**Manual Técnico**

**Requisitos y Performance:**

Para poder correr el programa es necesario contar con los siguientes requerimientos:

1. Sistema Operativo Window 10.
2. Procesador de 2.5Ghz.
3. RAM 4Gb.
4. Navegador Web para su visualización.

Luego de contar con los requisitos del equipo se describe el sistema de carpetas.

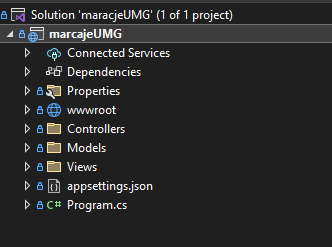
**Tecnologías:**

Para el desarrollo del siguiente proyecto se utilizaron las siguientes tecnologías:

1. SQL Server 2019.
2. C# y ASP .net 6
3. CHTML
4. Framework Bootsrap para estilos.

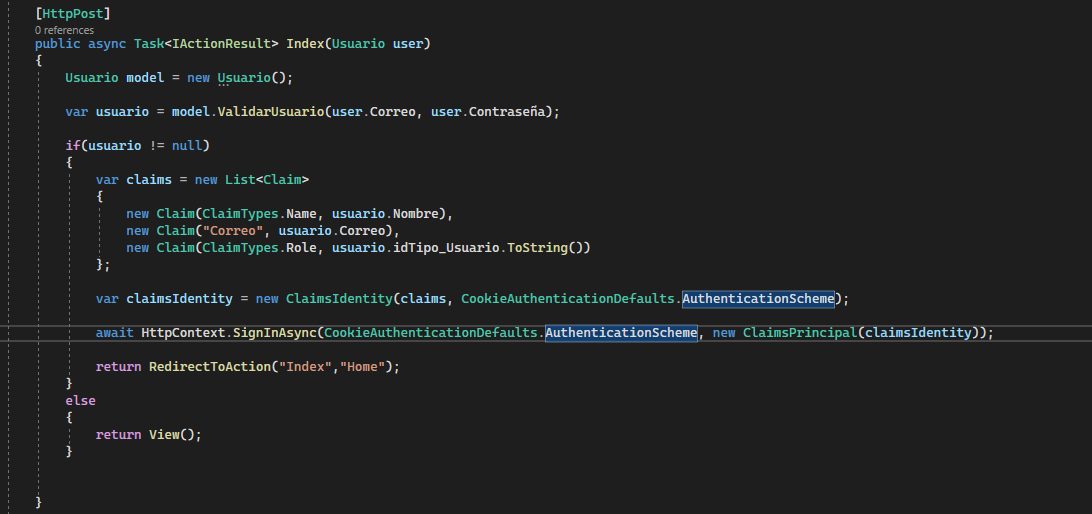
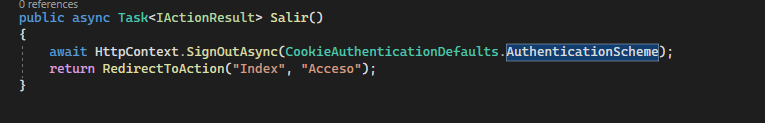
**Descripción de la estructur del proyecto:**

En la carpeta principal o raíz del proyecto encontramos la siguiente estructura:



A continuación, se describe el contenido de las carpetas:

* **Properties**: Contiene el archivo launchSettings.json que contiene la configuración para el despliegue del proyecto.
* **Controllers**: Contiene los siguientes archivos:

1. **AccesoController.cs:** Este archivo contiene la lógica necesaria para la administración del ingreso, contiene los siguientes métodos:
   1. **Index**: el cual dirige a la vista de LogIn y también la generación de una cookie mediante un objeto Claim.
   2. **Salir:** Este método elimina la cookie al momento de cerrar la sesión y que no quede almacenada esta.
2. **CatedraticoController.cs :** Contiene toda la lógica necesaria para el crud de administración de Catedráticos, tiene los siguientes métodos:
   1. **Index:** Esta es la vista principal que muestra la tabla que contiene toda la información de los catedráticos:

**Texto

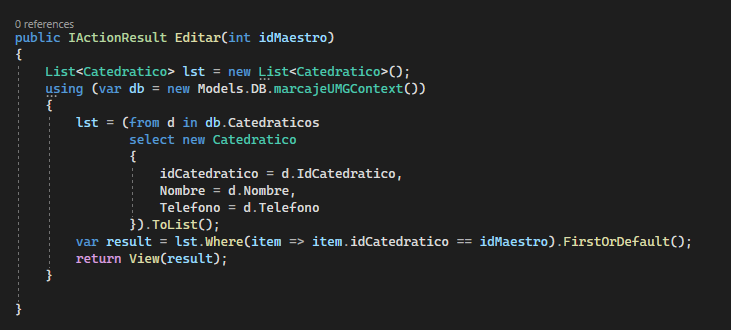
Descripción generada automáticamente**

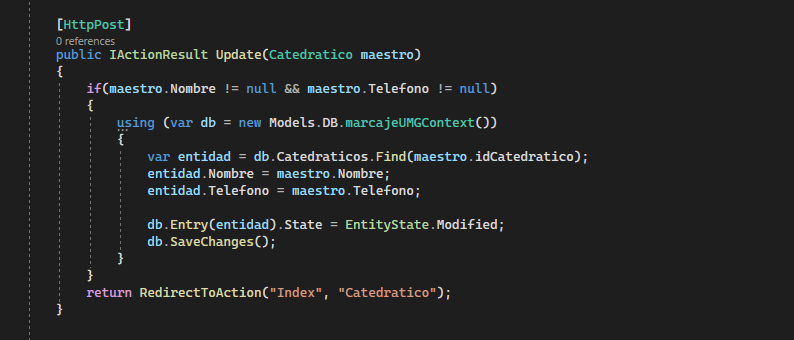
* 1. **Crear:** Este método se utiliza redireccionar a la vista para crear e ingresar nuevos catedráticos.
  2. **Save:** Este método se utiliza para Guardas la información ingresada por método POST.

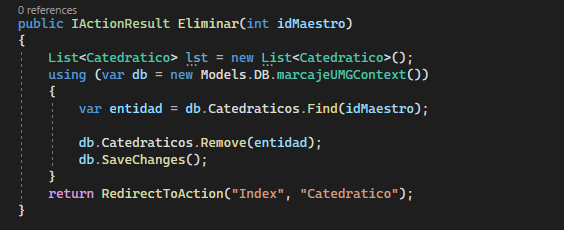
**Texto

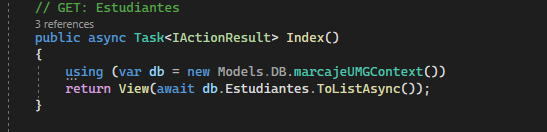
Descripción generada automáticamente**

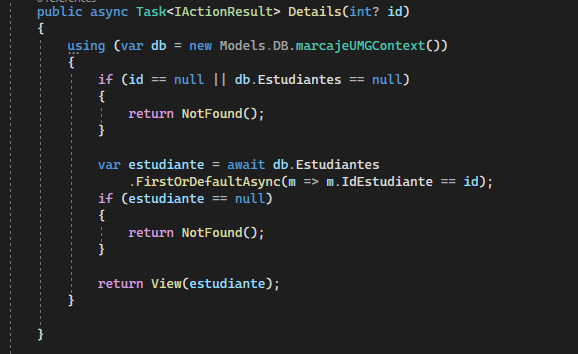
* 1. **Editar:** Captura los datos del catedrático que se desea editar y así mismo los muestra en la vista de edición respectivamente.

****

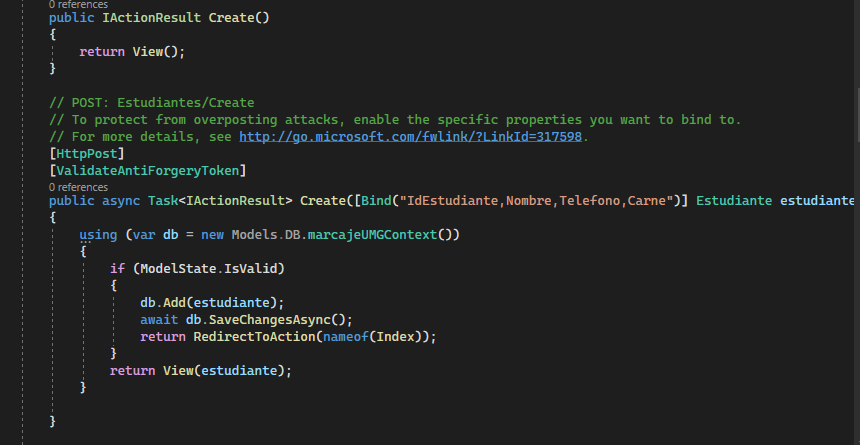
* 1. **Update:** Este método se utiliza para redirigir a la vista con todos los datos capturados de la base de datos respecto al id obtenido.
  2. **Eliminar:** Este método se utiliza para eliminar registros de la base de datos:

****

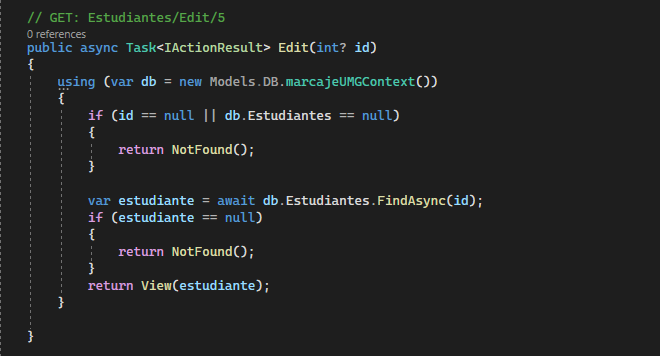
1. **EstudianteController.cs :** Este controlador maneja los métodos para poder controlar las vistas de Estudiantes los cuales son:
   1. **Index:** Este método muestra la tabla principal donde se verán todos los datos de la tabla Estudiante.
   2. **Details:** Este método muestra los detalles del elemento seleccionado en una vista adicional.

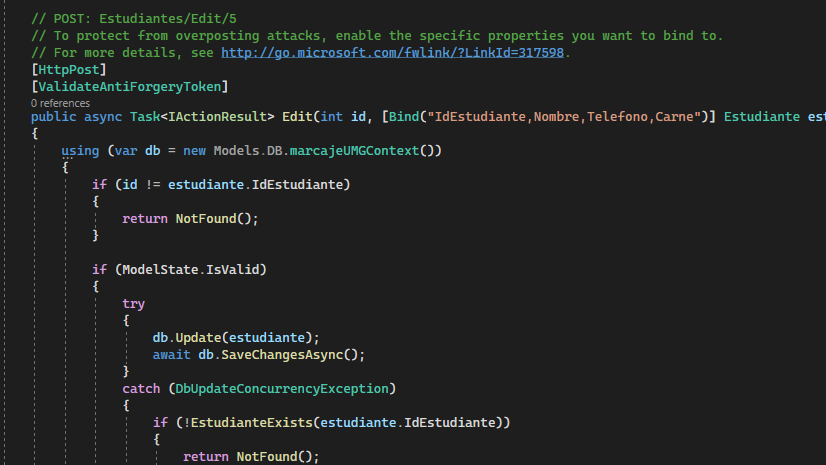
****

* 1. **Create:** Estos métodos se utilizan para la creación de registros en la tabla Estudiante.

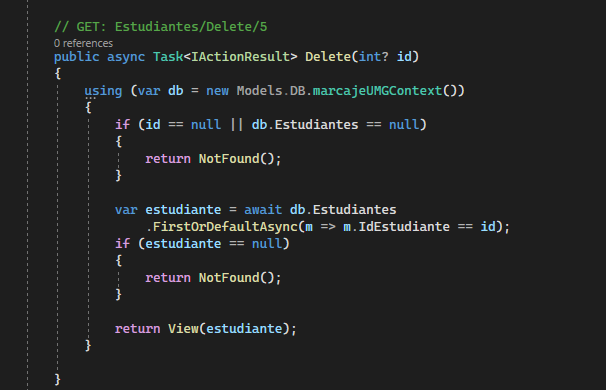
****

* 1. **Edit:** Estos métodos fueron utilizados para capturar los datos mostrarlos en un formulario de edición y también el método de envió mediante POST.

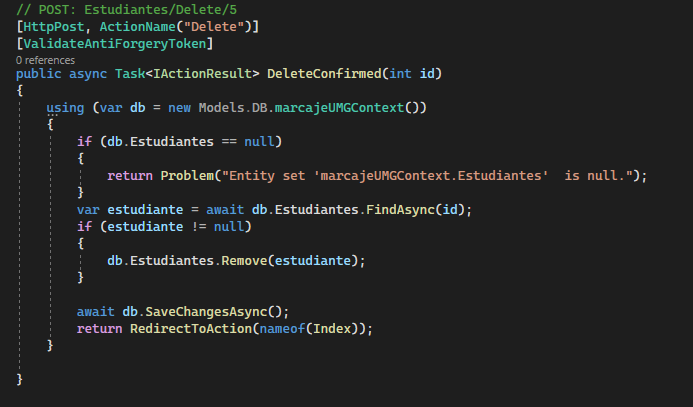
****

****

* 1. **Delete:** Este método se utiliza para la eliminación de registros en la table de Estudiante.

****

* 1. **DeleteConfirmed:** Este método es para llevar a la vista de eliminación que confirma la eliminación del registro.

****

1. **HomeController.cs:** Este controlador se utiliza para manejar las vistas y los permisos necesarios para cada una de las vistas de cada Rol y usuario.
2. **UsuariosController.cs:** Este controlador es utilizado para poder manejar el crud de los Usuarios respectivos relacionados con su ID de propietario y un identificador.

* Models: Contiene los siguientes archivos y carpeta:
  1. **DB (dir):** Contiene el modelo de todas las tablas de la base de datos, los métodos necesarios para obtener los datos de cada tabla.
  2. **Alumno.cs:** Tiene el modelo para la administración de los alumnos y sus respectivos datos.
  3. **Catedratico.cs:** Tiene el modelo para la administración de los catedráticos y sus respectivos datos.
  4. **Centro.cs:** Tiene el modelo para la administración de los centros y sus respectivos datos.
  5. **ErrorViewModel.cs:** Tiene el modelo para la de los errores y sus respectivos datos.
  6. **Usuario.cs:** Tiene el modelo para la administración de los alumnos y sus respectivos datos.
* Views: Contiene los siguientes archivos y carpetas:
  1. **Acceso (dir):** Contiene las vistas necesarias para poder manejar el acceso de los usuarios y sus respectivos datos.
  2. **Catedratico (dir):** Contiene las vistas necesarias para poder manejar los datos de los catedraticos.
  3. **Estudiantes (dir):** Contiene las vistas necesarias para poder manejar los datos de los estudiantes.
  4. **Home (dir):** Contiene las vistas principales y la vista home que verán todos los usuarios.
  5. **Shared (dir):** Contiene las vistas necesarias compartidas entre todas las demás, que tengan esta vista.
  6. **Usuarios (dir):** Contiene las vistas necesarias para poder manejar los datos de los usuarios.
* appsettings.json: Contiene toda la información respecto a la conexión con la base de datos y el login.
* Program.cs: Contiene toda la información del arranque de la aplicación y los frameworks que deberá usar al arranque.