página Web de ASP.NET o Web Form está compuesto por un archivo de texto (.aspx) que funciona como un contenedor de los controles Web que se desea formen parte de la interfaz de usuario
  - **El componente lógico.** Se compone de código procedural escrito en cualquier lenguaje que soporte .NET Framework y que es ejecutado por el servidor Web. El código procedural expresa la lógica del negocio de la página Web de ASP.NET
    - Puede residir en un bloque de código de servidor en la propia página de ASP.NET o en una clase independiente. Si el código lógico se define en una clase independiente, residirá en un archivo de texto denominado archivo de código subyacente (.aspx.cs)

Existe una separación de código de la parte visual que define la interfaz de usuario de la página Web y del código de la parte lógica que interactúa con ella

## 2.2 Sintaxis general de las páginas Web de ASP.NET



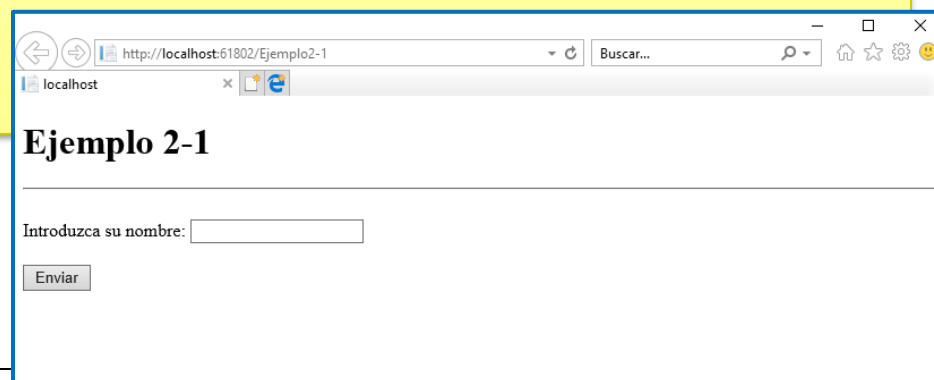
Ejemplo2-1.aspx

### Ejemplo

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Ejemplo2-1.aspx.cs"
    Inherits="EjemplosTeoriaT2.Ejemplo2_1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h1>Ejemplo 2-1</h1> <hr />
            <br />
            <asp:Label ID="Label1" runat="server" Text="Introduzca su nombre: "></asp:Label>
            <asp:TextBox ID="txtNombre" runat="server"></asp:TextBox> <br /> <br />
            <asp:Button ID="btnEnviar" runat="server" Text="Enviar"/> <br /> <br />
            <asp:Label ID="lblTexto" runat="server" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```



## 2.2 Sintaxis general de las páginas Web de ASP.NET



### Elementos específicos que forman una página de ASP.NET

- La estructura de una página de ASP.NET o Web Form es similar a la de las páginas de HTML, aunque contiene **elementos específicos** que son reconocidos y procesados por el servidor Web cuando se ejecuta la página:
  - La directiva **@ Page** u otras directivas opcionales según convenga
  - Un elemento **<form>** que es procesado por el servidor Web
  - Diversos **controles de servidor Web** representados por código de marcado que reconoce ASP.NET. Los nombres de sus etiquetas de marcado comienzan con el prefijo **asp**:
    - El atributo **runat** establecido con el valor “**server**” hace que **el procesamiento de ese control o elemento se realice en el servidor Web**, y no en el cliente
  - Diversos **elementos estáticos de HTML** que serán interpretados solo por el navegador
  - Las páginas de ASP.NET **pueden incluir bloques de código lógico** de servidor que representa la lógica que interactúa con la página para generar la respuesta. Lo habitual es que se especifique en un archivo separado, denominado archivo de código subyacente
- La extensión de nombre de archivo **.aspx** hace que ASP.NET procese la página
  - Las páginas con extensión **.htm** o **.html** serán interpretadas solo por el navegador Web



### Directivas

- Las páginas de ASP.NET contienen directivas que especifican las propiedades y la configuración de la página
- La directiva más utilizada es **@ Page**, que permite especificar diversas opciones de configuración para la página:
  - **Language**. Establece el lenguaje de programación para el código lógico
  - **AutoEventWireup**. Indica si los eventos de la página están autoconectados
  - **CodeBehind**. Especifica la ruta de acceso y el nombre del archivo de código subyacente al que se hace referencia la página de ASP.NET
  - **Inherits**. Establece la clase que define el código lógico en el archivo de código subyacente mediante CodeBehind
  - Otras opciones de configuración de la directiva @ Page son: las opciones de traza y depuración, la definición de páginas maestras asociadas, etc.

Si la directiva @ Page no incluye una configuración específica, la configuración se hereda del archivo de configuración de la aplicación Web, denominado Web.config





### Directivas

- Otras directivas comunes
  - **@ Import.** Esta directiva permite especificar los espacios de nombres a los que se desea hacer referencia en el código
  - **@ OutputCache.** Esta directiva permite especificar los parámetros del almacenamiento de la página de ASP.NET en la memoria caché
  - **@ Implements.** Esta directiva permite especificar que la página debe implementar una interfaz .NET
  - **@ Register.** Esta directiva permite registrar controles de usuarios para su uso en la página. La directiva @ Register declara el prefijo de la etiqueta del control y la ubicación de su ensamblado.
- Algunos otros tipos de archivos que pueden aparecer en una aplicación basada en ASP.NET utilizan otras directivas:
  - Las páginas maestras de ASP.NET utilizan la directiva **@ Master**
  - Los controles de usuario de ASP.NET utilizan la directiva **@ Control**

## 2.2 Sintaxis general de las páginas Web de ASP.NET



### Elemento <form>

- Una página de ASP.NET incluye controles que permiten a los usuarios interactuar con la página y enviarla al servidor Web, por lo que necesariamente incluye un elemento **<form>** que debe cumplir ciertas reglas:
  - La página sólo puede contener un elemento <form>
  - El elemento <form> **debe contener el atributo *runat* establecido con el valor “server”**. Este atributo hace referencia a que el procesamiento del formulario se realizará en el servidor Web, y no en el cliente

```
<form id="form1" runat="server">
```

- Los controles Web de servidor que pueden realizar devolución de datos deben estar dentro del elemento <form>.
- La etiqueta <form> no debe contener ningún atributo *action*
  - ASP.NET establece estos atributos dinámicamente cuando se procesa la página y reemplaza cualquier configuración que se pueda establecer, porque el destino del procesamiento del formulario está predefinido

## 2.2 Sintaxis general de las páginas Web de ASP.NET



### Controles de servidor Web

- Las páginas de ASP.NET incluyen **controles de servidor Web**
  - Permiten al usuario interactuar con la página: botones, cuadros de texto, listas desplegables, etc.
  - Se procesan en el servidor Web, lo que permite utilizar código lógico de servidor para establecer sus propiedades y controlar sus eventos
  - Las páginas de ASP.NET pueden incluir dos tipos de **controles de servidor Web**:

- **Controles de servidor de ASP.NET.** Utilizan la sintaxis de ASP.NET

```
<asp:TextBox ID="txtNombre" runat="server"></asp:TextBox>  
<asp:Button ID="btnEnviar" runat="server" Text="Enviar"/>
```

- **Controles de servidor basados en HTML.** Son elementos ordinarios de HTML que actúan como controles de servidor cuando pasan a ser procesados en el servidor Web, en lugar de en el cliente. Para ello, basta con agregar el atributo *runat="server"* y un atributo *id* al elemento HTML

```
<form id="form1" runat="server">
```

## 2.2 Sintaxis general de las páginas Web de ASP.NET



### Elementos estáticos de HTML

- Además de controles de servidor, las páginas de ASP.NET incluyen **elementos estáticos de HTML** que son ejecutados por el navegador Web
  - Expresan la parte estática de la interfaz de usuario que permite establecer la presentación visual
  - Es el caso de la etiqueta `<h1>` en el ejemplo anterior (Ejemplo2-1.aspx) que se será ejecutada en el navegador (cliente). No tendrá ningún protagonismo desde el punto de vista del procesamiento en el servidor.

```
<h1>Ejemplo 2-1</h1> <hr />
```

## 2.2 Sintaxis general de las páginas Web de ASP.NET



### Proceso de desarrollo de una Página Web de ASP.NET

- Tareas de desarrollo de una Página de ASP.NET
  - 1) **Crear una página de ASP.NET o Web Form** en el sitio Web que se está desarrollando
  - 2) **Agregar en el Web Form los controles de servidor Web y los elementos HTML** adecuados para definir la interfaz de usuario de la página
    - La interfaz de usuario de las páginas de ASP.NET puede realizarse sin necesidad de escribir el código de marcado directamente, empleando la vista de diseño visual del Web Form
      - Al insertar un control sobre la página desde cuadro de herramientas de Microsoft Visual Studio, se incluirá el código de marcado correspondiente de manera automática, pudiendo entonces modificar las propiedades del control directamente sobre el código de marcado o bien desde la ventana de propiedades del control Web insertado
  - 3) **Puede añadirse código lógico de procesamiento en el servidor** empleado el archivo de código subyacente asociado

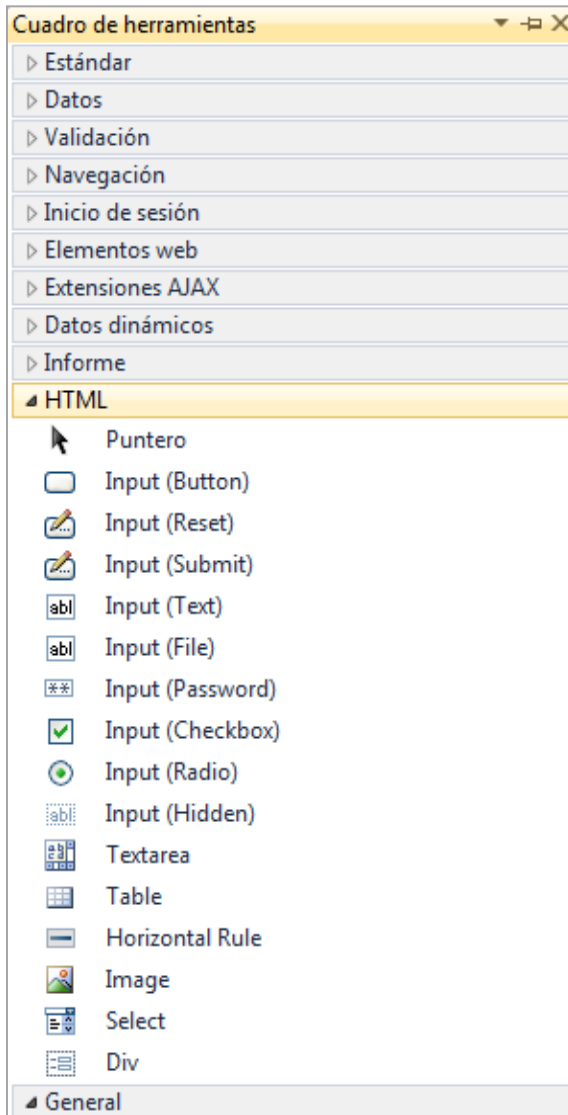


### Generalidades

- **Controles de servidor Web**. Definición:
  - Son **objetos contenidos en las páginas de ASP.NET que se ejecutan en el lado del servidor** y que representan el código de marcado que se enviará al cliente (navegador Web) para su presentación visual
- Tipos:
  - **Controles de servidor basados en HTML**. Son elementos ordinarios de HTML que se convierten en accesibles desde en el código de servidor
  - **Controles de servidor de ASP.NET**. Controles de servidor enriquecidos, porque incorporan el mayor nivel de funcionalidad incorporada. Incluyen:
    - Controles simples de formulario, como: botones, cuadros de texto, etiquetas, etc.
    - Controles avanzados con fines específicos, como: control calendario, control menú, control de vista de árbol, controles de datos, etc.
  - **Controles de validación**. Controles que facilitan la validación de los valores de la información introducida por los usuarios en los controles de entrada de datos
  - **Controles de usuario**. Controles personalizados creados por el desarrollador que pueden ser reutilizados en otras páginas de ASP.NET



### Controles de servidor basados en HTML



- Sintaxis de HTML
- Al ser convertidos en controles de servidor, quedan expuestos al código lógico de procesamiento en el servidor

```
<input type="text" id="txtName"
      runat="server" />
```

- Existen en el espacio de nombres  
**System.Web.UI.HtmlControls**



### Controles de servidor basados en HTML

- Los elementos de HTML ordinarios en una página Web ASP.NET no están disponibles para el procesamiento del servidor
  - Se tratan como texto opaco y se pasan al navegador para ser interpretados por este
- Si un elemento HTML se convierte en control de servidor, entonces queda expuesto al servidor Web y, por tanto, es accesible desde el código lógico de procesamiento en el servidor

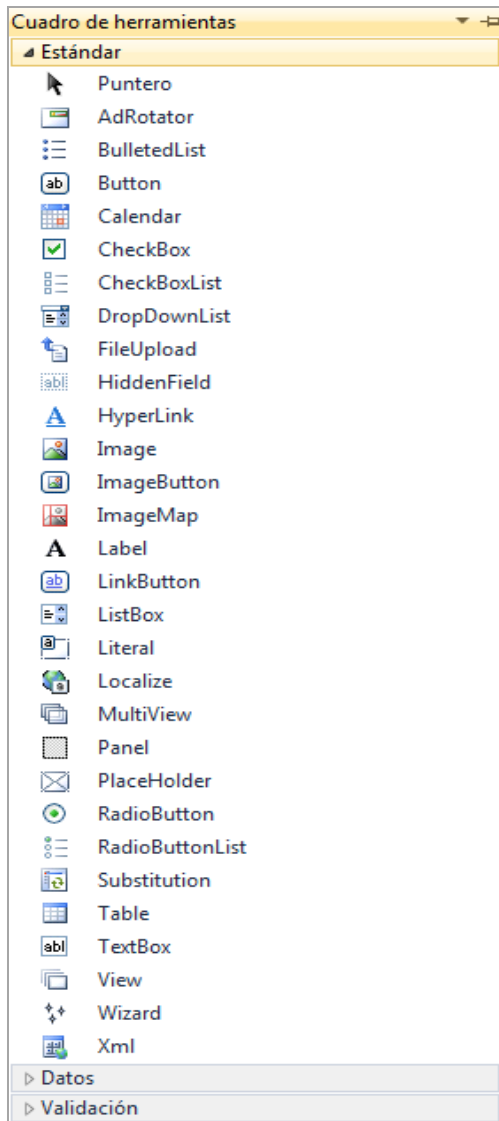
```
<input type="text" id="txtName" runat="server" />
```

- Cualquier elemento HTML estático de una página de ASP.NET se puede convertir en control de servidor basado en HTML agregando el atributo `runat="server"`
  - Solo suelen convertirse en control de servidor aquellos elementos HTML a los que se necesita acceder desde el código lógico de procesamiento en el servidor





### Controles de servidor de ASP.NET



- Controles específicos de ASP.NET

```
<asp:TextBox id="TextBox1" runat="server">  
    Text_to_Display  
</asp:TextBox>
```

- El código de marcado HTML que se envía al cliente es generado dinámicamente para cada control

```
<input name="TextBox1" type="text"  
    value="Text_to_Display" Id="TextBox1"/>
```

- Existen en el espacio de nombres  
**System.Web.UI.WebControls**



### Controles de servidor de ASP.NET

- Son un conjunto de controles de servidor Web enriquecidos, que están diseñados de manera específica para la tecnología ASP.NET

```
<asp:TextBox id="TextBox1" runat="server">  
    Text_to_Display  
</asp:TextBox>
```

- Ofrecen funcionalidades adicionales, tales como:
  - Un **modelo de objetos enriquecido** en relación con el que ofrecen los controles de servidor basados en HTML, lo que les proporciona mayor funcionalidad y prestaciones
  - Los atributos de las etiquetas de los controles de servidor de ASP.NET son propiedades que son específicas de ese tipo de control
  - Para algunos controles, la capacidad de especificar si un evento del control provoca un envío inmediato al servidor o cuando se envía la página
  - Detección automática del navegador. Se presenta en la página de respuesta de una forma u otra según el navegador empleado y de la configuración del control
  - Capacidad de definir el propio diseño del control, mediante plantillas (*templates*)



### Controles de servidor de ASP.NET

- **Algunos de los controles de servidor de ASP.NET**
  - **TextBox.** Representa un cuadro de texto
  - **Button.** Representa un botón que envía el contenido de un formulario al servidor
  - **CheckBox.** Representa una casilla de verificación
  - **CheckBoxList.** Lista de selección múltiple de elementos CheckBox
  - **DropDownList.** Lista desplegable (cuadro combinado)
  - **ListBox.** Cuadro de lista que permite la selección múltiple
  - **HiperLink.** Representa un elemento de hipervínculo a otra página
  - **Image.** Este control representa una imagen
  - **ImageButton.** Representa una imagen que actúa como botón
  - **Label.** Etiqueta de texto
  - **LinkButton.** Botón con una apariencia similar a los enlaces
  - **RadioButton.** Representa un botón de opción
  - **RadioButtonList.** Lista selección de elementos RadioButton

## 2.3 Controles de servidor Web



### Uso de controles de servidor basados en HTML o controles de servidor de ASP.NET

- **Lo más habitual es emplear controles de servidor de ASP.NET**
  - Porque incorporan un mayor nivel de funcionalidad y prestaciones, los que facilita el desarrollo de las páginas y aplicaciones Web interactivas basadas en ASP.NET con procesamiento en el servidor Web

Usar controles de servidor basados en HTML, si:	Utilizar controles de servidor de ASP.NET, si:
Se prefiere un modelo de objetos HTML	Se prefiere un modelo de objetos basado en el modelo de desarrollo .NET
Se desea agregar funcionalidades de página ASP.NET Web a páginas HTML existentes	Se está construyendo una página que puede ser utilizada por varios navegadores
El control interactúa con scripts de cliente y servidor	Se necesitan prestaciones específicas sobre los controles habituales o bien, controles que incorporan funcionalidades avanzadas



### Mantenimiento automático del estado de la página

- Estado de la página. Definición:
  - Conjunto de los valores de todos los controles de servidor insertados en una página de ASP.NET, tanto los de ASP.NET como los basados en HTML
- **Cuando una página de ASP.NET realiza una acción de ida y vuelta (Post-Back) entre el cliente y el servidor Web, el estado de la página se mantiene**
  - Así, los valores de los controles de servidor insertados en la página se conservan, de forma automática, en la página después de que sean enviados desde el cliente al servidor Web a través del formulario de la página de ASP.NET (Web Form)
    - Lo que facilita el desarrollo porque los valores de los controles se encuentran en todo momento en el servidor Web y, por tanto, son accesibles desde el código lógico de procesamiento en el servidor
  - La causa es que las páginas de ASP.NET se procesan a sí mismas. Es decir, que el destino del procesamiento del formulario que constituye una página de ASP.NET (WebForm) es la misma página, a no ser que se establezca lo contrario

El mantenimiento automático del estado de la página es una característica de las páginas de ASP.NET que afecta a todos los controles de servidor insertados en la página

## 2.4 Acceso a la documentación en línea sobre ASP.NET

### Sitio oficial para desarrolladores Microsoft Learn

The screenshot displays the Microsoft Learn website. The top navigation bar includes links for 'Microsoft', 'Learn', 'Documentación', 'Cursos', 'Certificaciones', 'Q&A', 'Ejemplos de código', and 'Más'. A search bar and a 'Iniciar sesión' link are also present. The main header for the ASP.NET section includes 'ASP.NET', 'Idiomas', 'Cargas de trabajo', 'API existentes', and 'Recursos'. A 'Descarga de .NET' button is located in the top right corner of the main content area.

### Documentación de ASP.NET

Aprenda a usar ASP.NET Core para crear aplicaciones y servicios web rápidos, seguros, multiplataforma y basados en la nube. Consulte los tutoriales, el código de ejemplo, los aspectos básicos, la referencia de API y mucho más.

- INTRODUCCIÓN: Creación de una aplicación de ASP.NET Core en...
- INFORMACIÓN...: Información general de ASP.NET Core
- DESCARGAR: Descarga de .NET
- NOVEDADES: Novedades en la documentación de ASP.NET Core
- INTRODUCCIÓN: Creación de su primera UI web
- INTRODUCCIÓN: Creación de su primera API web
- INTRODUCCIÓN: Creación de su primera aplicación web en tiempo real
- INFORMACIÓN...: Documentación de ASP.NET 4.x

### Desarrollo de aplicaciones ASP.NET Core

Elección de aplicaciones web interactivas, API web, aplicaciones con patrón de MVC, aplicaciones en tiempo real y mucho más

- Aplicaciones Blazor interactivas del lado cliente: Desarrollo con componentes de UI reutilizables que pueden aprovechar WebAssembly para obtener un rendimiento casi nativo
- Aplicaciones de API HTTP: Desarrollo de servicios HTTP con ASP.NET Core
  - Creación de una API web mínima con ASP.NET Core
  - Creación de una API web con controladores de
- UI web centrada en páginas con Razor Pages: Desarrollo de aplicaciones web centradas en páginas con una separación de intereses nítida
  - Creación de su primera aplicación web de



1) Acceder a: <https://learn.microsoft.com/es-es/>

2) Seleccionar la opción **Documentación** y el enlace **ASP.NET**

- La opción **Documentación de ASP.NET 4.x** proporciona información técnica de desarrollo sobre ASP.NET WebForms
- La opción **Información general de ASP.NET Core** proporciona información técnica de desarrollo sobre ASP.NET Core

Para acceder directamente: <https://learn.microsoft.com/es-es/aspnet/core/?view=aspnetcore-6.0>

## 2.4 Acceso a la documentación en línea sobre ASP.NET

### Sitio oficial de ASP.NET de Microsoft



Para acceder, visitar la dirección Web: <https://dotnet.microsoft.com/es-es/apps/aspnet>

- Proporciona una información más general, así como sobre de las últimas novedades y tendencias que incorpora la tecnología de desarrollo ASP.NET