Práctica 5 — Comercio electrónico: Mini-tienda con carrito

1. Objetivo de la práctica

Desarrollar una mini-tienda online que permita gestionar productos, stock, usuarios y pedidos. El sistema debe incluir control de acceso por roles, carrito de compra mediante sesiones, subida de imágenes, uso de programación orientada a objetos y almacenamiento en base de datos.

El resultado debe ser una aplicación funcional, clara y bien estructurada.

2. Descripción general de la aplicación

La aplicación debe incluir:

- Un sistema de login con roles diferenciados.
- Un panel principal con acceso a productos, pedidos y gestión interna.
- Un CRUD de productos (según rol).
- Un carrito de compra almacenado en sesión.
- Un sistema de pedidos con control de stock.
- Subida de imágenes para los productos.
- Listados de productos y pedidos en tablas HTML.
- Clases PHP que representen usuarios, productos, carrito y pedidos.

3. Requisitos funcionales obligatorios

3.1. Autenticación y roles: el sistema debe tener, como mínimo, los siguientes roles:

Empresario

- Permisos máximos.
- CRUD completo de productos.
- Gestión de empleados (alta, baja, modificación).
- Ver todos los pedidos.
- Modificar el stock de cualquier producto.

Empleado

- CRUD de productos.
- Ver todos los pedidos.
- No puede gestionar usuarios.

Usuario/Cliente

- Ver productos.
- Añadir productos al carrito.

- Modificar cantidades del carrito.
- Realizar pedidos.
- Ver su historial de pedidos.

Requisitos generales del login

- Formulario de login con usuario y contraseña.
- Comprobación de credenciales contra la base de datos.
- Uso de sesiones para mantener la sesión iniciada.
- Páginas protegidas según rol.
- Página de logout que destruya la sesión.

3.2. Gestión de productos (CRUD completo)

(Empresario y empleado) Cada producto debe tener:

- Nombre
- Descripción
- Precio
- Stock
- Categoría
- Imagen (subida al servidor)

Los usuarios con permisos podrán:

- Crear nuevos productos
- Editar productos existentes
- Eliminar productos
- Listar todos los productos en una tabla HTML

3.3. Subida de imágenes

Al crear o editar un producto:

- Se debe permitir subir una imagen.
- La imagen se guardará en una carpeta del proyecto (por ejemplo, /imgs).
- En la base de datos solo se almacenará la ruta de la imagen.
- Validar tipo de archivo (jpg, png) y tamaño máximo.

3.4. Carrito de compra (sesiones)

El carrito debe almacenarse en una sesión y permitir:

- Añadir productos.

- Modificar cantidades.
- Eliminar productos del carrito.
- Calcular subtotal, IVA y total.
- Comprobar stock antes de confirmar el pedido.

El carrito puede representarse mediante: un array asociativo o un array de objetos Producto.

3.5. Gestión de pedidos

Cada pedido debe tener:

- Usuario que lo realiza
- Fecha del pedido
- Total
- Líneas de pedido (producto, cantidad, precio)

El sistema debe permitir:

- Confirmar un pedido desde el carrito.
- Guardar el pedido y sus líneas en la base de datos.
- Reducir el stock de los productos comprados.
- Mostrar un listado de pedidos:
    - Todos los pedidos (empresario y empleado)
    - Solo los propios (usuario/cliente)

3.6. Posibles estructuras de almacenamiento *(Tienen libertad si consideran otras formas de estructuración)*

A) Carrito como array. Ejemplo:

```
$carrito[id_producto] = [
  'nombre' => ...,
  'precio' => ...,
  'cantidad' => ...
];
```

O un array de objetos Producto.

B) Array de productos o pedidos

Al listar productos o pedidos:

- Cargar datos desde la BD
- Crear arrays para filtrar por categoría, precio, fecha, etc.

- Permitir ordenar por precio, nombre o fecha

3.7. Programación orientada a objetos. Debes crear, como mínimo, las clases:

Clase Usuario

- Propiedades básicas (id, nombre, rol)
- Constructor
- Métodos getters/setters

Clase Producto

- Propiedades: nombre, descripción, precio, stock, categoría, imagen
- Constructor
- Métodos getters/setters

Clase Carrito

- Propiedades: lista de productos
- Métodos: añadir, eliminar, actualizar cantidad, calcular total

Clase Pedido

- Propiedades: usuario, fecha, total
- Métodos: añadir línea, calcular total, guardar en BD

3.7. Validación de formularios

- Nombre del producto obligatorio
- Precio válido (numérico y positivo)
- Stock válido (entero ≥ 0)
- Categoría válida
- Imagen válida (tipo y tamaño)
- Datos del pedido correctos antes de confirmarlo

Se pueden usar expresiones regulares donde sea adecuado.

4. Extras opcionales

- Cupones de descuento aplicados al carrito
- Buscador avanzado por nombre, categoría o rango de precio
- Historial de pedidos detallado
- Productos destacados o recomendados
- Sistema de valoraciones de productos