**Aplicación hecha en el lenguaje C# usando el framework Entity de .NET, usando el modelo MVC (modelo-vista-controlador).**

1. Usamos principalmente la conexión con SQL server, por lo que es importante que verifiques que tengas cargada la base de datos **‘BdPrueba.sql’** la cual tiene la estructura para el correcto funcionamiento de la aplicación.
2. Para ejecutar el programa es necesario que este instalada la versión 8.0 de la [Plataforma de .NET](https://dotnet.microsoft.com/es-es/download/dotnet/8.0).
3. Verifica que estén correctamente añadidas las siguientes librerías al proyecto, en VSCode ,abre una terminal en la carpeta del proyecto **(ctrl + ñ)** y ejecuta los siguientes comandos:

**dotnet add package Microsoft.EntityFrameworkCore**

**dotnet add package Microsoft.EntityFrameworkCore.SqlServer**

**dotnet add package Microsoft.EntityFrameworkCore.Tools**

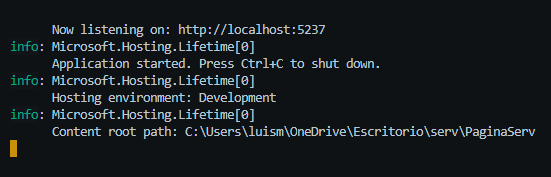
1. Una vez que se hayan instalado correctamente compilamos la aplicación:

**dotnet build**

1. Posteriormente ejecutamos la aplicación con el servidor en .Net:

**dotnet run**

1. Una vez que aparezca este resultado:



En el navegador web ingresamos a la ruta <http://localhost:5237> y visualizamos la pagina de login para ingresar a la pagina.

1. Para detener el programa, nos dirigimos a VSCode y en la terminal pulsamos **(ctrl+c)** para detener el servidor y realizar alguna modificación, **cabe destacar que cada ves que se vaya a realizar un cambio en el código de la aplicación se debe de detener la aplicación y volver a ejecutarla (punto 5 y punto 4)**
2. El correo por defecto para ingresar es: [correo123@gmail.com](mailto:correo123@gmail.com) y la contraseña es **123456**
3. Es probable que la parte de la consulta de los ingredientes no funcione ya que no hay registros agregados en las tablas de ingredientes y platillos, verifica que las relaciones entre ingredientes y platillos estén correctamente en la tabla **ingredientesPlatillos**, por lo que hay que añadirlos a las tablas, si ya están agregados debería funcionar sin problema.

Una vez que los pasos anteriores estén correctamente podemos empezar a usar el programa

La estructura de el proyecto esta basada en la estructura (MVC) modelo-vista-controlador por lo cual esta organizado en varias carpetas.

NombreDelProyecto

├── Controllers // Controladores que gestionan la lógica de la aplicación

│ └── HomeController.cs // Ejemplo de controlador

│

├── Models // Modelos de datos, que representan la estructura de los datos (En los modelos se encuentran los getters y setters para el manejo de los datos de las tablas de la base de datos. \*Verificar que el tipo de dato que tiene el modelo coincida con el tipo de dato que tiene la base de datos).

│ └── LoginViewModel.cs // Ejemplo de modelo

│

├── Views // Vistas o plantillas de la interfaz de usuario (HTML)

│ ├── Home // Carpeta de vistas para el controlador Home

│ │ └── Index.cshtml // Ejemplo de vista para la acción Index del controlador Home

│ ├── Shared // Vistas compartidas entre varios controladores

│ │ └── \_Layout.cshtml // Plantilla principal de diseño de la aplicación

│ └── \_ViewImports.cshtml // Configuraciones de importación para las vistas

│ └── \_ViewStart.cshtml // Configuración de layout común para todas las vistas

│

├── wwwroot // Carpeta raíz pública de la aplicación para archivos estáticos

│ ├── css // Archivos CSS personalizados

│ │ └── site.css

│ ├── js // Archivos JavaScript personalizados

│ │ └── site.js

│ └── lib // Bibliotecas de terceros (Bootstrap, jQuery, etc.)

│ ├── bootstrap

│ └── jquery

│

├── appsettings.json // Configuración de la aplicación (p. ej., cadenas de conexión)

├── Program.cs // Punto de entrada principal de la aplicación (En esta parte ingresas los datos para la conexión con sqlsever

└── Startup.cs // Configuración de servicios y pipeline de la aplicación

Referencia:<https://www.rafaelacosta.net/Blog/2024/11/2/aspnet-core-en-visual-studio-estructura-de-un-proyecto?AspxAutoDetectCookieSupport=1>

En la parte de la vista para agregar imágenes, las añadimos a la vista con esta estructura:

**<img src="@Url.Content("~/images/Products/plato.png")" alt="Plato de comida">**

En la cual agregamos la ruta de la imagen que guardamos en el proyecto, la carpeta raíz (wwwroot) en donde se encuentran los componentes de la vista como hojas de estilos, imágenes, javascript, librerías externas, y el icono de la aplicación.

Las carpetas están divididas, basada en la misma estructura de la vista.