

Universidad Don Bosco
Desarrollo de Software Empresarial

CICLO 02- 2025



Manual Técnico

GRUPO TEORICO: 01

INTEGRANTES:

Apellidos	Nombres	Carnet
Alvarenga Claros	Carlos Alexander	AC211104
Doradea Herrera	José Roberto	DH211056
Moreno Lozano	Cruz Enrique	ML210800

DOCENTE: Ing. Emerson Francisco Cartagena Candelario

FECHA ENTREGA: 08 de octubre de 2025

Contenido del Manual Técnico

1. INTRODUCCIÓN	2
1.1 Descripción del Sistema.....	2
2. ARQUITECTURA DEL SISTEMA.....	2
3. TECNOLOGÍAS UTILIZADAS	4
3.1 Framework y Lenguaje.....	4
3.2 Base de Datos.....	4
3.3 Documentación de API.....	4
3.4 Herramientas de Desarrollo.....	4
4. ESTRUCTURA DEL PROYECTO	5
5. MODELOS DE DATOS	6
6. MIGRACIONES Y BASE DE DATOS.....	6
6.1 Crear Migración Inicial.....	6
6.2 Comandos Útiles de Entity Framework	6
7. ROLES Y PERMISOS.....	6
7.1 Cliente	6
7.2 Asociado (Dueño de Restaurante)	7
7.3 Administrador.....	7
8. SEGURIDAD	7
8.1 Autenticación	7
8.2 Autorización	7
8.3 Protección CSRF	7
8.4 Validación de Datos	7
9. FLUJOS DE TRABAJO.....	8
9.1 Flujo de Registro y Login	8
9.2 Flujo de Creación de Asociado (Admin)	8
9.3 Flujo de Pedido (Cliente).....	8
9.4 Flujo de Gestión de Pedido (Asociado).....	8

1. INTRODUCCIÓN

1.1 Descripción del Sistema.

TechFood Solutions es una aplicación web de delivery de comida, similar a plataformas como Uber Eats y Pedidos Ya. El sistema permite a los clientes realizar pedidos de comida de una gran variedad de restaurantes registrados en nuestra aplicación, a los administradores les permite gestionar la plataforma y a los asociados (dueños de restaurantes) administrar sus menús y pedidos.

1.2 Objetivos del Sistema

- Facilitar el proceso de pedido de comida a domicilio
- Gestionar múltiples restaurantes en una sola plataforma
- Proporcionar herramientas de administración eficientes
- Ofrecer una experiencia de usuario intuitiva y rápida

1.3 Alcance

El sistema tiene como alcance:

- Gestión de usuarios (Clientes, Administradores, Asociados)
- Catálogo de restaurantes y productos
- Carrito de compras y procesamiento de pedidos
- Panel de administración para agregar asociados
- Panel de gestión para asociados
- API RESTFUL para integración.

2. ARQUITECTURA DEL SISTEMA

2.1 Patrón Arquitectónico: MVC (Model-View-Controller)

Model (Modelo)

- Representa la lógica de negocio y datos
- Interactúa con la base de datos a través de Entity Framework
- Define las entidades del sistema

View (Vista)

- Interfaz de usuario en Razor Views
- Presenta los datos al usuario
- Separada por roles: Account, Admin, Asociado, Cliente, Cart, Home

Controller (Controlador)

- Gestiona las solicitudes HTTP
- Procesa la lógica de aplicación
- Coordina entre modelos y vistas

2.2 Capas del Sistema

1 CAPA DE PRESENTACIÓN

Views - Razor Pages

Interfaz de usuario que interactúa directamente con el usuario final. Incluye todas las vistas organizadas por roles: Account, Admin, Asociado, Cliente, Cart, Home y Shared.



2 CAPA DE CONTROLADORES

Controllers - Lógica de flujo

Gestiona las solicitudes HTTP del usuario y coordina entre las vistas y los servicios. Controla el flujo de la aplicación y maneja la lógica de presentación.



3 CAPA DE SERVICIOS

Services - Lógica de negocio

Contiene la lógica de negocio reutilizable y las reglas del sistema. Implementa operaciones complejas como gestión de pedidos, autenticación, notificaciones y validaciones.

4 CAPA DE ACCESO A DATOS

Models + Entity Framework

Define las entidades del dominio y maneja toda la interacción con la base de datos. Entity Framework ORM facilita el mapeo objeto-relacional y las operaciones CRUD.



5 BASE DE DATOS

SQL Server

Almacenamiento persistente de toda la información del sistema: usuarios, restaurantes, productos, pedidos, categorías y relaciones entre entidades.

Responsabilidades de cada capa:

- Presentación:** Experiencia visual e interacción del usuario
- Controladores:** Orquestación de peticiones y flujo de datos
- Servicios:** Reglas de negocio y operaciones complejas
- Acceso a Datos:** Abstracción de comunicación con BD
- Base de Datos:** Almacenamiento persistente y seguro

3. TECNOLOGÍAS UTILIZADAS

3.1 Framework y Lenguaje

- Framework: ASP.NET Core MVC
- Lenguaje: C# (última versión estable)
- Patrón de diseño: MVC (Model-View-Controller)

3.2 Base de Datos

- Motor: Microsoft SQL Server
- ORM: Entity Framework Core
- Migraciones: Code First Approach

3.3 Documentación de API

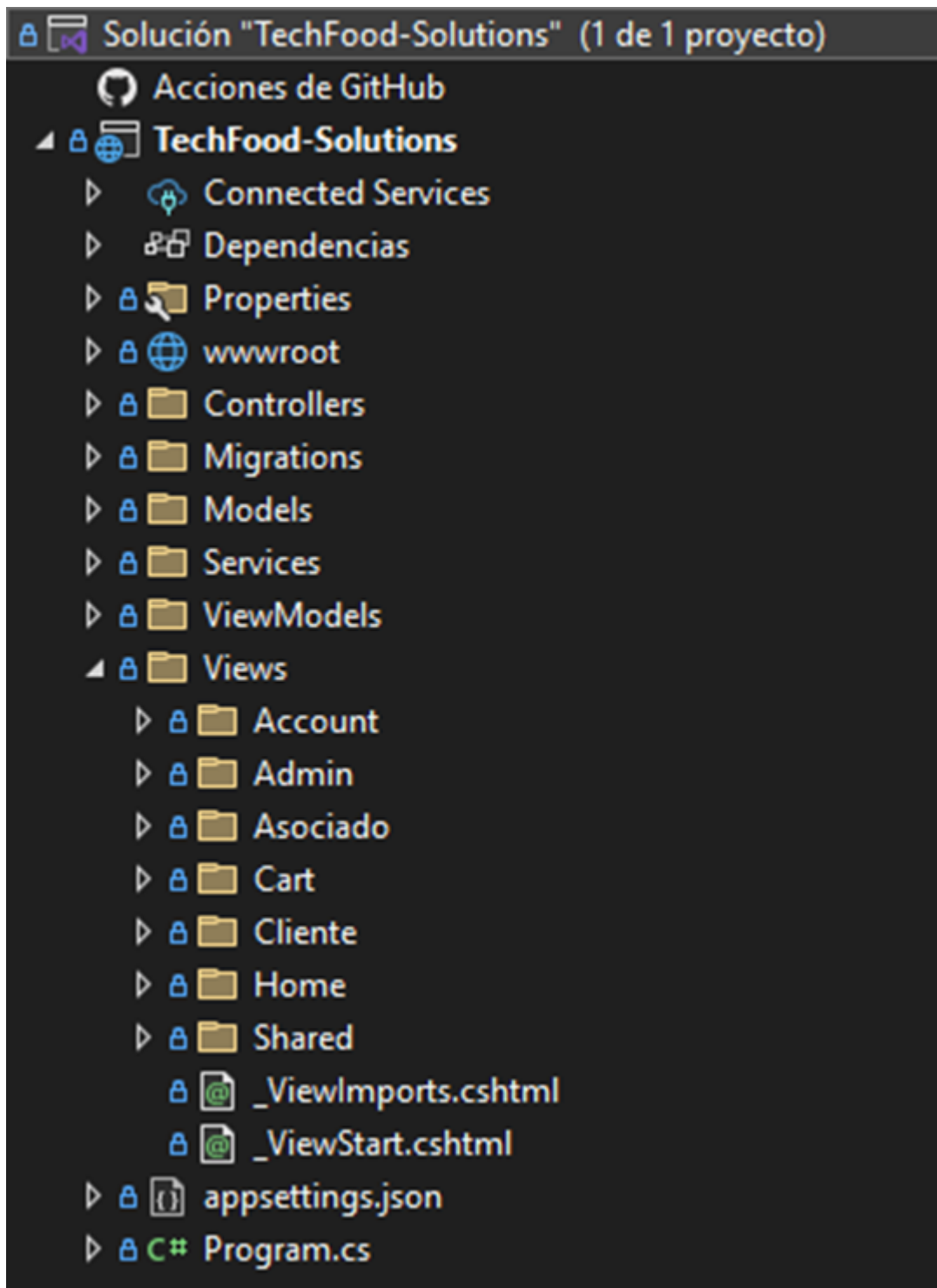
- Swagger/OpenAPI: Para documentación interactiva de endpoints
- Swashbuckle: Biblioteca para generar documentación Swagger

3.4 Herramientas de Desarrollo

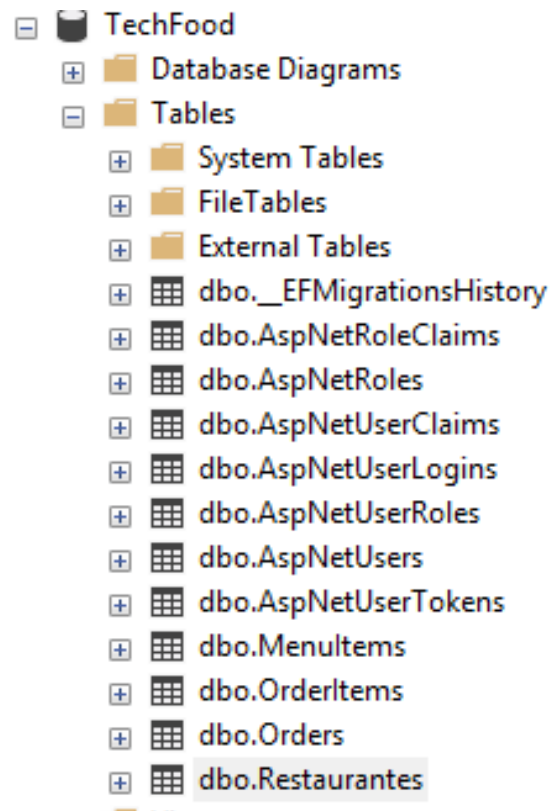
- Visual Studio 2022 o Visual Studio Code

- SQL Server Management Studio (SSMS)
- Postman (para pruebas de API)

4. ESTRUCTURA DEL PROYECTO



5. MODELOS DE DATOS



6. MIGRACIONES Y BASE DE DATOS

6.1 Crear Migración Inicial

- Add-Migration Initial
- Create Update-Database

6.2 Comandos Útiles de Entity Framework

- *# Crear nueva migración* (Add-Migration NombreMigracion)
- *# Aplicar migraciones* (Update-Database)
- *# Revertir última migración* (Update-Database PreviousMigrationName)
- *# Eliminar última migración* (Remove-Migration)
- *# Ver script SQL de migraciones* (Script-Migration)

7. ROLES Y PERMISOS

7.1 Cliente

Permisos:

- Ver restaurantes disponibles
- Ver menú de productos
- Agregar productos al carrito

- Realizar pedidos
- Ver historial de pedidos
- Actualizar perfil personal

7.2 Asociado (Dueño de Restaurante)

Permisos:

- Ver dashboard con estadísticas de su restaurante
- Gestionar productos de su restaurante (CRUD)
- Ver pedidos de su restaurante
- Actualizar estado de pedidos
- Modificar información del restaurante
- Ver reportes de ventas

7.3 Administrador

Permisos:

- Gestionar todos los usuarios del sistema
- Agregar nuevos restaurantes
- Asignar asociados a restaurantes
- Ver estadísticas globales del sistema
- Gestionar categorías de productos
- Activar/desactivar restaurantes
- Acceso completo al sistema

8. SEGURIDAD

8.1 Autenticación

- Uso de Cookie Authentication de ASP.NET Core
- Contraseñas hasheadas con BCrypt o Identity

8.2 Autorización

- Atributos [Authorize] en controladores
- Políticas de autorización basadas en roles
- Validación de permisos en cada acción

8.3 Protección CSRF

- Tokens anti-falsificación en formularios
- Validación automática de ASP.NET Core

8.4 Validación de Datos

- Data Annotations en ViewModels
- Validación del lado del servidor

- Sanitización de entrada de usuario

9. FLUJOS DE TRABAJO

9.1 Flujo de Registro y Login

1. Usuario accede a /Account/Register
2. Completa formulario con datos personales
3. Sistema valida datos y crea cuenta
4. Se genera hash de contraseña
5. Usuario es redirigido a página de login
6. Usuario ingresa credenciales en /Account/Login
7. Sistema valida y genera cookie de autenticación
8. Redirección según rol del usuario

9.2 Flujo de Creación de Asociado (Admin)

1. Admin accede a /Users
2. Completa el formulario con información del Asociado
3. Asocia ese asociado a un restaurante existente o crea uno nuevo
4. Sistema valida y guarda asociado y restaurante en BD
5. Confirmación y redirección a lista de restaurantes

9.3 Flujo de Pedido (Cliente)

1. Cliente navega por restaurantes disponibles
2. Selecciona un restaurante
3. Agrega productos al carrito
4. Accede al carrito en /Cart
5. Verifica productos y cantidades
6. Procede al checkout
7. Ingresa su información para la entrega
8. Confirma pedido
9. Sistema crea pedido en BD con estado "Pendiente"
10. Se crea el pedido y queda guardado en registros

9.4 Flujo de Gestión de Pedido (Asociado)

1. Accede a /Asociado/ y le muestra el dashboard principal
2. Tiene acciones de editar restaurante, gestionar menú y agregar nuevo producto
3. Asociado puede realizar las mismas acciones que un Cliente, pero no las de un Administrador.