

## Descripción

**Nombre de la Plataforma:** ShopAll

**Descripción:** ShopAll es una plataforma de comercio electrónico diseñada para ofrecer una amplia variedad de productos a consumidores de todo el mundo. La plataforma se enfoca en proporcionar una experiencia de compra intuitiva y segura, y está destinada a crecer a medida que más vendedores y compradores se unan a ella.

## Especificación

**Usuarios Esperados:**

1. **Compradores:** Usuarios individuales interesados en comprar productos en la plataforma. Deben poder navegar por categorías de productos, buscar artículos específicos, agregar productos al carrito, completar transacciones y dejar reseñas.
2. **Vendedores:** Personas y empresas que desean vender sus productos en la plataforma. Los vendedores deben poder crear perfiles de tienda, agregar productos, gestionar inventario, establecer precios y recibir notificaciones sobre pedidos.

**Tipos de Productos:** La plataforma ofrece una amplia variedad de productos, incluyendo electrónica, moda, hogar, deportes, belleza y más. Cada producto tiene atributos como nombre, descripción, precio, categoría y detalles específicos según el tipo de producto.

**Transacciones:** Las transacciones son el corazón de la plataforma. Los usuarios pueden agregar productos al carrito, proceder al pago y completar la compra. Se deben registrar los detalles de cada transacción, incluidos los productos comprados, el monto total, la dirección de envío y la información de pago.

## Requerimientos

### 1. Normalización y Estructura de Bases de Datos:

En el modelo de la BD se deben aplicar las reglas de normalización de bases de datos, hasta 3NF para minimizar la redundancia. También se debe considerar la integridad referencial, el uso correcto de claves primarias y foráneas, así como la implementación de índices para optimizar la recuperación de datos.

### 2. Principios de Programación Orientada a Objetos:

Se debe considerar la aplicación de los cuatro principios fundamentales de POO: encapsulamiento, abstracción, herencia y polimorfismo. Contar con una adecuada estructura de las clases y objetos para confirmar la correcta aplicación de estos principios, el uso de interfaces y clases abstractas, y la implementación de patrones de diseño para resolver problemas comunes, así como empaquetamiento de las clases en las capas adecuadas.

### **3. Implementación de Spring Boot:**

Se debe considerar una adecuada configuración y estructura del proyecto Spring Boot, incluyendo el uso de anotaciones como `@RestController`, `@Service`, `@Repository`, y la separación de responsabilidades. Así como, la configuración de los beans y el uso de diferentes capas como controladores, servicios, y acceso a datos.

### **4. Manejo de Excepciones y Validación de Datos:**

Es necesario considerar una estrategia de manejo de excepciones globales con `@ControllerAdvice` o `@RestControllerAdvice` y la implementación de validaciones de entrada usando `@Valid` y la anotación `@Validator`. Así como, la cobertura de casos de error y la claridad de los mensajes de error proporcionados al cliente.

### **5. Cobertura de Pruebas Unitarias:**

Se deben generar las pruebas unitarias con JUnit para garantizar que cubran una amplia gama de casos de uso. Comprobar la implementación de Mockito para simular dependencias. Las pruebas deben ser capaces de ejecutarse de manera independiente y en cualquier entorno.

### **6. Seguridad con Spring Security y JWT:**

Considerar la configuración de Spring Security para la autenticación y autorización, y cómo se generan, distribuyen y validan los JWT. Verificar la seguridad de las rutas, el cifrado de contraseñas, y la gestión de permisos para diferentes roles de usuarios.

### **7. Uso y Manejo de Git:**

Considerar el uso efectivo de Git, incluyendo la estructura de ramas, estrategias de merge, y resolución de conflictos. Comprobar el uso de commits semánticos, mensajes descriptivos, y la aplicación de pull requests para revisión de código antes de la integración en la rama principal.

## Evaluación

#	Descriptor	Porcentaje	Cumplimiento
1	Estructura de la BD	10%	
2	POO	10%	
3	SpringBoot	40%	
4	Excepciones	10%	
5	Pruebas Unitarias	10%	
6	Seguridad	10%	
7	GIT	10%	
TOTAL		100%	