

### Primer Desafío Práctico

## Sistema de Tienda en Línea con Facturación Electrónica

Desarrollo de un **sistema de ventas en línea** con gestión de productos, carrito de compras y generación de facturas electrónicas, utilizando estructuras de datos en **C# (Windows forms)**.

### Objetivo:

Implementar una aplicación gráfica que permita:

- Gestionar productos (alta, baja, modificación).
- Simular compras con un carrito.
- Generar facturas electrónicas en formato estructurado (JSON/XML).
- Usar **árboles binarios (ABB y AVL)** para búsquedas eficientes y **montículos (Heap)** para prioridades (ej.: productos en oferta).

**Tiempo de Entrega:** El examen se defenderá grupos de 2-3 integrantes en la semana 12.  
Fecha límite de entrega 9 de abril.

## Enunciado del Problema

Una tienda en línea necesita un sistema para:

1. **Gestionar productos** (código, nombre, precio, categoría, stock).
2. **Simular compras** (agregar productos al carrito, calcular totales).
3. **Generar facturas electrónicas** (cumpliendo con formato estándar).
4. **Priorizar productos** (ej.: "Ofertas del día" usando un *heap*).

## Requisitos Funcionales

### 1. Gestión de Productos

- **Estructuras de datos:**
  - **Árbol AVL** para evitar códigos duplicados y búsqueda rápida por ID.
  - **ABB** para mostrar productos ordenados por nombre o precio.
- **Operaciones:**
  - Agregar/eliminar/editar productos.
  - Buscar por código, nombre o categoría.
  - Mostrar stock crítico (ej.: productos con menos de 5 unidades).

## 2. Carrito de Compras

- **Estructura:** Lista enlazada o arreglo dinámico.
- **Funciones:**
  - Agregar/remover productos.
  - Calcular subtotal, impuestos (IVA) y total.
  - Aplicar descuentos (ej.: cupones).

## 3. Facturación Electrónica

- **Formato:** Generar factura en **JSON o XML** con:

```
{
  "Fecha": "2025-03-30",
  "Cliente": "Nombre del cliente",
  "Productos": [
    { "Código": "001", "Nombre": "Laptop", "Precio": 1200, "Cantidad": 1 }
  ],
  "Total": 1200
}
```

- **Validaciones:**
  - No permitir facturar si el carrito está vacío.
  - Verificar stock antes de finalizar la compra.

## 4. Ofertas Prioritarias (Heap)

- Usar un **montículo maximizante** para mostrar productos en oferta (prioridad = % de descuento).
- Ejemplo:
  - Producto A: 30% descuento (alta prioridad).
  - Producto B: 10% descuento (baja prioridad).

---

## Requisitos Técnicos

### 1. Interfaz Gráfica (Windows Forms):

- Pestañas para:
  - **Productos** (CRUD + búsqueda).
  - **Carrito** (lista de productos + botones para modificar).
  - **Facturación** (vista previa + botón "Generar PDF/JSON").
- Gráficos con `System.Drawing` para mostrar estructuras (ej.: árbol de productos).

### 2. Estructuras de Datos:

- **Árbol AVL:** Para evitar códigos repetidos.
  - **Heap:** Para gestionar ofertas prioritarias.
  - **Lista enlazada:** Para el carrito de compras.
3. **Validaciones:**
- Código de producto debe ser único.
  - Precio y stock no pueden ser negativos.
  - No facturar sin productos en el carrito.

### Entrega:

- Código fuente + ejecutable (.zip).
- Informe PDF (3 páginas máx.) con:
  - Explicación de estructuras usadas.
  - Capturas de pantalla.
  - Ejemplo de factura generada.

### Ejemplo de Interfaz



```
[ Pestaña 1: Productos ]
- DataGridView (Lista de productos)
- Botones: Agregar/Editar/Eliminar
- Búsqueda (por código, nombre, categoría)

[ Pestaña 2: Carrito ]
- ListBox (Productos agregados)
- Label (Subtotal, IVA, Total)
- Botón "Generar Factura"

[ Pestaña 3: Ofertas ]
- Gráfico de Heap (Panel)
- Botón "Mostrar Oferta del Día"
```

### (Opcional)

- Guardar datos en archivos (ej.: .txt o .json).
- Integrar un generador de PDF para facturas (usando librerías como iTextSharp).

¡Éxitos!  

## Rúbrica de Evaluación

### Sistema de Tienda en Línea con Facturación Electrónica

Criterio	Excelente (5 pts)	Satisfactorio (3-4 pts)	En Desarrollo (1-2 pts)	No Cumple (0 pts)
<b>Funcionalidad Completa</b>	Todas las funciones (CRUD, carrito, facturación, ofertas) están implementadas y funcionan correctamente.	Falta 1-2 funciones o tienen errores menores.	Faltan 3 o más funciones o presentan errores graves.	No funciona ninguna función básica.
<b>Uso de Estructuras de Datos</b>	<b>ABB:</b> Búsqueda eficiente por nombre/precio. <b>AVL:</b> Manejo de IDs únicos sin errores. <b>Heap:</b> Priorización correcta de ofertas. <b>Listas:</b> Carrito dinámico con todas las operaciones.	<b>ABB/AVL:</b> Pequeños errores en búsquedas. <b>Heap:</b> Priorización funciona, pero con lógica imperfecta. <b>Listas:</b> Carrito funciona, pero con limitaciones.	<b>ABB/AVL:</b> Solo implementado parcialmente. <b>Heap:</b> No prioriza correctamente. <b>Listas:</b> Carrito básico sin todas las funcionalidades.	No utiliza estructuras de datos o son incorrectas.
<b>Interfaz Gráfica</b>	Interfaz intuitiva, con pestañas claras, gráficos de árboles y validaciones visuales.	Interfaz funcional, pero con diseño básico o falta de gráficos.	Interfaz confusa o sin elementos clave (ej.: sin carrito visible).	No tiene interfaz gráfica usable.
<b>Documentación</b>	Informe PDF claro con explicaciones, capturas y ejemplo de factura. Código bien comentado.	Informe incompleto o código con comentarios mínimos.	Documentación muy escasa o sin ejemplos.	No incluye documentación.