

Estudiante					Fecha		
Horario	Dia		Hora		dd	mm	aa

Tipo de Evidencia Para Evaluar						Nota
Producto						

Objetivo.

Desarrollar una aplicación web funcional que simule un sistema de Carrito de Compras. Los estudiantes deberán aplicar y demostrar su comprensión de los siguientes conceptos fundamentales de la programación y el desarrollo web:

- **Programación Orientada a Objetos (POO):** Se debe diseñar una arquitectura de software basada en clases, como Cliente, Cuenta, CuentaCorriente y CuentaAhorros. El proyecto debe hacer un uso explícito de los pilares de la POO, especialmente la **herencia** y el **polimorfismo**, para modelar de manera eficiente la relación entre los diferentes tipos de cuentas y sus operaciones.
- **Manipulación del DOM (Modelo de Objetos del Documento):** La interacción del usuario con la aplicación (ingresar datos, hacer clic en botones, visualizar saldos, etc.) debe gestionarse dinámicamente utilizando JavaScript. Esto implica manipular el **DOM** para actualizar la interfaz en tiempo real sin recargar la página, mostrando la información del usuario y los resultados de las transacciones de manera fluida y responsiva.
- **Diseño de la Aplicación:** Se requiere crear una interfaz de usuario (**UI**) que sea profesional, intuitiva y fácil de usar. El diseño debe ser responsivo, adaptándose a diferentes tamaños de pantalla. La aplicación debe guiar al usuario a través de un flujo claro: desde el inicio de sesión o registro hasta la pantalla de transacciones. Un diseño limpio y bien estructurado garantizará una experiencia de usuario positiva.

Sistema de Simulación de Tienda Web de Productos Tecnológicos

El proyecto tiene como objetivo modelar la interacción entre clientes y un catálogo de productos tecnológicos, permitiendo realizar operaciones comunes de una tienda en línea como: visualizar productos, agregar al carrito, modificar cantidades, calcular el total y confirmar una compra. El desarrollo se llevará a cabo aplicando los principios de la **Programación Orientada a Objetos (POO)**.

Requisitos Técnicos

- **Entorno de Desarrollo:** El proyecto se implementará para la web utilizando las siguientes tecnologías:
 - **Frontend:** HTML5, CSS3 y JavaScript.
 - **Backend:** Se utilizará JavaScript para manejar la lógica de negocio y la persistencia de datos (aunque sea de forma simulada en memoria, sin una base de datos real).
 - **Diseño:** La interfaz de usuario debe tener un diseño profesional, moderno y responsivo, que sea intuitivo y fácil de usar para los clientes.
-

Requisitos Funcionales

El sistema deberá gestionar las siguientes funcionalidades:

- **Gestión de Clientes:**
 - **Registro de Usuarios:** Los nuevos clientes deben poder crear una cuenta, proporcionando un nombre, correo, contraseña e información personal.
 - **Inicio de Sesión:** Los usuarios registrados deben poder iniciar sesión con sus credenciales. El sistema debe validar el acceso y limitar los intentos a un máximo de tres.
 - **Modificación de Datos:** Los usuarios que hayan iniciado sesión deben poder actualizar su información personal.
- **Gestión de Productos y Carrito de Compras:**
 - **Catálogo de Productos:** Visualizar un listado de productos tecnológicos con nombre, imagen, descripción y precio.
 - **Agregar al Carrito:** Permitir que los clientes añadan productos a su carrito de compras.
 - **Modificar Cantidades:** Cambiar la cantidad de un producto en el carrito.
 - **Eliminar Productos:** Quitar un producto del carrito.
 - **Consulta del Total:** Mostrar en tiempo real el valor total a pagar según los productos seleccionados.
 - **Confirmar Compra:** Generar una orden de compra con el detalle de productos adquiridos.

Diseño Orientado a Objetos (POO)

La arquitectura del sistema se basará en las siguientes clases:

Clases Principales

- **Cliente:** Representará a cada usuario del sistema.
 - **Atributos:**
 - nombre, correo, direccionEnvio
 - usuario, contraseña
 - **Métodos:**
 - registrar()
 - iniciarSesion()
 - modificarDatos()
- **Producto:** Clase base (superclase) que representa un producto disponible en la tienda.
 - **Atributos:**
 - idProducto, nombre, descripcion, precio, stock
 - **Métodos:**
 - mostrarInfo()
 - actualizarStock()
- **CarritoCompra:** Representará el carrito de compras del cliente.
 - **Atributos:**
 - listaProductos[]
 - total
 - **Métodos:**
 - agregarProducto()
 - eliminarProducto()
 - actualizarCantidad()
 - calcularTotal()
- **Orden:** Representará la compra final realizada por el cliente.
 - **Atributos:**
 - numeroOrden, cliente, listaProductos[], total, fecha, estado
 - **Métodos:**
 - generarOrden()
 - mostrarResumen()

Clases Heredadas

- **Laptop:** Hereda de la clase Producto.
 - Atributos adicionales: procesador, memoriaRAM, almacenamiento.
- **Smartphone:** Hereda de la clase Producto.
 - Atributos adicionales: sistemaOperativo, tamañoPantalla, camara.

- **Accesorio:** Hereda de la clase Producto.
 - Atributos adicionales: tipoAccesorio (ejemplo: audífonos, mouse, teclado).

Herencia y Polimorfismo

- Se utilizará la **herencia** para reutilizar atributos y métodos comunes en la clase base **Producto**.
- Se aplicará **polimorfismo** al sobrescribir el método `mostrarInfo()` en Laptop, Smartphone y Accesorio, de manera que cada tipo de producto muestre información específica.

Interfaz de Usuario (UI)

Se debe desarrollar una interfaz de usuario para interactuar con el sistema, con las siguientes vistas principales:

1. **Pantalla de Inicio:**
 - Campos para Usuario y Contraseña.
 - Botón de "Iniciar Sesión".
 - Opción "Registrar Usuario".
2. **Catálogo de Productos:**
 - Listado de productos con imagen, descripción y precio.
 - Botón "Agregar al Carrito".
3. **Carrito de Compras:**
 - Vista de productos añadidos, con opción de modificar cantidad o eliminar.
 - Visualización del total en tiempo real.
 - Botón "Confirmar Compra".
4. **Orden Generada:**
 - Resumen de compra con productos, cantidades, total y fecha.

🔗 El nombre de la aplicación será a elección del equipo.



Lista de Chequeo: Evaluación de Proyecto "Tienda Web de Productos Tecnológicos"

A continuación, se presenta un desglose de los puntos clave a evaluar en el proyecto, siguiendo los requisitos técnicos, funcionales y principios de la Programación Orientada a Objetos (POO).

1. Maquetación del Proyecto

Diseño y Usabilidad:

- ☐ La interfaz de usuario (UI) tiene un diseño profesional, moderno y atractivo.
- ☐ La navegación entre las diferentes vistas (Inicio, Catálogo, Carrito, Orden) es intuitiva y clara.
- ☐ Se presenta un ícono o logotipo representativo de la tienda.

Responsividad:

- ☐ El diseño de la interfaz se adapta correctamente a diferentes tamaños de pantalla (escritorio, tablet, móvil).

Elementos de la UI:

- ☐ Pantalla de Inicio: contiene campos para Usuario y Contraseña, opción de Registro y botón para Iniciar Sesión.
- ☐ Pantalla de Catálogo: muestra productos con nombre, imagen, descripción y precio.
- ☐ Carrito de Compras: permite visualizar los productos añadidos, modificar cantidades, eliminar productos y ver el total.
- ☐ Orden Generada: presenta un resumen de la compra realizada.

2. Requisitos Técnicos

Tecnologías Utilizadas:

- ☐ Uso correcto de **HTML5** y **CSS3** para la estructura y el estilo.
- ☐ Toda la lógica de negocio está implementada en **JavaScript**.

Persistencia de Datos:

- ☐ Los datos de clientes, productos y órdenes se gestionan de forma simulada en memoria.

3. Requisitos Funcionales

Gestión de Clientes:

- ☐ El sistema permite registrar nuevos usuarios con nombre, correo y contraseña.
- ☐ El inicio de sesión valida credenciales.
- ☐ El acceso está limitado a un máximo de tres intentos fallidos.
- ☐ Los usuarios pueden modificar sus datos personales después de iniciar sesión.

Gestión de Productos y Carrito de Compras:

- ☐ El catálogo muestra la información básica de los productos.
- ☐ Los clientes pueden agregar productos al carrito.
- ☐ Es posible modificar la cantidad de un producto en el carrito.
- ☐ Es posible eliminar un producto del carrito.
- ☐ El total se actualiza en tiempo real.
- ☐ Al confirmar la compra, se genera una orden con el resumen de la transacción.

4. Diseño Orientado a Objetos (POO)

Clases Principales:

- ☐ Se implementa la clase **Cliente** con sus atributos y métodos (registrar(), iniciarSesion(), modificarDatos()).
- ☐ Se implementa la clase **Producto** como superclase, con atributos y métodos base.
- ☐ Se implementa la clase **CarritoCompra** con su lógica (agregar, eliminar, actualizarCantidad, calcularTotal).
- ☐ Se implementa la clase **Orden** para generar y mostrar el resumen de compras.

Clases Heredadas:

- ☐ Se implementa una clase **Laptop** que hereda de Producto.
- ☐ Se implementa una clase **Smartphone** que hereda de Producto.
- ☐ Se implementa una clase **Accesorio** que hereda de Producto.

Principios de POO:

- ☐ Se utiliza la **herencia** para compartir atributos y métodos entre clases derivadas de Producto.
- ☐ Se aplica **polimorfismo** al sobrescribir métodos como mostrarInfo() en las distintas subclases.