

**Harmony Dealer**  
**HD E-commerce**  
**Documento Diseño de Servicios (Contrato):**  
**Versión 1.0**

## Revisión

Fecha	Versión	Descripción	Autor
04-06-2024	1.0	Primera especificación del Servicio para notificaciones para el módulo de notificaciones	Manuel Alejandro Peñarete Paez
10-07-2024	2.0	Revisión y corrección	Manuel Alejandro Peñarete Paez
16-08-2024	3.0	Revisión y corrección	Manuel Alejandro Peñarete Paez

## Aprobaciones

Nombre / Rol	Fecha	Firma

# Documento de Especificación del Servicio de Entidad Incidente

## 1. Breve Descripción

Este contrato contiene los acuerdos para la definición del Web Service Harmony Dealer el cual es un servicio que gestiona la venta de instrumentos musicales en la plataforma de ecommerce Harmony Dealer. El servicio permite a los usuarios realizar compras, gestionar su carrito, procesar pagos y administrar sus pedidos.

## 2. Versión del servicio

1.0

## 3. Dueños del servicio

De acuerdo con una matriz RACI (Responsible, Accountable, Consulted, Informed> se indican las áreas de negocio o personas responsables del servicio:

Responsible (R)	Equipo de Desarrollo de Software de Harmony Dealer.
Accountable (A)	Gerente de Tecnología de Harmony Dealer.
Consulted (C)	Equipo de Ventas y Atención al Cliente.
Informed (I)	Dirección General y Equipo de Marketing.

## 4. Descripción de la interfaz de los servicios

En esta sección se describen las funciones u operaciones que se pueden implementar usando los servicios entregados por el servicio:

url: <https://host:4200/harmomydealer/v1>

Nombre de la función o servicio	Descripción de la función	Información de entrada	Información de salida	Método HTTP
<b>createOrder</b>	Permite procesar y crear una nueva orden de compra.	El servicio recibe un Request JSON con los objetos: - UserId - CartItems (object-array) - PaymentDetails - ShippingAddress	El response del servicio muestra el estado de la creación de la orden, incluyendo un OrderId y el total del pedido.	POST
<b>getOrderById</b>	Permite obtener los detalles de una orden por su ID.	El servicio recibe el parámetro orderId en la URL.	El response contiene los detalles de la orden, incluyendo productos, precios, estado de la orden, y fecha de creación.	GET
<b>getProductList</b>	Permite obtener una lista de todos los productos disponibles en la tienda.	N/A	El response contiene un array de productos con detalles como nombre, precio, y disponibilidad.	GET

<b>processPayment</b>	Permite procesar el pago de una orden utilizando un servicio externo como PayPal.	El servicio recibe un Request JSON con los objetos: - OrderId - PaymentMethod - PaymentDetails	El response incluye el estado de la transacción, confirmación del pago, y redirección a la página de confirmación de la compra.	POST
<b>deleteProductById</b>	Permite eliminar un producto del catálogo por su ID (solo accesible por administradores).	El servicio recibe el parámetro productId en la URL.	El response confirma si el producto fue eliminado correctamente o muestra errores en caso de fallos.	DELETE

## 5. Manejo de requerimientos no funcionales

En esta sección se describen los requerimientos no funcionales que deben soportarse alrededor del servicio:

Aspecto no funcional	Requerimiento	Detalle del requerimiento
<b>Invocación</b>	Tipo de invocación	Sincrónica para transacciones críticas como pagos, asíncrona para operaciones como el envío de correos de confirmación.
<b>Seguridad</b>	Autenticación y autorización	Se requiere autenticación JWT para usuarios y administradores, con roles diferenciados (USER, ADMIN).
<b>Conexión</b>	Seguridad en la conexión	El servicio debe garantizar conexiones seguras usando HTTPS y encriptación TLS.
<b>Rendimiento</b>	Tiempo de respuesta	El tiempo de respuesta del servicio no debe superar los 2 segundos para las operaciones de consulta de productos y pedidos.
<b>Escalabilidad</b>	Soporte de carga	El servicio debe ser capaz de soportar un incremento del 30% en la carga de usuarios y pedidos en periodos de alta demanda.
<b>Disponibilidad</b>	Alta disponibilidad	El servicio debe estar disponible al menos el 99.9% del tiempo, con un plan de recuperación ante fallos implementado.
<b>Compatibilidad</b>	Integración de terceros	El servicio debe poder integrarse con servicios externos como plataformas de pago (PayPal, Stripe) y servicios de envío.

## 6. Manejo de requerimientos funcionales

### 6.1 Configuración en base de datos.

Motor de base de datos: PostgreSQL

Esquema: ecommerce

Tablas:

**users:** Almacena la información de los usuarios (id, nombre, email, contraseña encriptada, rol, fecha de creación).

**products:** Almacena los productos disponibles para la venta (id, nombre, descripción, precio, stock, categoría, imagen).

**orders:** Registra las órdenes de compra realizadas (id, user\_id, total, estado, fecha de creación).

**order\_products:** Relaciona productos con órdenes (order\_id, product\_id, cantidad, precio\_unitario, cantidad\_total).

**categories:** Define las categorías de productos (id, nombre, descripción).