

MANUAL DE PROGRAMADOR

Tecnologías aplicadas: ASP.NET Core MVC (.NET 8), Entity Framework Core, Identity, SQL Server (LocalDB), ClosedXML, QuestPDF, Bootstrap 5, Chart.js

1. Arquitectura del Sistema

1.1 Capas y componentes:

- **Presentación:** Archivos `.cshtml` también conocidos como Views Razor, los cuales son acompañados de Bootstrap 5 y layout compartido (`Views/Shared/_Layout.cshtml`).
- **Controladores:** Basado en una arquitectura MVC, son los encargados del flujo, validación y devolución de vistas y archivos.
- **Servicios** de dominio: *IIventarioService* y *InventarioService* son los encargados de encapsular las reglas de negocio relacionadas con el control de inventarios.
- **Datos:** *ApplicationDbContext* de la mano con EntityFramework core de la mano de migraciones, las cuales desembocan en SQL Server LocalDB en desarrollo.
- **Seguridad:** *ASP.NET Core Identity* cual es el encargado de administrar usuarios y roles, en conjunto con el proceso de log-in y log-out.
- **Reportes:** Se utilizaron las herramientas de *ClosedXML* y *QuestPDF* para la configuración y generación de reportes en Excel y formato PDF respectivamente.

1.2 Estructura de carpetas:

Controllers/

- HomeController (dashboard)
- ProductosController (CRUD y Reportes Excel/PDF)
- MovimientosController (Gestión de entradas/salidas y Reportes Excel/PDF)

- AlertasController (Gestor de stock bajo)
- UsuariosController (Gestor de roles y usuarios)

Services:

- IInventarioService
- InventarioService

Models:

- Producto
- MovimientoInventario
- TipoMovimiento
- MovimientoCreateVM
- DashboardVM
- UserListItemVM
- EditUserRolesVM

Data:

- ApplicationDbContext, IdentitySeed

Views:

- subcarpetas por controlador (Productos, Movimientos, Alertas, Usuarios, Home)

2. Configuración del entorno de Desarrollo

2.1 Prerrequisitos:

- Visual Studio 2022
- .NET SDK 8.x
- SQL Server LocalDB
- Git

2.2 appsettings.json:

```
{
  "ConnectionStrings": {
```

```
    "DefaultConnection":  
    "Server=(localdb)\\MSSQLLocalDB;Database=ProyectoCatedraDES;Trusted_Connection=True;MultipleActiveResultSets=true;TrustServerCertificate=True;Connect Timeout=30"  
  }  
}
```

2.3 Restaurar y aplicar migraciones:

En Package Manager Console: **Update-Database**

2.4 NuGet utilizados:

- Microsoft.EntityFrameworkCore.SqlServer
- Microsoft.EntityFrameworkCore.Tools
- Microsoft.AspNetCore.Identity.EntityFrameworkCore
- ClosedXML
- QuestPDF
- Microsoft.VisualStudio.Web.CodeGeneration.Design

3. Guía de ejecución y despliegue

3.1 Descargar y construir la solución al clonar el repositorio desde:
https://github.com/Franckalv/Inventory_System_Project.git

3.2 Una vez construida la solución, se recomienda realizar un Update-Database para aplicar migraciones y cargar la base de datos de manera local.

3.3 Se ejecuta la solución y al cargar el archivo web, iniciar sesión con el usuario de pruebas:

Correo electrónico: admin@ddb.local

Contraseña: Admin#12345

4. Explicación de la base del código

4.1 Entidades:

Entidades: Id, Codigo (único), Nombre, Descripción, Precio (decimal), StockActual, StockMinimo, RowVersion ([Timestamp] para concurrencia).

MovimientoInventario: Id, ProductId (FK), Tipo (Entrada/Salida), Cantidad, Fecha (UTC), UsuarioId, Comentario.

4.2 Servicios de dominio:

IIventarioService / InventarioService:

- RegistrarEntradaAsync: +stock y agrega movimiento (transacción).
- RegistrarSalidaAsync: valida stock, -stock y agrega movimiento.

4.3 Controladores clave:

- **ProductosController:** CRUD + validación de código único + concurrencia (RowVersion) + ExportarExcel/Pdf.
- **MovimientosController:** formularios CrearEntrada/CrearSalida + listado + ExportarExcel/Pdf con filtro productoid.
- **AlertasController:** consulta StockActual <= StockMinimo
- **UsuariosController:** listado de usuarios y edición de roles (Admin / Operador).

5.4 Dashboard:

- **HomeController.Index:** arma DashboardVM con KPIs, movimientos recientes y series de 7 días.
- Vista usa **Chart.js** (CDN) para el gráfico.

5. Anexos

- Credenciales demo: **admin@ddb.local / Admin#12345**