## Estructura General del Proyecto

```

C:\Users\DAVID\OneDrive\Escritorio\SistemaIndicadores\

├── SistemaIndicadores.API/     # Backend

├── SistemaIndicadores.Shared/  # Biblioteca compartida

├── SistemaIndicadores.WEB/     # Frontend

└── backup/                     # Copia de seguridad

```

## 1. SistemaIndicadores.API (Backend)

### Estructura

```

SistemaIndicadores.API/

├── Controllers/                # Controladores de la API

│   ├── AuthController.cs      # Autenticación y autorización

│   ├── CategoriasController.cs # Gestión de categorías

│   ├── IndicadoresController.cs # Gestión de indicadores

│   ├── CalculosController.cs   # Cálculos de indicadores

│   └── UsuariosController.cs   # Gestión de usuarios

├── Data/                       # Capa de acceso a datos

│   ├── DataContext.cs         # Contexto de Entity Framework

│   ├── Configurations/        # Configuraciones de entidades

│   └── Migrations/            # Migraciones de la base de datos

└── Properties/                 # Configuración del proyecto

    └── launchSettings.json    # Configuración de inicio

```

### Archivos Principales

- \*\*Program.cs\*\*: Punto de entrada de la API, configura servicios y middleware

- \*\*appsettings.json\*\*: Configuraciones generales (conexión BD, JWT, etc.)

- \*\*SistemaIndicadores.API.csproj\*\*: Referencias y configuración del proyecto

### Controladores

1. \*\*AuthController.cs\*\*

   - Ruta: `/api/auth`

   - Funciones: Login, registro, renovación de tokens

2. \*\*CategoriasController.cs\*\*

   - Ruta: `/api/categorias`

   - Funciones: CRUD de categorías

   - Métodos: GET, POST, PUT, DELETE

3. \*\*IndicadoresController.cs\*\*

   - Ruta: `/api/indicadores`

   - Funciones: CRUD de indicadores

   - Métodos: GET, POST, PUT, DELETE

4. \*\*CalculosController.cs\*\*

   - Ruta: `/api/calculos`

   - Funciones: Registro y consulta de cálculos

   - Métodos: GET, POST, PUT

## 2. SistemaIndicadores.Shared (Biblioteca Compartida)

### Estructura

```

SistemaIndicadores.Shared/

└── Entities/                   # Modelos de datos

    └── Models.cs              # Definición de entidades

```

### Entidades Principales (Models.cs)

1. \*\*Usuario\*\*

   - Propiedades: Id, Nombre, Email, Password, Rol

   - Relaciones: Calculos (1:N)

2. \*\*Categoria\*\*

   - Propiedades: Id, Nombre, Descripcion, Activo

   - Relaciones: Indicadores (1:N)

3. \*\*Indicador\*\*

   - Propiedades: Id, Nombre, Descripcion, Formula, UnidadMedida

   - Relaciones: Categoria (N:1), Calculos (1:N)

4. \*\*CalculoIndicadores\*\*

   - Propiedades: Id, ValorReal, ValorMeta, PorcentajeCumplimiento

   - Relaciones: Indicador (N:1), Usuario (N:1)

## 3. SistemaIndicadores.WEB (Frontend)

### Estructura

```

SistemaIndicadores.WEB/

├── Auth/                      # Componentes de autenticación

│   └── CustomAuthStateProvider.cs # Proveedor de autenticación

├── Shared/                    # Componentes compartidos

│   ├── MainLayout.razor      # Diseño principal

│   ├── NavMenu.razor         # Menú de navegación

│   ├── Loading.razor         # Componente de carga

│   └── Error.razor           # Manejo de errores

└── Pages/                    # Páginas de la aplicación

    ├── Index.razor           # Página principal

    ├── Login.razor           # Inicio de sesión

    ├── Indicadores/          # Páginas de indicadores

    ├── Categorias/           # Páginas de categorías

    └── Calculos/            # Páginas de cálculos

```

### Componentes Principales

1. \*\*MainLayout.razor\*\*

   - Diseño principal de la aplicación

   - Implementa tema claro/oscuro

   - Maneja la navegación principal

2. \*\*NavMenu.razor\*\*

   - Menú de navegación responsivo

   - Control de acceso basado en roles

   - Animaciones de transición

3. \*\*Loading.razor\*\*

   - Indicador de carga animado

   - Mensajes personalizables

   - Transiciones suaves

4. \*\*Error.razor\*\*

   - Manejo de errores personalizado

   - Opciones de reintento

   - Mensajes informativos

### Páginas Principales

1. \*\*Index.razor\*\*

   - Dashboard principal

   - Resumen de indicadores

   - Gráficos y estadísticas

2. \*\*Login.razor\*\*

   - Formulario de inicio de sesión

   - Validación de credenciales

   - Redirección inteligente

3. \*\*Indicadores/\*\*

   - Lista.razor: Vista general de indicadores

   - Crear.razor: Formulario de creación

   - Editar.razor: Formulario de edición

   - Detalles.razor: Vista detallada

## 4. Configuración y Archivos Globales

### Archivos de Configuración

1. \*\*SistemaIndicadores.sln\*\*

   - Solución completa del proyecto

   - Referencias entre proyectos

   - Configuración de compilación

2. \*\*docker-compose.yml\*\*

   - Configuración de contenedores

   - Servicios definidos

   - Variables de entorno

3. \*\*.gitignore\*\*

   - Exclusiones de control de versiones

   - Archivos temporales

   - Directorios de compilación

### Backup

```

backup/

└── [Copia completa del proyecto]

```

- Copia de seguridad completa

- Incluye todos los archivos fuente

- Mantiene la estructura original

## Tecnologías y Versiones

### Backend

- ASP.NET Core 8.0

- Entity Framework Core 8.0.1

- SQL Server (Base de datos)

- JWT para autenticación

### Frontend

- Blazor WebAssembly 8.0

- MudBlazor 6.11.2

- LocalStorage para persistencia

- Componentes personalizados

### Herramientas de Desarrollo

- Visual Studio 2022

- .NET SDK 8.0

- Git para control de versiones

- SQL Server Management Studio

## Flujo de Datos

1. \*\*Autenticación\*\*

   ```

   Cliente -> Login.razor -> AuthController -> JWT -> LocalStorage

   ```

2. \*\*Gestión de Indicadores\*\*

   ```

   Cliente -> Indicadores/\*.razor -> IndicadoresController -> DataContext -> SQL Server

   ```

3. \*\*Cálculos\*\*

   ```

   Cliente -> Calculos/\*.razor -> CalculosController -> DataContext -> SQL Server

   ```

## Seguridad

1. \*\*Autenticación\*\*

   - JWT con expiración

   - Refresh tokens

   - Almacenamiento seguro

2. \*\*Autorización\*\*

   - Roles definidos

   - Políticas de acceso

   - Validación por ruta

3. \*\*Datos\*\*

   - Validación de entrada

   - Sanitización de datos

   - Encriptación de contraseñas

## Mantenimiento

1. \*\*Backup\*\*

   - Copia completa en /backup

   - Restauración documentada

   - Versionamiento

2. \*\*Actualizaciones\*\*

   - Paquetes NuGet

   - Dependencias frontend

   - Scripts de migración

3. \*\*Monitoreo\*\*

   - Logs de aplicación

   - Métricas de rendimiento

   - Reportes de errores