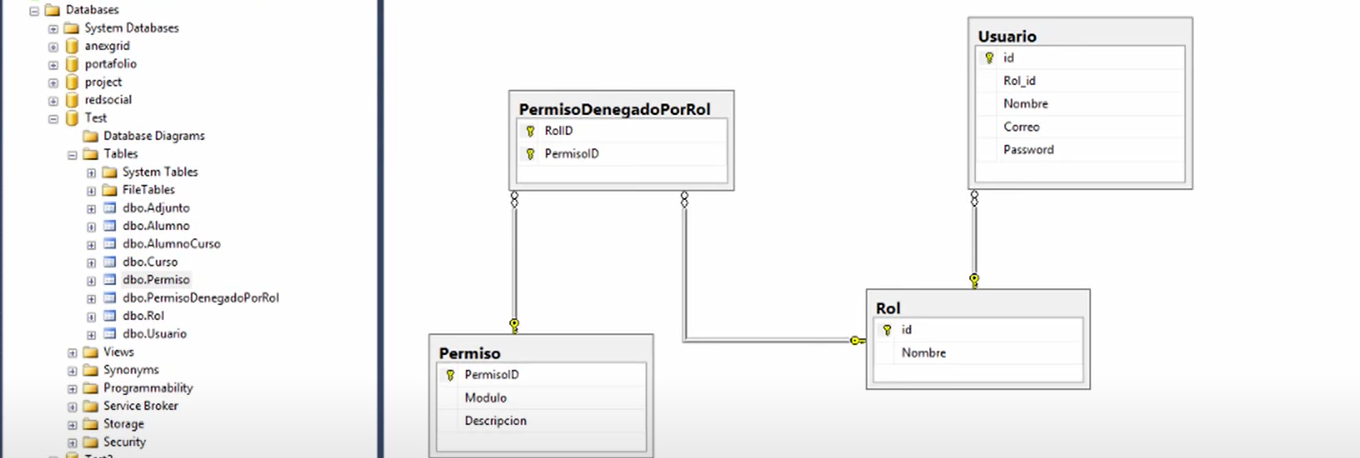
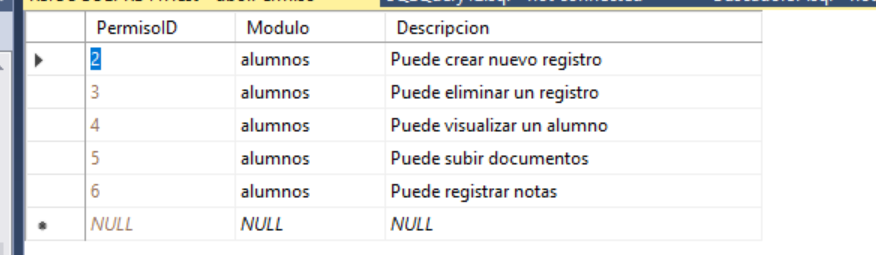
Documentación prueba lavital:

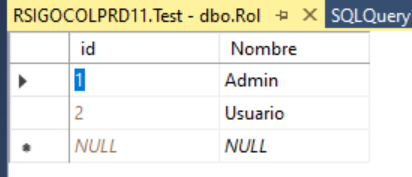
-Relaciones de las tablas usuarios, roles y permisos:



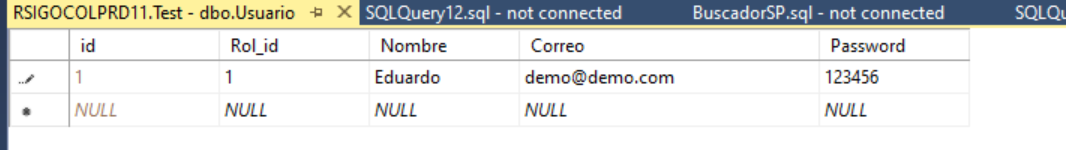
-Tabla de Permiso:



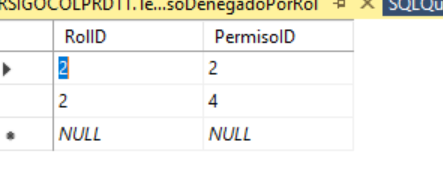
-Tabla de Rol:



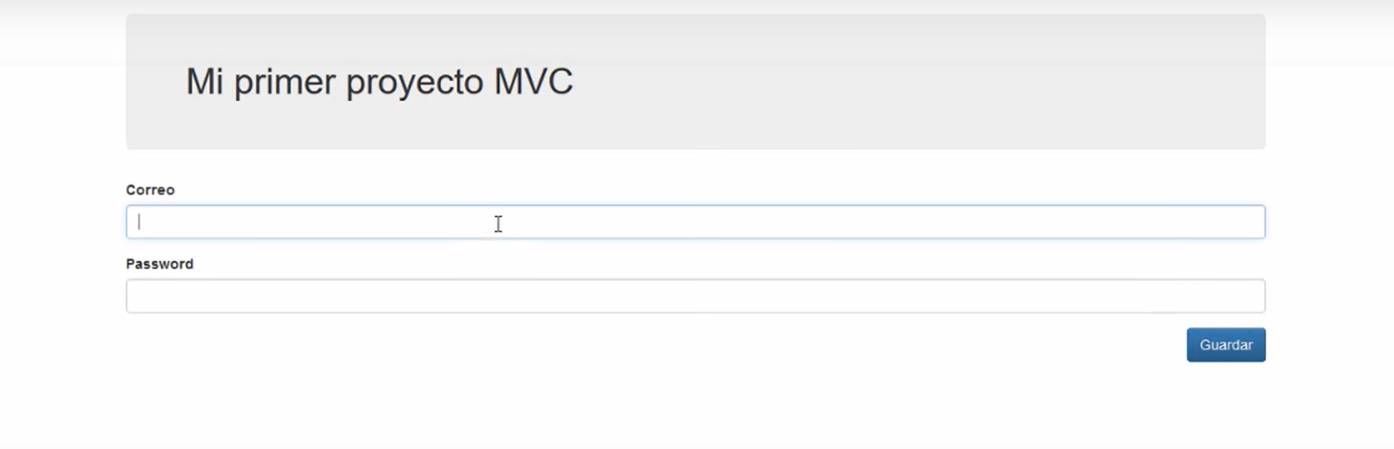
-Tabla de Usuario:



-Tabla de PermisoDenegadoPorRol



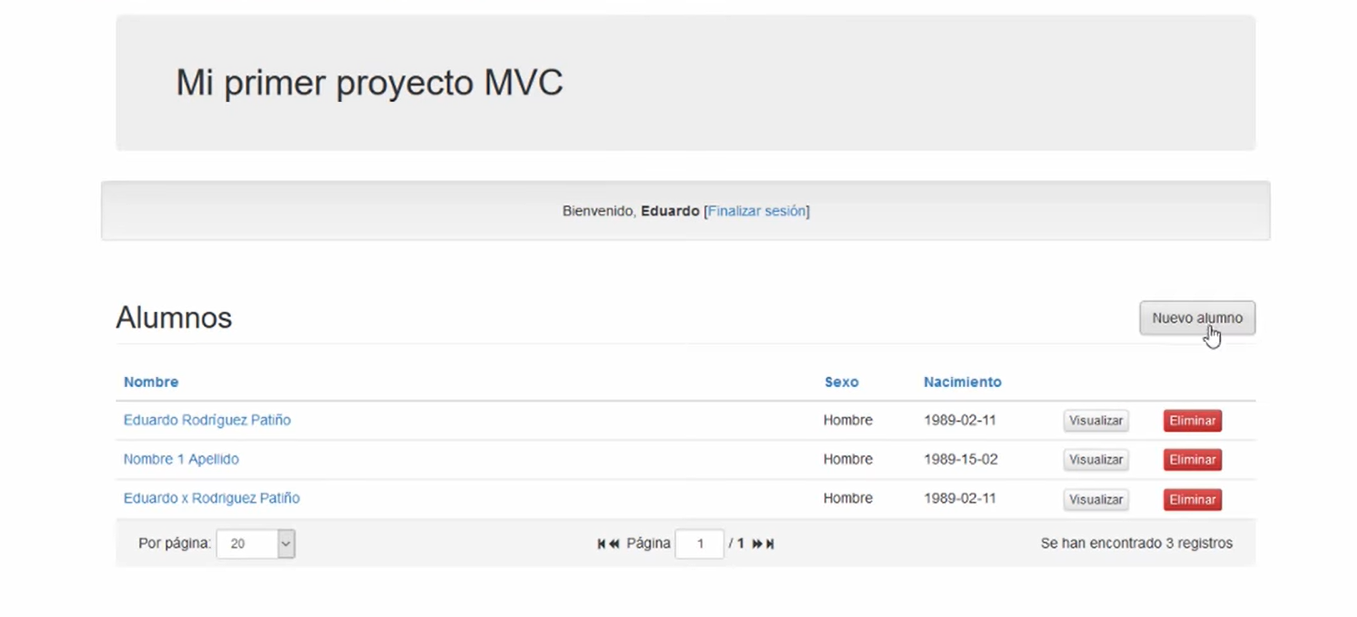
-Ventana de Login



Usuario : [demo@demo.com](mailto:demo@demo.com)

Pass : 123456

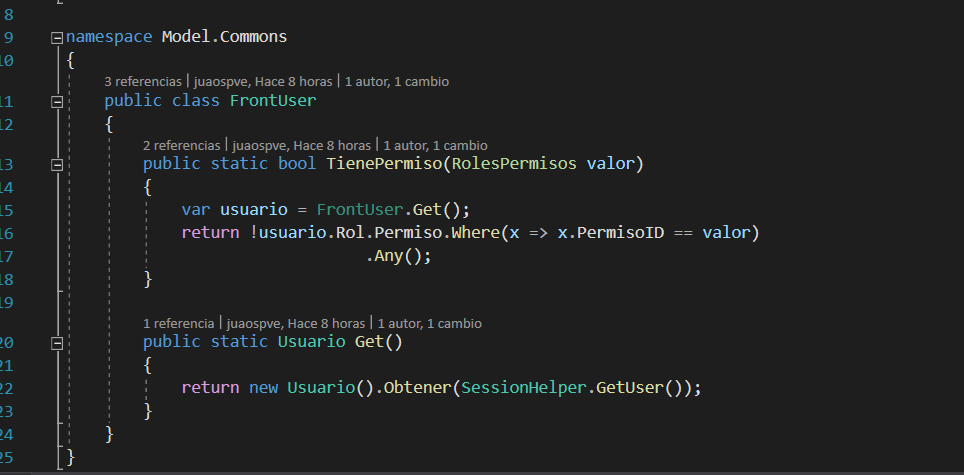
* Formulario de Usuarios:



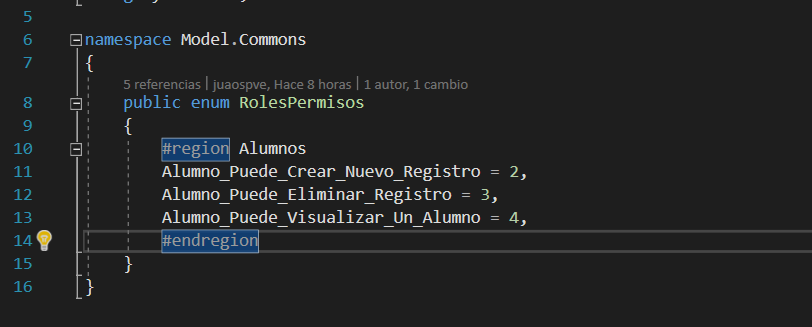
Nota : Dependiendo de los permisos que tenga asignado el usuario, así mismo puede ver los botones que puede usar en este formulario teniendo en cuenta el rol y el permiso denegado por rol.

* Se implementó en el modelo la siguiente lógica:

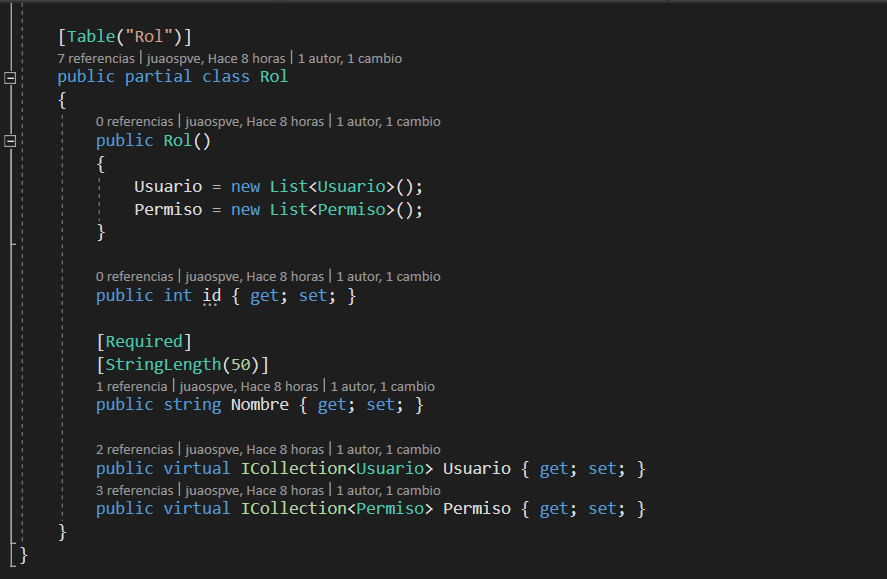
-La clase FrontUser.cs, esta clase nos permite saber que usuario esta autenticado en nuestro sistema y además implementa un método para saber si tiene permiso o no.



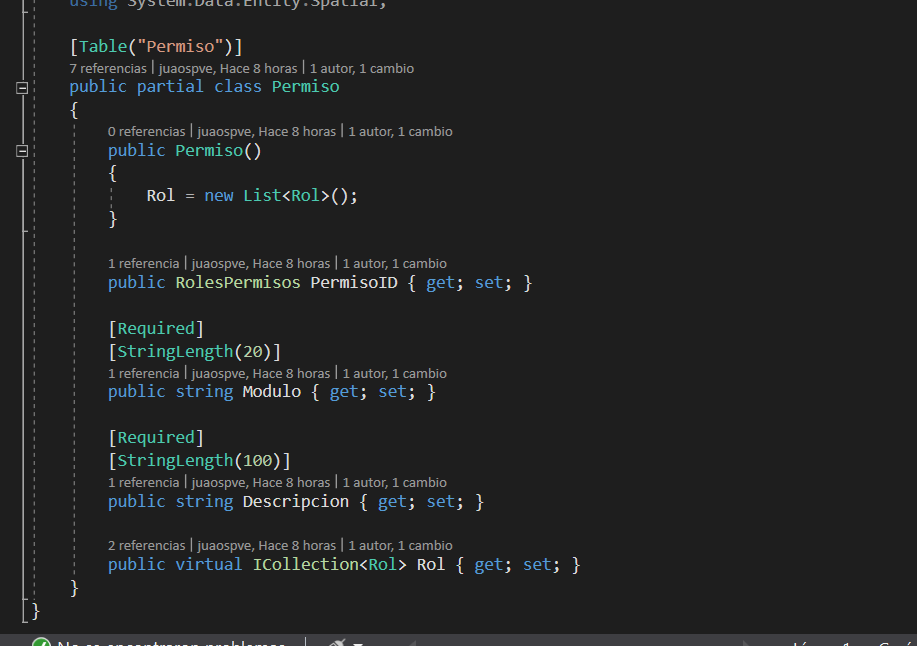
-La clase RolesPermisos.cs, mediante enums, se especifican los permisos que van a tener los roles.



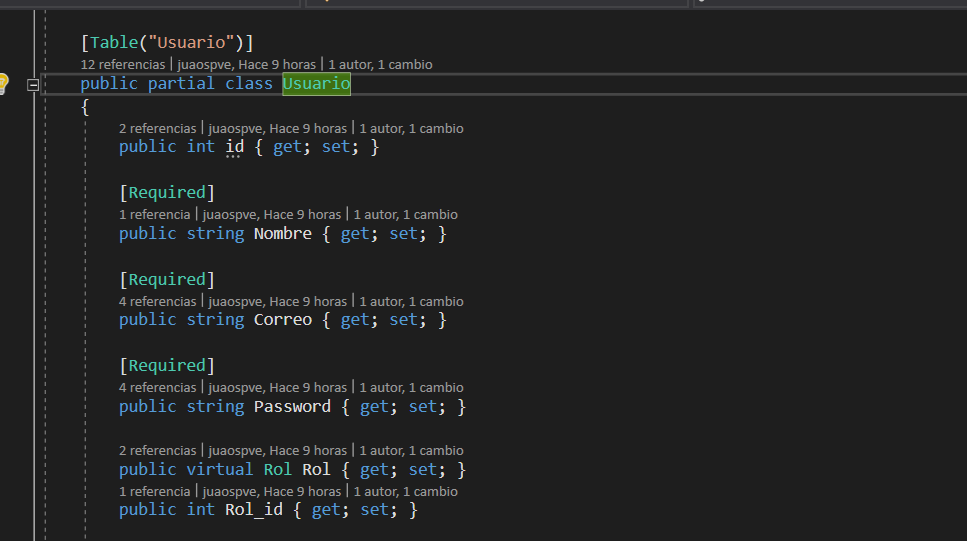
-La clase Rol.cs la cual hace referencia a la tabla Rol, en su constructor instancia la lista de usuarios y permisos.

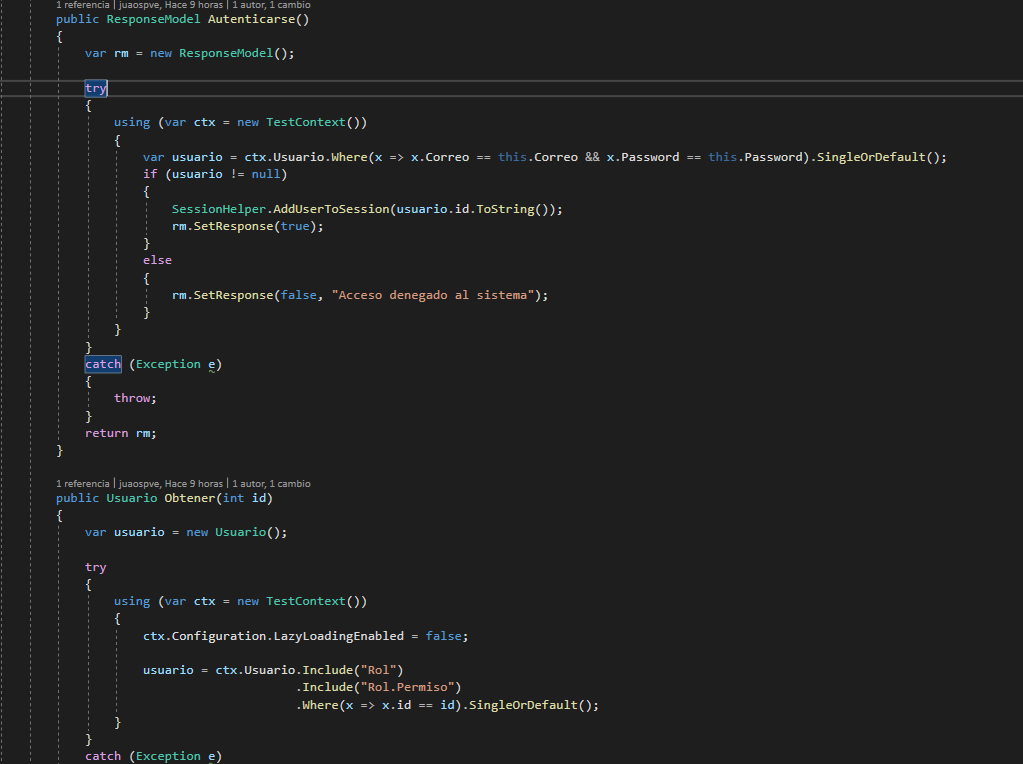


-La clase Permiso.cs haciendo referencia a la tabla Permiso.



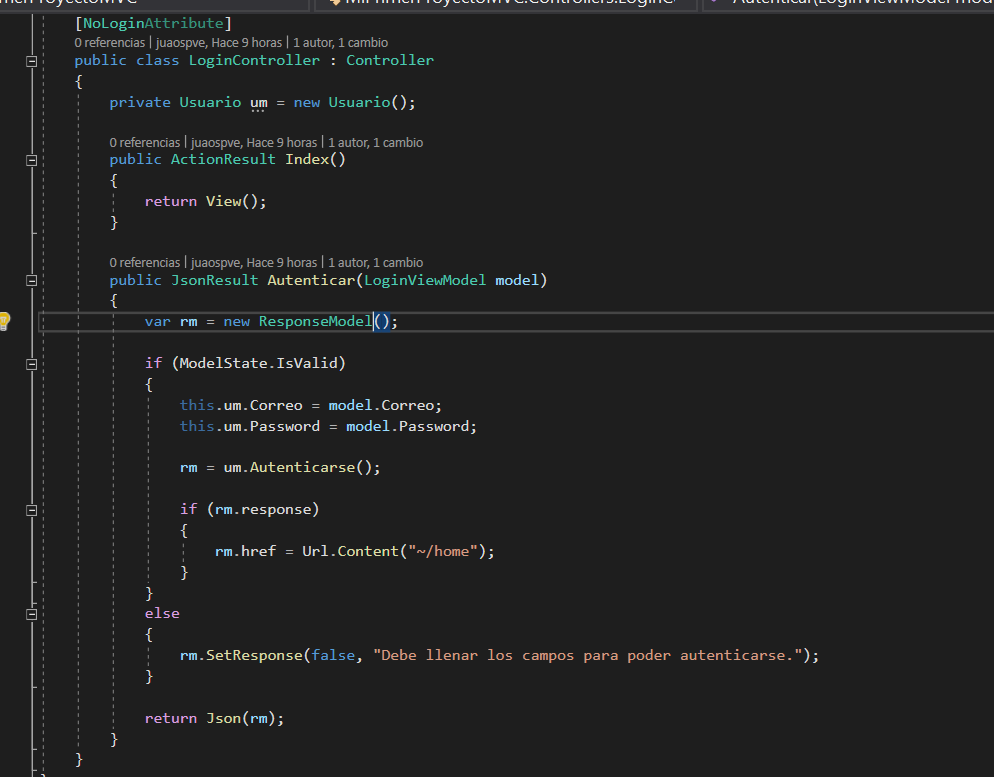
-La clase Usuario.cs la cual hace referencia a la tabla Usuario y tiene los métodos de autenticación y obtención de los roles del usuario .

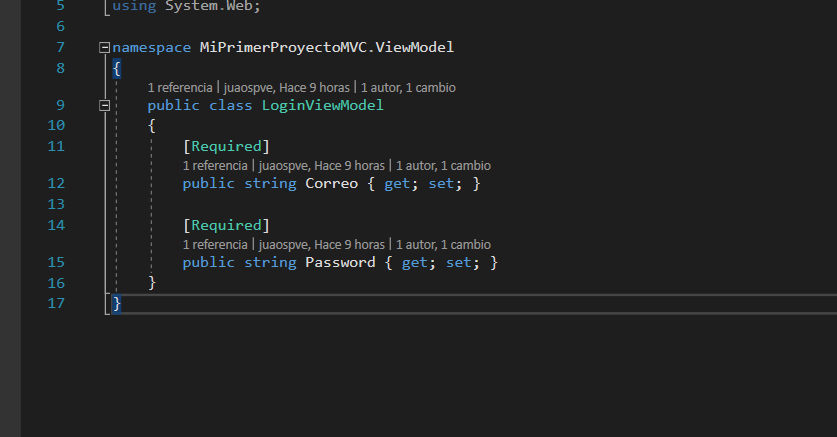




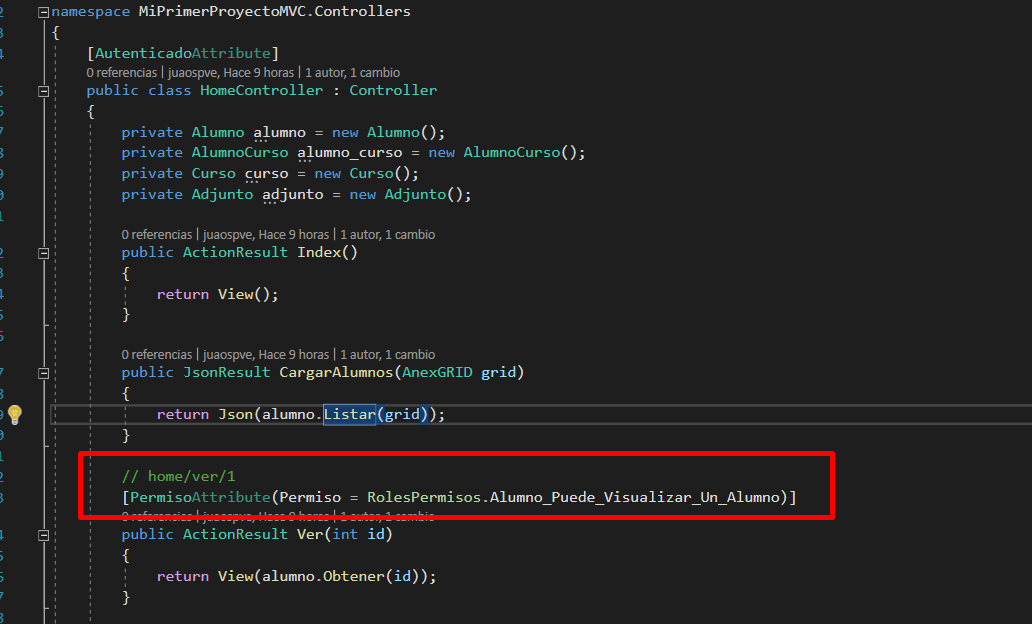
* Dentro de los Controladores tenemos:

-La clase LoginController.cs la cual va ligada al index o a la pagina de login inicial con el método de autenticación que hace referencia al modelo LoginViewModel donde recibe el correo y password necesarios para realizar el logueo.



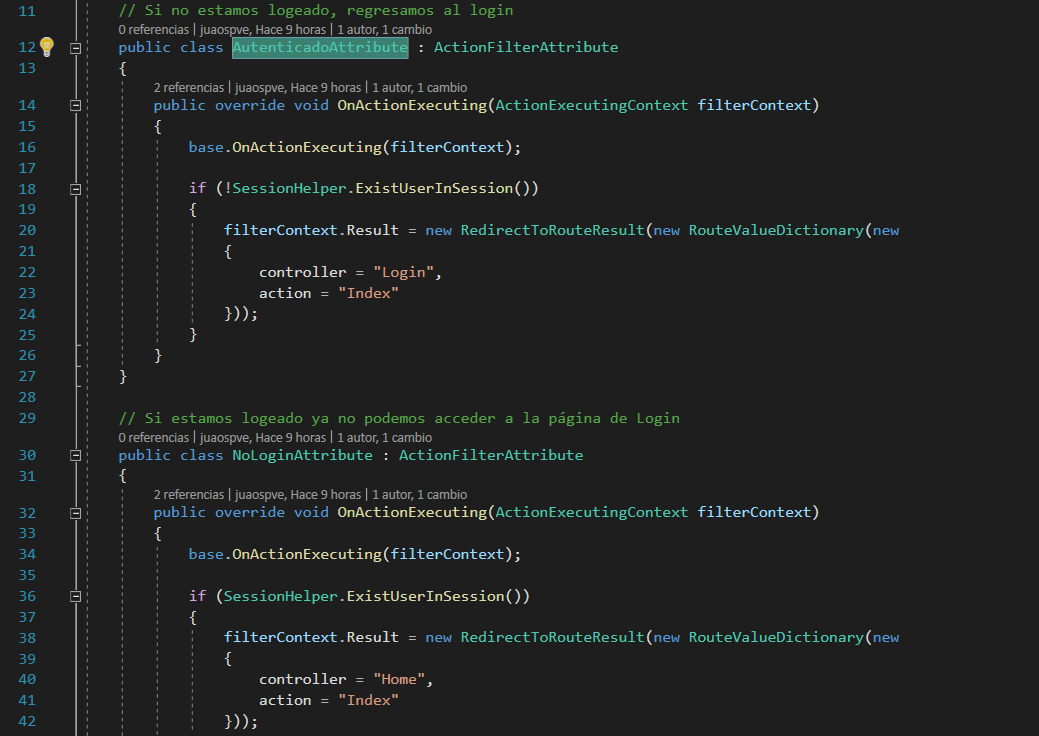


-En el HomeController.cs instanciamos los objetos correspondientes a los datos de alumnos y cursos pero es aquí donde se implementa la lógica de permisos mediante RolesPermisos y sus enums.

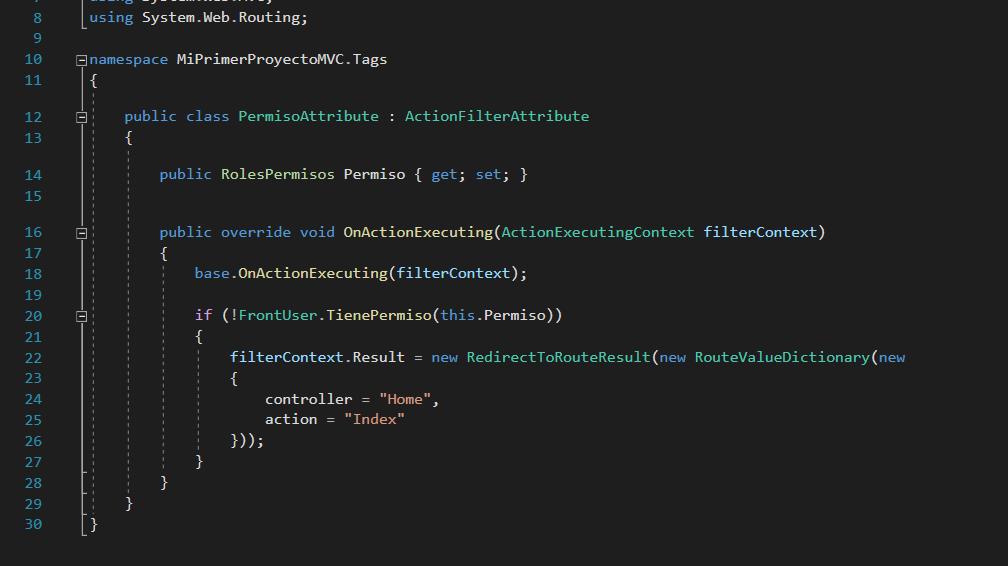


* En los Atributos tenemos:

-La clase AutenticadoAttribute.cs en la cual se tienen los métodos donde se define si se está logueado o no y que acción tomar frente a ello.



-La clase PermisoAttribute.cs en la cual se define si el usuario no tiene el permiso, lo redireccionamos a la página de principal.



* Dentro de las vistas también se controla la visibilidad de los botones para que no queden denegados solo por url.

