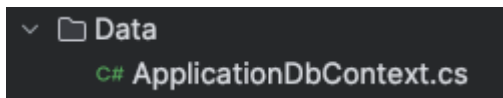
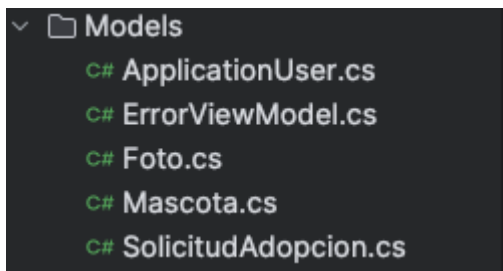


Estructura de archivos del proyecto

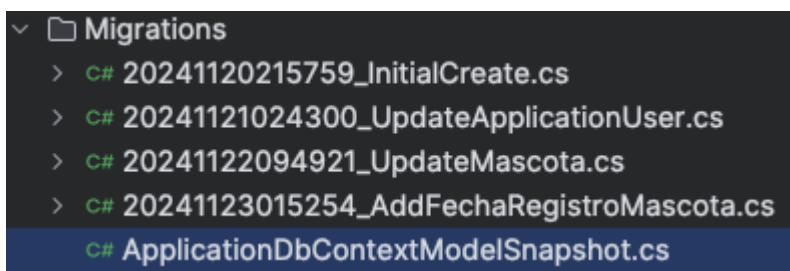
Dentro de Data esta el contexto para crear los modelos y las relaciones de los mismos en la base de datos



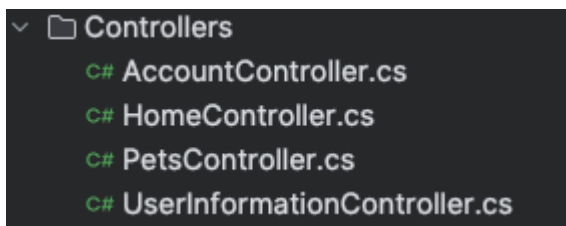
Dentro de Models están definidos los modelos de la base de datos



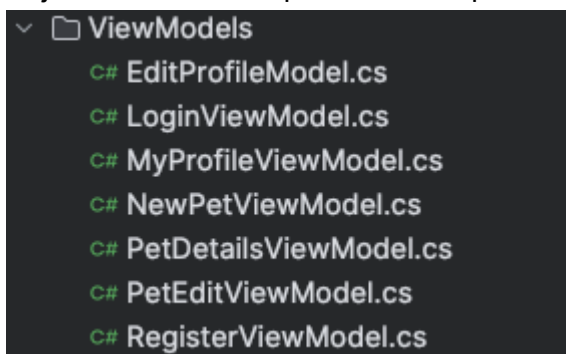
En Migrations se guardan todas las migraciones de modificaciones a los modelos en la base de datos



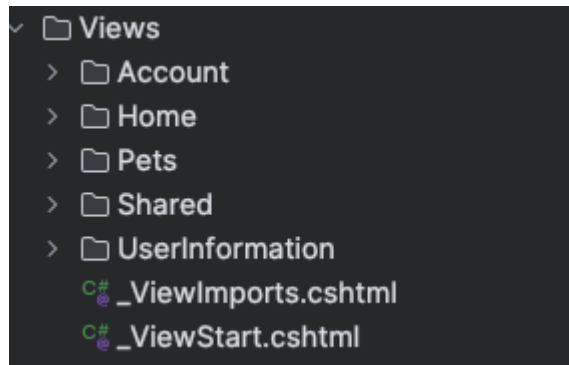
Dentro de Controllers están todos los controladores que orquestan las vistas y las funciones que llaman



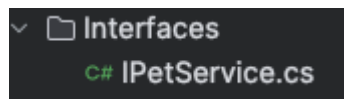
En ViewModels se definen los modelos que tendrán las vistas, de esta manera organizamos mejor la información que cada vista puede tener



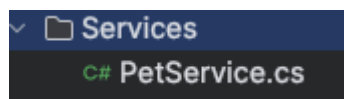
Dentro de Views tenemos las vistas de html del aplicativo



En Interfaces tenemos las interfaces para nuestros servicios



Dentro de Services tenemos definidos nuestros servicios



En nuestro Program.cs definimos nuestra conexión a la base de datos (MySQL)

```
// Configuración de MySQL
builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseMySQL(
        builder.Configuration.GetConnectionString( name: "DefaultConnection"),
        ServerVersion.AutoDetect(builder.Configuration.GetConnectionString( name: "DefaultConnection"))
    ));
```

Configuramos Identity para la gestión de usuarios

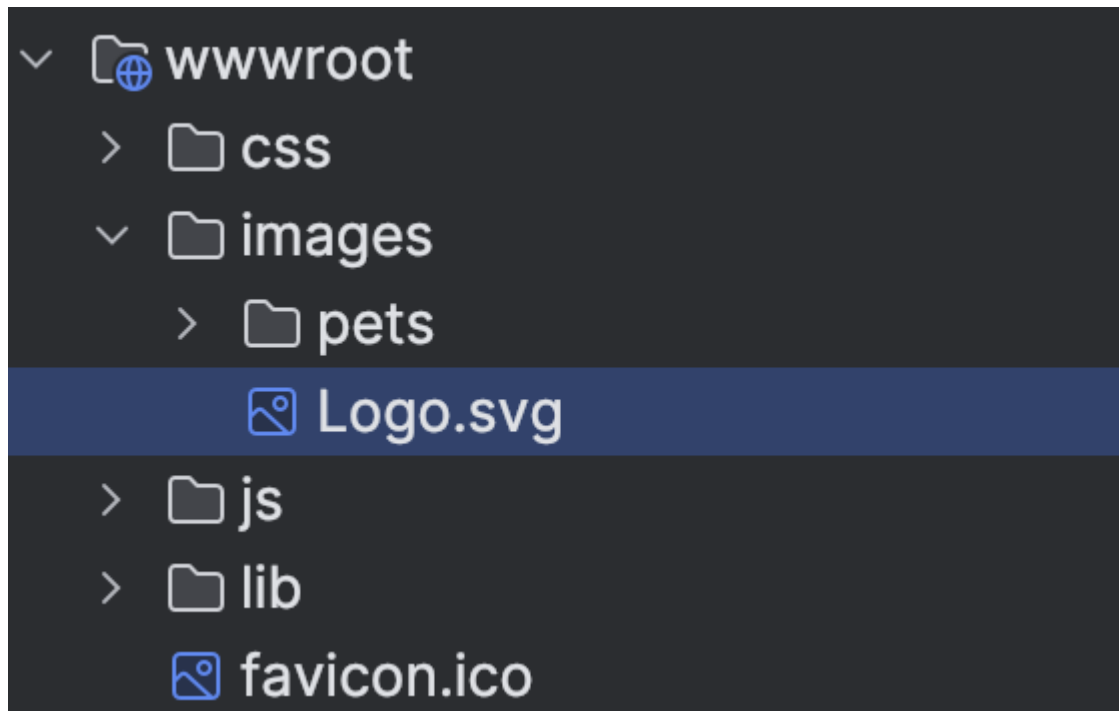
```
// Configuración de Identity
builder.Services.AddDefaultIdentity<ApplicationUser>(configureOptions: options => options.SignIn.RequireConfirmedAccount = false)
    .AddEntityFrameworkStores<ApplicationDbContext>();
```

Agregamos nuestros servicios con AddScoped y AddTransient

```
// Registrar ApplicationDbContext como un servicio Scoped
builder.Services.AddScoped<ApplicationDbContext>();

// Services
builder.Services.AddTransient<IPetService, PetService>();
```

Dentro de `wwwroot/images/pets` se guardan las imágenes de las mascotas que se registran en la aplicación



Instalación del proyecto

Para el correcto funcionamiento del proyecto, se requiere contar con una instancia de MySQL Server en su versión más reciente (8.4.1 LTS), la cual puede estar alojada de manera local o en una instancia remota. Adicionalmente, es necesario crear un esquema (schema) con el nombre ****SalvandoHuellitasProdDB**** en la base de datos.

En el archivo `appsettings.json`, configuramos la conexión a la base de datos indicando el nombre de la base de datos, el usuario y la contraseña. En este caso, se utiliza el usuario predeterminado de MySQL, `root`, junto con su contraseña. Es importante asegurarse de que los valores coincidan con las credenciales configuradas en su instancia de MySQL.

```
{
  "ConnectionStrings": {
    "DefaultConnection": "server=localhost;port=3306;database=SalvandoHuellitasProdDB;user=root;password=0+%1,npHD70@9vz*WV2p#3LCY10xA*S_"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

Para ejecutar las migraciones de la base de datos y aplicar los cambios correspondientes, es necesario utilizar el siguiente comando en la terminal desde el directorio del proyecto:

```
dotnet ef database update
```

Adicionalmente, se deben instalar las siguientes dependencias a través de NuGet para garantizar el correcto funcionamiento del aplicativo. Es importante instalar las versiones específicas, ya que el proyecto fue desarrollado utilizando .NET Core 8.







Lista de dependencias requeridas:

Microsoft.EntityFrameworkCore: (Versión 8.0.0)

Microsoft.EntityFrameworkCore.Tools: (Versión 8.0.0)

MySQL.EntityFrameworkCore: (Versión 8.0.33)

Para instalar cada paquete, puede utilizar los siguientes comandos:

	Microsoft.AspNetCore.Identity.EntityFrameworkCore • 8.0.11	9.0.0
	Microsoft.AspNetCore.Identity.UI • 8.0.11	9.0.0
	Microsoft.EntityFrameworkCore.Tools • 8.0.11	9.0.0
	Microsoft.VisualStudio.Web.CodeGeneration.Design • 8.0.7	9.0.0
	bootstrap-select • 1.13.18	
	Pomelo.EntityFrameworkCore.MySql • 8.0.2	

```
dotnet add package Microsoft.EntityFrameworkCore --version 8.0.0
```

```
dotnet add package Microsoft.EntityFrameworkCore.Tools --version 8.0.0
```

```
dotnet add package MySQL.EntityFrameworkCore --version 8.0.33
```

Nota: Asegúrese de mantener las dependencias actualizadas solo si son compatibles con el resto del proyecto y revise las notas de la versión antes de actualizar cualquier paquete.

Estructura de la base de datos

Modelo **ApplicationUser**

Representa a los usuarios registrados en la aplicación y extiende la funcionalidad proporcionada por la clase base **IdentityUser** de ASP.NET Identity. Esta clase base incluye propiedades estándar para la autenticación y gestión de usuarios, como el correo electrónico, nombre de usuario y contraseña.

En este modelo personalizado, se añaden las siguientes propiedades específicas para adaptarse a los requisitos de la aplicación:

- **Nombre:** Almacena el nombre completo del usuario, facilitando la personalización y la identificación dentro del sistema.
- **Teléfono:** Permite registrar el número de teléfono del usuario, útil para propósitos de contacto o verificación.

Este modelo se utiliza en conjunto con el sistema de identidad para manejar el registro, autenticación y gestión de usuarios de manera robusta y escalable.

Modelo **Mascota**

Se encarga de representar a las mascotas registradas en la aplicación, almacenando toda la información necesaria para su adopción y gestión. Incluye las siguientes características:

- **Id:** Identificador único de la mascota.
- **Nombre (requerido):** Nombre de la mascota.
- **Tipo (requerido):** Clasificación de la mascota, como perro, gato, etc.
- **Descripción (requerida):** Información adicional que detalla las características de la mascota.
- **Género (requerido):** Género de la mascota (macho o hembra).
- **Edad (requerida):** Edad de la mascota, expresada en años.
- **Tamaño (requerido):** Tamaño de la mascota (pequeño, mediano, grande).
- **Sociable con (requerido):** Especifica con quién es sociable la mascota, como niños u otros animales.
- **Localización (requerida):** Ubicación de la mascota, esencial para el proceso de adopción.
- **Requisitos de adopción (requeridos):** Condiciones necesarias para que la mascota pueda ser adoptada.
- **Adoptado (requerido):** Indica si la mascota ya ha sido adoptada (valor booleano).
- **Fecha de registro (requerida):** Fecha en la que la mascota fue registrada en la plataforma.

Relaciones

- **Usuarioid (clave foránea):** Identificador del usuario que registró a la mascota, vinculado al modelo **ApplicationUser**.
- **Usuario:** Referencia al usuario responsable del registro.

- **Fotos:** Lista de imágenes asociadas a la mascota, gestionadas a través del modelo **Foto**.

Este modelo es clave para la gestión de las mascotas en la plataforma, ya que permite registrar, organizar, listar y filtrar información esencial para facilitar el proceso de adopción.

Modelo **Foto**

Representa las imágenes asociadas a las mascotas dentro de la aplicación, facilitando su gestión y almacenamiento. Estas fotos son clave para la identificación visual de las mascotas registradas. El modelo incluye las siguientes propiedades:

- **Id:** Identificador único de la foto.
- **Url** (*requerida*): Dirección o enlace donde se encuentra almacenada la imagen de la foto.

Relaciones

- **Mascotaid** (*clave foránea*): Identificador de la mascota a la que pertenece la foto, vinculado al modelo **Mascota**.
- **Mascota:** Referencia al modelo **Mascota**, que permite acceder a la información de la mascota asociada.

Este modelo es fundamental para mejorar la experiencia visual dentro de la plataforma, permitiendo que cada foto esté correctamente vinculada a la mascota correspondiente y facilitando la exploración de mascotas para adopción.

Modelo **SolicitudAdopcion**

Representa una solicitud realizada por un usuario para adoptar una mascota. Este modelo almacena la información relacionada con la adopción, incluyendo la mascota solicitada, el usuario que realiza la solicitud y la fecha en que fue realizada. Las propiedades del modelo son las siguientes:

- **Id:** Identificador único de la solicitud de adopción.
- **Mascotaid** (*clave foránea*): Identificador de la mascota para la que se está realizando la solicitud. Se relaciona con el modelo **Mascota**.
- **Mascota:** Referencia al modelo **Mascota**, que permite acceder a los detalles de la mascota solicitada.
- **Usuarioid** (*clave foránea*): Identificador del usuario que está solicitando la adopción. Se relaciona con el modelo **ApplicationUser**.
- **Usuario:** Referencia al modelo **ApplicationUser**, que permite acceder a los detalles del usuario solicitante.

- **Fecha:** Fecha en que la solicitud de adopción fue realizada. Por defecto, se establece con la fecha y hora actuales (`DateTime.Now`).

Este modelo es clave para gestionar el proceso de adopción, permitiendo que los usuarios puedan realizar solicitudes para adoptar mascotas y que los administradores o propietarios puedan revisarlas.

Diagrama Entidad Relación

