

# Análisis de Requerimientos – Backend eCommerce

## 1. Introducción

- **Nombre del proyecto:** Somostiendamas.com.ar
- **Objetivo:** Desarrollar un backend que permita gestionar productos, usuarios, compras y pagos de manera eficiente y segura.
- **Alcance:** Implementación de una API REST en Spring Boot con base de datos relacional.

## 2. Requerimientos Funcionales

### *Productos*

- CRUD de productos (crear, leer, modificar, borrar).
- Categorías y subcategorías.
- Gestión de stock.

### *Carrito de Compras*

- Agregar, actualizar y eliminar productos.
- Persistencia del carrito por usuario.

### *Órdenes y Compras*

- Creación de órdenes de compra.
- Estado de compra (pendiente, pagado, enviado, entregado, **devuelto o reintegrado**).
- Historial de compras.
- **Selección de forma de envío:**
  - Domicilio.
  - Retiro en sucursal a través de empresas de envío.
  - Retiro en domicilio por el vendedor.

### *Pagos*

- Integración con pasarelas de pago.
- Validación de pagos y actualización de estado.
- **Métodos de pago admitidos:**
  - Transferencia bancaria.
  - Tarjeta de débito.
  - Tarjeta de crédito.

## Comunicación y Notificaciones

- **Envío de notificaciones por correo electrónico:**
  - Confirmación de compra.
  - Cambio de estado del pedido.
  - Recordatorios de pago si la orden está pendiente.
  - **Notificaciones adicionales:**
    - Ofertas y descuentos.
    - Estado del envío.

## Reportes y Análisis de Desempeño

- **Para vendedores:**
  - Reporte de ventas por período (diario, semanal, mensual).
  - Productos más vendidos.
  - Ingresos generados.
  - Clientes recurrentes.
- **Para administradores:**
  - Estadísticas generales de la plataforma.
  - Volumen de ventas.
  - Cantidad de órdenes y pagos procesados.

## Devoluciones y Derecho al Arrepentimiento

- **Botón de arrepentimiento** para que el cliente pueda devolver el producto dentro de los **10 días hábiles**.
- Proceso automatizado de devolución:
  - Solicitud de devolución generada a través del botón.
  - Actualización de estado del producto a **devuelto**.
  - Reembolso del dinero (si corresponde) tras recibir el producto en buenas condiciones.

## Requerimientos No Funcionales

- **Seguridad:**
  - Autenticación mediante **JWT** (JSON Web Tokens) para protección de rutas y sesiones.
  - **Cifrado de contraseñas** utilizando **BCrypt** o **PBKDF2** a través de **Spring Security**.
  - Protección contra ataques comunes (CSRF, XSS, etc.) utilizando las configuraciones de **Spring Security**.
  - Implementación de roles y permisos para usuarios (cliente, vendedor, administrador) con control de acceso basado en roles (RBAC).
- **Escalabilidad:**
  - **Soporte para alto tráfico** mediante el uso de **caché** (como **Redis** o **Ehcache**) para almacenar datos de consulta frecuentes y reducir la carga de la base de datos.

- **Manejo de múltiples instancias** utilizando técnicas como **load balancing** en un entorno de microservicios (si es necesario en el futuro).
- Uso de **asíncronía** y colas (con **RabbitMQ** o **Kafka**) para procesar tareas que no requieren respuesta inmediata (por ejemplo, envío de correos electrónicos, notificaciones, etc.).
- **Disponibilidad:**
  - **Alta disponibilidad** mediante la implementación de **clústeres** de bases de datos y servicios redundantes.
  - Utilización de **monitoreo** en tiempo real (con herramientas como **Prometheus** y **Grafana**) para detectar posibles problemas en la API.
  - Despliegue en **infraestructura de alta disponibilidad**, como **AWS**, **Google Cloud** o **Azure**.
- **Rendimiento:**
  - Optimización de consultas SQL mediante el uso de **índices** adecuados y la revisión de las consultas generadas por **JPA**.
  - Implementación de **paginación** y **filtrado** eficiente para consultas que devuelven grandes cantidades de datos.
  - Uso de **caché** para mejorar el tiempo de respuesta de las consultas más frecuentes y reducir la carga en la base de datos.

## 4. Modelo de Datos (ERD resumido)

- Usuario (id, nombre, email, contraseña, rol).
- Producto (id, nombre, descripción, precio, stock, categoría\_id).
- Carrito (id, usuario\_id, producto\_id, cantidad, precio).
- Orden (id, usuario\_id, total, estado).
- Pago (id, orden\_id, método, estado, monto).

Este puede ser ampliado según los requerimientos solicitados y necesarios.

## 5. Tecnologías y Herramientas

- **Backend:** Spring Boot, JPA/Hibernate.
- **Base de datos:** MySQL / PostgreSQL.
- **Seguridad:** Spring Security + JWT.

## **Tiempo Estimado de Realización**

### ***1. División de Tareas y Estimaciones de Tiempo:***

#### **Análisis y Planificación: 1 a 2 semanas**

- Definición de requerimientos y arquitectura

#### **Diseño de la Base de Datos: 1 a 2 semanas**

- Modelado y estructuración

#### **Desarrollo del Backend: 4 a 8 semanas**

- Implementación de lógica de negocio y APIs

#### **Integración de Pasarela de Pago: 1 a 2 semanas**

- Ej.: MercadoPago

#### **Implementación de Seguridad: 1 a 2 semanas**

- Autenticación, autorización y protección

#### **Pruebas y Aseguramiento de Calidad (QA): 2 a 4 semanas**

- Pruebas funcionales y de seguridad

#### **Despliegue y Configuración: 1 semana**

- Preparación del entorno de producción

#### **Documentación y Capacitación: 1 a 2 semanas**

- Manuales y formación para el equipo

*Tiempo Total Estimado: 12 a 22 semanas (aproximadamente 3 a 5 meses).*

## Resumen del Desarrollo Backend:

1. **API REST:**
  - Desarrollo de la API REST para gestionar la lógica de negocios.
  - Implementación de los endpoints necesarios para operaciones CRUD de productos, usuarios, pagos, órdenes, etc.
2. **Base de Datos:**
  - Diseño y creación de tablas para almacenar la información.
  - Gestión de relaciones entre entidades.
  - Implementación de respaldo de datos y copias de seguridad periódicas para evitar pérdidas en caso de imprevistos.
3. **Seguridad:**
  - Implementación de protocolos de seguridad como JWT para autenticación.
  - Encriptado de datos sensibles (como contraseñas y pagos) utilizando tecnologías adecuadas.
  - Protección de la API para usuarios, vendedores y administradores con roles y permisos.
4. **Deploy:**
  - Despliegue de la API y el proyecto en un servidor de producción.
  - Configuración de servidores web (Nginx/Apache) y bases de datos para mantener la página funcionando.
  - Integración con servicios de nube (por ejemplo, DonWeb) para alojar la aplicación.
5. **Post Producción:**
  - Mantenimiento por 3 meses tras la implementación.
  - Resolución de problemas y corrección de errores que puedan surgir durante el uso.
  - Monitoreo continuo del sistema y la base de datos para asegurar su correcto funcionamiento.

### 1. Alcance y Complejidad del Proyecto

El desarrollo del sistema incluye una **API REST** que gestionará toda la lógica de negocios, permitiendo la administración de usuarios, productos, pagos, órdenes y envíos de manera eficiente y segura. Esto garantizará un sistema estable y de alto rendimiento.

Además, se implementará una **base de datos optimizada**, diseñada para manejar grandes volúmenes de información, asegurando la integridad y el respaldo de los datos.

En términos de **seguridad**, se aplicarán protocolos avanzados, incluyendo autenticación mediante **JWT**, cifrado de contraseñas y protección de datos sensibles. Esto es fundamental para resguardar la información personal y financiera de los usuarios.

El sistema también será **escalable**, es decir, estará preparado para manejar un alto número de usuarios y transacciones sin afectar el rendimiento. Para ello, se planificará una infraestructura adecuada y se realizarán pruebas de carga.

Finalmente, el proyecto incluye el **despliegue en la nube** (por ejemplo, en DonWeb), asegurando que el sistema quede completamente operativo. También se ofrece un período de **mantenimiento post-producción** de 3 meses para resolver cualquier posible imprevisto que pueda surgir.

---

## 2. Tiempo Estimado

El desarrollo del sistema tomará entre **3 y 5 meses**, organizados en diferentes fases:

- ✓ **Análisis y planificación:** Definición de requerimientos y arquitectura del sistema.
- ✓ **Desarrollo:** Implementación de la API REST y base de datos.
- ✓ **Pruebas y validación:** Evaluación del rendimiento, seguridad y estabilidad.
- ✓ **Despliegue y configuración:** Publicación del sistema en la nube y ajustes finales.
- ✓ **Mantenimiento postproducción:** Resolución de posibles imprevistos.

Este plazo asegura que el sistema quede bien implementado, sin apresurar fases críticas como las pruebas y la optimización de seguridad.

---

## 3. Mantenimiento Post-Producción y Optimización

Luego de la entrega, se incluye un período de **mantenimiento postproducción de 3 meses**, lo que significa que ante cualquier inconveniente, el soporte técnico estará disponible para realizar ajustes y solucionar problemas sin costos adicionales.

Además, se continuará optimizando la **base de datos y la infraestructura** para garantizar un rendimiento óptimo, asegurando que el sistema pueda escalar y adaptarse a futuras necesidades.

---

## 4. Costo y Justificación

El precio del proyecto se calcula en base a:

- ◆ La cantidad de horas de desarrollo necesarias para implementar una solución completa y segura.
- ◆ El nivel de responsabilidad requerido para diseñar un sistema robusto y confiable.
- ◆ La inclusión de funcionalidades avanzadas como pagos en línea, seguridad con cifrado de datos y la gestión eficiente de órdenes y productos.

Este presupuesto se encuentra dentro de los valores estándar del mercado, considerando la complejidad del desarrollo y los servicios incluidos.

---

## 5. Beneficios y Valor Añadido

- ✓ **Sistema de calidad:** Seguro, eficiente y preparado para escalar en el futuro.
- ✓ **Soporte postproducción:** Se brinda mantenimiento durante 3 meses ante cualquier imprevisto.
- ✓ **Despliegue completo:** La plataforma quedará funcionando al 100% en la nube, lista para el uso de clientes, vendedores y administradores.

Con este enfoque, el cliente recibe una solución llave en mano, con garantía de funcionamiento y soporte técnico en caso de ser necesario.

*El costo de este proyecto refleja tanto la complejidad como el tiempo necesario para desarrollar una solución robusta y escalable. En el proceso de desarrollo, se trabajará en cada una de las áreas del sistema, desde la creación de una API REST segura hasta la optimización de la base de datos para manejar alto tráfico. Además, el despliegue en la nube y el soporte postproducción de tres meses garantizan que la aplicación funcione correctamente desde el primer día y esté lista para escalar cuando sea necesario. Los precios están alineados con el mercado actual, tomando en cuenta la experiencia necesaria para llevar a cabo este proyecto de manera eficiente.*

### ***Precio final de la API puesta en producción (cloud server no incluido):***

- 3.500.000 pesos (2900USD aproximadamente)
- Cloud Server: entre 3500 y 16000 pesos mensuales aproximadamente (marzo 2025) dependiendo de las características que se requieran (CPU, memoria, almacenamiento, transferencia).