

# Actividad Práctica: Desarrollo de Aplicación Web con ASP .NET Core

## Objetivo

Aplicar los conocimientos adquiridos sobre ASP .NET Core para desarrollar una aplicación web dinámica que incluya el uso de Razor Pages y la persistencia de datos mediante Entity Framework Core.

## Instrucciones

### 1. Configuración Inicial:

- Crea un nuevo proyecto ASP .NET Core utilizando Visual Studio Code o Visual Studio.
- Configura el proyecto con el patrón de diseño MVC.

### 2. Creación de Páginas Razor:

- Implementa al menos tres páginas Razor que incluyan contenido dinámico.
- Utiliza directivas de control, bucles y condicionales en las vistas.

### 3. Integración con Entity Framework Core:

- Define al menos dos modelos de datos relacionados con el tema de tu aplicación.
- Configura Entity Framework Core para gestionar la persistencia de datos.



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA



Explorador de soluciones

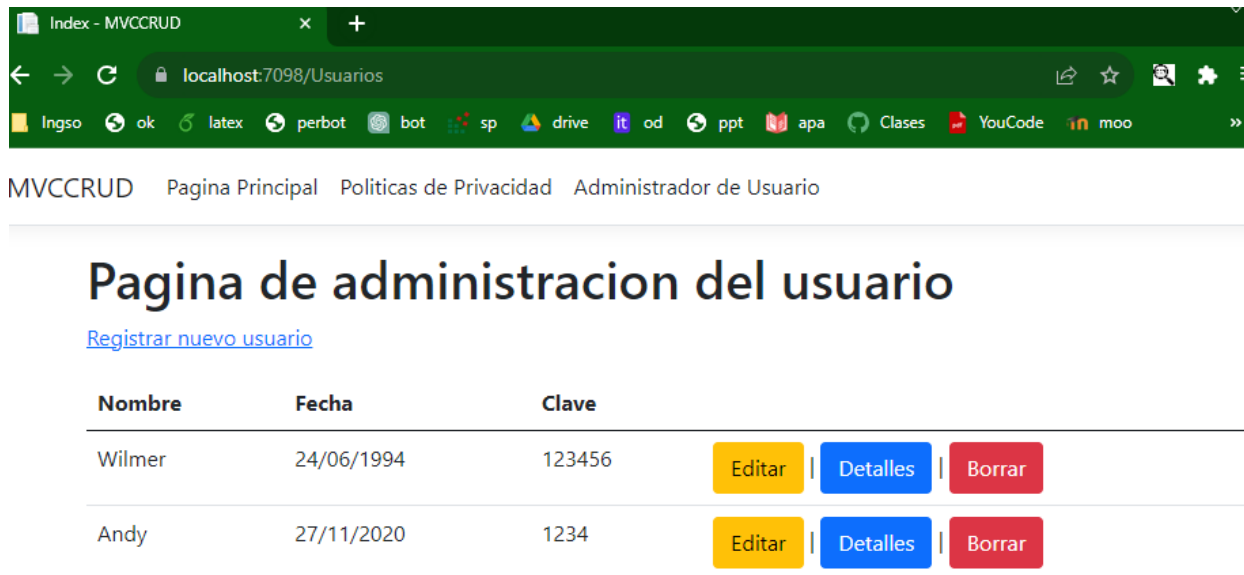
Buscar en Explorador de soluciones (Ctrl+<sup>o</sup>)

Solución "MVCCRUD" (1 de 1 proyecto)

- ▲ **MVCCRUD**
  - ▶ Connected Services
  - ▲ Dependencias
    - ▶ Analizadores
    - ▶ Marcos de trabajo
    - ▲ Paquetes
      - ▶ EntityFramework (6.4.4)
      - ▶ Microsoft.AspNetCore.Components.QuickGrid.EntityFrameworkExtensions (8.0.0)
      - ▶ Microsoft.EntityFrameworkCore (8.0.0)
      - ▶ Microsoft.EntityFrameworkCore.SqlServer (8.0.0)
      - ▶ Microsoft.EntityFrameworkCore.Tools (8.0.0)
      - ▶ Microsoft.VisualStudio.Web.CodeGeneration.Design (8.0.0)
  - ▲ Properties
    - launchSettings.json
  - ▶ wwwroot
  - ▲ Controllers
    - ▶ HomeController.cs
    - ▶ UsuariosController.cs
  - ▲ Models
    - ▶ ErrorViewModel.cs
    - ▶ MVCCRUDContext.cs
    - ▶ Usuario.cs
  - ▲ Views
    - ▲ Home
      - Index.cshtml
      - Privacy.cshtml
    - ▲ Shared
      - ▶ \_Layout.cshtml
      - ▶ \_ValidationScriptsPartial.cshtml
      - Error.cshtml
    - ▲ Usuarios
      - Create.cshtml
      - Delete.cshtml
      - Details.cshtml
      - Edit.cshtml
      - Index.cshtml
      - \_ViewImports.cshtml
      - \_ViewStart.cshtml
  - ▶ appsettings.json
  - ▶ Program.cs

## 4. Operaciones CRUD:

- Implementa operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para los modelos definidos.
- Asegúrate de validar las entradas del usuario y gestionar posibles errores.



The screenshot shows a web browser window with the address bar at `localhost:7098/Usuarios`. The page title is "Index - MVCCRU". The navigation bar includes links for "MVCCRU", "Pagina Principal", "Políticas de Privacidad", and "Administrador de Usuario". The main heading is "Pagina de administracion del usuario", with a link "Registrar nuevo usuario" below it. A table displays user data with columns for "Nombre", "Fecha", and "Clave". Each row includes three action buttons: "Editar" (yellow), "Detalles" (blue), and "Borrar" (red).

Nombre	Fecha	Clave	
Wilmer	24/06/1994	123456	<a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Borrar</a>
Andy	27/11/2020	1234	<a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Borrar</a>

**Git Hub:**

<https://github.com/WilmerBuestan/webAVZ/>

# Instrucciones para Ejecutar y Probar la Aplicación MVCCRUD

¡Bienvenido/a a la aplicación MVCCRUD! A continuación, encontrarás instrucciones claras sobre cómo ejecutar y probar la aplicación.

## Requisitos Previos

1. Entorno de Desarrollo: Asegúrate de tener un entorno de desarrollo configurado con Visual Studio.
2. Base de Datos: Asegúrate de tener un servidor SQL Server en ejecución y crea la base de datos utilizando el script proporcionado BDMVCCRUD.sql.

## Configuración del Proyecto

1. Clonar el Repositorio: Clona este repositorio en tu máquina local utilizando el siguiente comando:

*git clone* <https://github.com/WilmerBuestan/webAVZ/MVCCRUD>

O descarga el código fuente como un archivo ZIP y extráelo en tu máquina.

2. Abrir la Solución: Abre el archivo de solución MVCCRUD.sln en Visual Studio.
3. Configurar la Cadena de Conexión: Abre el archivo appsettings.json en la carpeta MVCCRUD y ajusta la cadena de conexión a tu base de datos SQL Server.

```
"ConnectionStrings": {  
  "DefaultConnection": "Server=localhost;Database=BDMVCCRUD;Trusted_Connection=True;"  
},
```

## Ejecutar la Aplicación

1. Compilar el Proyecto: Compila el proyecto en Visual Studio para asegurarte de que no hay errores

de compilación.

2. Actualizar la Base de Datos: Abre la Consola del Administrador de Paquetes (Tools -> NuGet Package Manager -> Package Manager Console) y ejecuta el siguiente comando para aplicar las migraciones y actualizar la base de datos:

#### *Update-Database*

3. Iniciar la Aplicación: Presiona F5 o selecciona "IIS Express" para iniciar la aplicación.
4. Acceder a la Aplicación: Abre tu navegador web y accede a <http://localhost:7098>.

## **Probar la Aplicación**

1. Crear un Nuevo Registro:

Haz clic en el enlace "Registrar nuevo usuario" en la página principal.

Completa el formulario y haz clic en "Crear".

2. Editar un Registro:

En la página principal, haz clic en "Editar" junto al registro que deseas editar.

Modifica la información y haz clic en "Guardar".

3. Ver Detalles de un Registro:

En la página principal, haz clic en "Detalles" junto al registro que deseas ver en detalle.

4. Eliminar un Registro:

En la página principal, haz clic en "Borrar" junto al registro que deseas eliminar.

## **Detener la Aplicación**

5. Detener la Aplicación: En Visual Studio, detén la aplicación presionando Shift + F5 o haciendo clic en el botón de detener.

## **Problemas Comunes**

- Error de Conexión a la Base de Datos: Verifica que la cadena de conexión en appsettings.json es correcta y que la base de datos se ha creado correctamente.
- Error al Aplicar Migraciones: Asegúrate de que la base de datos esté accesible y no esté siendo utilizada por otra instancia.