|  |  |  |  |
| --- | --- | --- | --- |
| **Cargo:** | Docente Especialista | | |
| **Nombre:** | Cesar Gabriel Hoyos López | | |
| **Asignatura:** | Desarrollo Web 2 | | |
| **Carrera:** | Desarrollo de Software | **Nivel:** | Tercer nivel |
| **Estudiante:** | Alejandro Bedoya | | |

**ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO**

**PASOS DE CONFIGURACION DEL PROYECCTO CRUD DE VISUAL STUDIO 2022**

**Objetivos**

Realización de modificación de programa de Visual Studio.

**Antecedentes/Escenario**

En ASP.NET Core MVC, el patrón Modelo-Vista-Controlador organiza el desarrollo en tres capas: el modelo representa los datos y la lógica de negocio, el controlador actúa como intermediario entre los datos y la vista, y la vista es responsable de mostrar la información al usuario.

El controlador, como HomeController.cs, contiene acciones que responden a solicitudes del usuario. Cada acción puede crear una instancia de un modelo, llenar sus propiedades con datos y enviarla a una vista Razor utilizando return.

**Recursos necesarios**

Computadora con acceso a internet.

Visual Studio 2022.

SQL Server.

Realización los nuevos cambios del proyecto de Visual Studio 2022.

Formato de taller.

**Planteamiento del problema**

Genere un informe con capturas de cada una de las clases(.CS) generadas en la carpeta Cotroller y los cambios realizados para obtener todos los datos del modelo en el controlador (HomeController.cs) y consultar que son los archivos las Razor Views , Razor Propierties y explique cómo funcionan en C# utilizando Visual Studio 2022

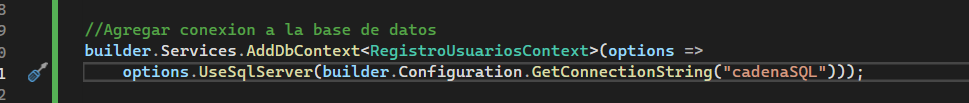
**Pasos por realizar**

* Paso 1: Iniciar el computador.
* Paso 2: Realizar la instalación de paquetes en Visual Studio.
* Paso 4: Realizar el formato de talleres.

**Desarrollo**

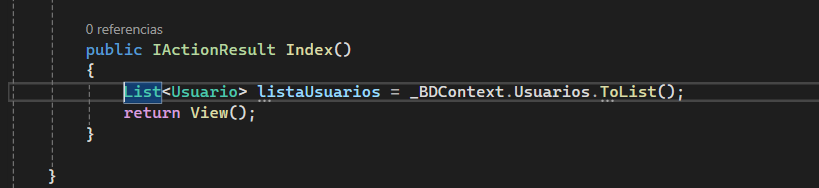
**Paso 1:**

Configuramos en la clase HomeControllers,cs establecemos la conexión a la base de datos para la aplicación Razor Pages usando Entity Framework Core.



**Paso 2:**

La variable listaUsuarios se crea y se llena con los datos de la base de datos, pero no se utiliza. El método simplemente retorna la vista con return View(); sin pasarle la lista de usuarios.

****

**Paso 2:**

**Razor Views y Razor Properties en C# con Visual Studio 2022**

**Razon Views**

Las Razor Views son archivos .cshtml que combinan HTML con C# utilizando la sintaxis Razor. Son utilizadas en aplicaciones ASP.NET Core MVC para mostrar contenido dinámico al usuario. Se ubican dentro de la carpeta Views y están asociadas a los controladores.

**Razor Properites**

Las Razor Properties son las propiedades del modelo que se acceden en una vista Razor usando la sintaxis. Propiedad. Estas propiedades permiten mostrar datos dinámicos en la interfaz del usuario. También puedes declarar variables o propiedades directamente en la vista con Razor, para lógica simple o presentación personalizada.

**¿Para que sirve Visual Studio 2022?**

Visual Studio 2022 facilita el trabajo con Razor Views al ofrecer plantillas de vistas, resaltado de sintaxis, IntelliSense y depuración. Las Razor Views permiten **crear interfaces de usuario dinámicas** conectadas con los datos del backend, haciendo posible desarrollar aplicaciones web modernas, organizadas y mantenibles.

**¿Cómo funciona en C#?**

Cuando un controlador llama a return View, envía un objeto a la vista. La vista recibe este modelo con @model y accede a sus propiedades mediante @Model. Razor interpreta el C# embebido en HTML y lo convierte en una página web dinámica.

**Conclusiones**

Usar Views permiten separar la lógica de presentación del código del servidor, facilitando el desarrollo de aplicaciones web limpias y estructuradas en ASP.NET Core MVC.

Implementar Properties ofrecen una manera directa de acceder a los datos del modelo, haciendo posible mostrar contenido dinámico de forma clara y sencilla en la vista.

Implementar Visual Studio 2022 mejora la productividad al trabajar con Razor mediante herramientas como IntelliSense, depuración integrada y plantillas, lo que acelera el desarrollo y reduce errores.

**Recomendaciones**

Utilizar el patrón MVC correctamente: Mantén la lógica de negocio en los modelos y evita sobrecargar las vistas con código complejo.

Declarar claramente el tipo de modelo con @model al inicio de cada vista Razor, para mejorar la legibilidad y aprovechar IntelliSense en Visual Studio.

Organizar tus vistas y controladores siguiendo las convenciones del framework para mantener tu proyecto ordenado, lo que facilita su mantenimiento y escalabilidad.