Proyecto Final (Grupal) — Pedidos360: Inventario y Pedidos B2B

Curso: SC-601 Programación Avanzada

· **Modalidad:** Grupos de hasta 5 estudiantes (*)

· **Tecnología sugerida:** ASP.NET Core MVC (C#), EF Core (Code-First + Migrations), SQL Server/SQLite, Bootstrap 5, jQuery (AJAX)

---

1) Objetivo

Desarrollar una **aplicación web MVC** para una pyme que permita **gestionar catálogo de productos, clientes e inventario**, **crear pedidos con cálculo de totales (subtotal, impuestos, descuentos)**, y **buscar productos en tiempo real** mediante **AJAX**. El sistema debe implementar **autenticación y autorización por roles** (Admin, Ventas, Operaciones), **validaciones cliente/servidor**, manejo de **errores** y **API** de apoyo para búsquedas y cálculo de totales.

---

2) Alcance funcional (MVP obligatorio)

2.1 Catálogo de productos

- CRUD de productos con: **Nombre**, **Categoría**, **Precio**, **Impuesto (%)**, **Stock**, **Imagen** (obligatoria al crear), **Activo**.
- Listado con **paginación** y **filtros** (por nombre y categoría).
- Validaciones: nombre requerido, precio > 0, stock ≥ 0.

2.2 Clientes

- CRUD de clientes con: **Nombre**, **Cédula/Jurídica**, **Correo**, **Teléfono**, **Dirección**.
- Búsqueda por nombre/cédula y validaciones básicas (correo válido, campos requeridos).

2.3 Pedidos

- Crear pedido seleccionando cliente.
- Agregar productos mediante **autosuggest AJAX** (búsqueda por nombre/código).
- Para cada ítem: **cantidad**, **precio unitario**, **descuento** (porcentaje o fijo). unidad de medida
- **Cálculo automático** de **Subtotal**, **Impuestos** y **Total** en **tiempo real** (AJAX).

- Al confirmar pedido: **disminuye stock**, guarda totales (auditoría) y registra usuario/fecha.
- Ver detalle de pedido. (Exportar a PDF/Excel = *opcional* que suma nota.)

2.4 Seguridad y errores

- **Roles**:
    - **Admin**: acceso total.
    - **Ventas**: crea/consulta pedidos, ve catálogo; **no** elimina productos.
    - **Operaciones**: gestiona inventario (ajustes de stock), ve pedidos.
- Autenticación con **Identity**; acciones protegidas con [Authorize] según rol.
- Vistas de error personalizadas (**404/500**) y manejo básico de excepciones.
- Validación **cliente/servidor** (DataAnnotations + jQuery unobtrusive).

---

3) API y endpoints mínimos (para AJAX)

- **GET** /api/productos/buscar?q=… → devuelve hasta 10 coincidencias { id, nombre, precio, impuesto, stock }.
- **POST** /api/pedidos/calcular → recibe líneas [{ productoId, cantidad, descuento }] y responde { subtotal, impuestos, total }.
- Seguridad básica: solo usuarios autenticados pueden consumir estos endpoints.

---

4) Modelo de datos sugerido

- **Producto**(Id, Nombre, CategoriaId, Precio, ImpuestoPorc, Stock, ImagenUrl, Activo)
- **Categoria**(Id, Nombre)
- **Cliente**(Id, Nombre, Cedula, Correo, Telefono, Direccion)
- **Pedido**(Id, ClienteId, UsuarioId, Fecha, Subtotal, Impuestos, Total, Estado)
- **PedidoDetalle**(Id, PedidoId, ProductoId, Cantidad, PrecioUnit, Descuento, ImpuestoPorc, TotalLinea)
- **Usuario** (Identity por defecto + **Rol**)

Normalización mínima: **3FN**. Los totales se recalculan pero se **persisten** en Pedido/Detalle para auditoría.

---

5) Hitos y entregables

Ajustar fechas según calendario del curso.

Hito 1 — Avance funcional

- **Entregables**:
    - Diagrama **ER** + justificación de normalización.
    - Solución MVC base, **Migrations** iniciales.
    - CRUD funcional de **Productos** y **Clientes** (validaciones y listados con filtros/paginación).
    - Prototipo de vistas (layout/partials) y **story map** con criterios de aceptación.
- **Evidencia**: demo corta (5–7 min).

Hito 2 — Integración y seguridad

- **Entregables**:
    - **Identity + Roles** aplicados en controladores/acciones.
    - **API + AJAX**: búsqueda de productos y cálculo de totales en vivo.
    - Manejo de **errores** (404/500) y logging básico.
    - Pruebas funcionales mínimas y **checklist de calidad**.
- **Evidencia**: demo enfocada en flujo de pedido.

Entrega final — Presentación y defensa

- **Repositorio Git** con historial visible de **todos** los integrantes.
- **Guía de despliegue** (README): pasos para clonar, update-database, correr; usuarios/roles de prueba.
- **Seeders/scripts** para datos iniciales.
- **Informe técnico** (máx. 10 páginas): propósito, modelo de datos, decisiones de diseño, evidencias de requisitos.
- **Demo en vivo** (10–12 min) + **defensa** (Q&A 5–8 min).

---

6) Criterios de evaluación (rúbrica)

Criterio	Descripción	%
Requerimientos funcionales (MVP)	CRUDs, flujo de pedidos con cálculo en vivo, actualización de stock, búsquedas AJAX	**30%**
Calidad de la base de datos	ER, normalización (≥3FN), integridad y consistencia de datos	**12%**
Seguridad y roles	Identity, autorización efectiva por rol, protección de endpoints	**10%**
Validaciones y UX	Validaciones cliente/servidor, mensajes claros, paginación/filtros, usabilidad	**10%**
API/AJAX	Endpoints implementados, consumo seguro, cálculo de totales	**10%**
Manejo de errores	Vistas 404/500, manejo de excepciones, logging básico	**6%**
Código y arquitectura	Estructura clara, convenciones, comentarios (mencionar patrones/SOLID si aplican)	**8%**
Documentación y despliegue	README, scripts/seeders, guía de instalación, usuarios de prueba	**6%**
Control de versiones	Commits distribuidos, PRs/revisiones, contribución real de cada miembro	**4%**
Presentación y defensa	Demo, claridad técnica, respuestas en Q&A	**4%**
Total		**100%**

Bono (hasta +5%): exportes PDF/Excel, bitácora/auditoría, reportes básicos de ventas por fecha/cliente, thumbnails de imágenes con validación de tamaño.

7) Checklist de aceptación (entrega final)

- **Productos**: CRUD con imagen obligatoria al crear; filtros y paginación.
- **Clientes**: CRUD con validaciones; búsqueda por nombre/cédula.
- **Pedidos**: autosuggest AJAX; cálculo en vivo de subtotal/impuestos/total; persistencia de totales.
- **Stock**: disminuye al confirmar pedido; no permite inventario negativo.
- **Roles**: Admin/Ventas/Operaciones con restricciones reales.
- **Validaciones**: DataAnnotations + cliente (no guarda datos inválidos).
- **API**: /api/productos/buscar y /api/pedidos/calcular con acceso autenticado.
- **Errores**: vistas personalizadas 404/500; manejo básico de excepciones.
- **DB**: ER + normalización explicada en el informe.
- **Repo**: commits de todos los integrantes; issues/PRs evidenciados.
- **README**: pasos de ejecución, usuarios de prueba, orden de revisión.

---

8) Requisitos no funcionales

- **Instalación express**: clonar → update-database → ejecutar → navegar. Si no compila/levanta con pasos claros, **no se evalúa**.
- **Datos de prueba**: 10–20 productos con precios/impuestos realistas y stock coherente; 5–10 clientes.
- **Notas sobre patrones/SOLID**: si se usan, **mencionarlos** en comentarios/README (no obligatorio implementarlos a fondo).

---

9) Roles sugeridos (equipo de 5)

1. **Líder técnico/arquitectura**: ruteo, layouts, estándares, code review.
2. **Back-end EF/Migrations**: modelo de dominio, repos/servicios, consultas y performance.
3. **Front-end MVC/UX**: vistas tipadas, helpers, validaciones, UX con Bootstrap/partials.
4. **API/AJAX**: endpoints, seguridad de la API, integración y pruebas de consumo.

5. **QA & Docs/DevOps**: pruebas funcionales, seeders/scripts, CI local, guía de despliegue.

Con 4 integrantes: fusionar (2) y (4).

---

10) Formato y reglas de entrega

- **Archivo**: ZIP Nombre_Apellido_Equipo_ProyectoFinal.zip con solución compilable.
- **Repositorio**: URL del Git usado en el desarrollo (historial visible).
- **Compilación automática**: si el proyecto no compila con los pasos indicados en el README, la nota puede ser **0**.
- **Presentación**: 10–12 min demo + 5–8 min Q&A. Cada integrante defiende una parte.

---

11) Guía para la demo (script sugerido)

1. Login como **Admin** y muestra catálogo con filtros/paginación.
2. Crear producto (imagen obligatoria) mostrando validaciones.
3. Crear cliente y buscarlo.
4. Crear pedido como **Ventas**: autosuggest, cambiar cantidades/descuentos, totales en vivo (AJAX).
5. Confirmar pedido y evidenciar **baja de stock**.
6. Ver detalle de pedido (si implementan PDF/Excel, mostrarlo aquí).

Anexos (plantillas útiles)

Usuarios de prueba (ejemplo):

- admin@demo.local / Passw0rd! → Rol: Admin
- ventas@demo.local / Passw0rd! → Rol: Ventas
- ops@demo.local / Passw0rd! → Rol: Operaciones

Semillas mínimas: 12 productos, 2 categorías, 8 clientes, 1 lista de impuestos.

Criterios de aprobación mínima: cumplir el **MVP** completo, pasar checklist, demo estable.

Consultas técnicas y dudas rápidas: usar el foro/Teams del curso con capturas, errores y pasos para reproducir. ¡Éxitos!

Prof. Raul