## **1. Introducción**

EcoMarket SPA es una empresa chilena dedicada a la venta de productos ecológicos y sostenibles. Debido a su crecimiento y expansión, el sistema monolítico actual ha evidenciado problemas de rendimiento y disponibilidad, afectando la eficiencia operativa y la experiencia del cliente. Este informe presenta una solución basada en arquitectura de microservicios con MySQL para mejorar la escalabilidad y estabilidad del sistema.

## **2. Análisis de Requerimientos**

### **2.1 Requerimientos Funcionales**

* **Gestión de Usuarios:** Creación, actualización y eliminación de cuentas.
* **Inventario:** Administración de productos, stock y pedidos.
* **Ventas y Facturación:** Procesamiento de ventas, devoluciones y emisión de facturas.
* **Logística y Envíos:** Seguimiento de pedidos y optimización de rutas de entrega.
* **Interacción con Clientes:** Historial de pedidos, reseñas y soporte.
* **Generación de Reportes:** Informes sobre ventas y rendimiento.

### **2.2 Entrevistas a Perfiles de Usuarios**

* **Administrador:** Necesita herramientas de control y reportes detallados.
* **Vendedor:** Requiere una interfaz rápida y fácil para registrar ventas.
* **Cliente:** Busca una plataforma segura y eficiente para realizar compras

### **2.2 Requisitos No Funcionales**

* **Escalabilidad:** Capacidad de crecimiento sin afectar el rendimiento.
* **Disponibilidad:** Sistema accesible en todo momento.
* **Seguridad:** Implementación de cifrado de datos y autenticación segura.
* **Mantenimiento:** Facilidad de actualización y depuración.
* **Eficiencia:** Optimizar tiempos de respuesta y carga del sistema.

## 

## 

## **3. Análisis del Sistema Actual**

### **3.1 Evaluación del Sistema Monolítico**

* **Dificultad en mantenimiento y actualizaciones.**
* **Baja escalabilidad ante aumento de usuarios.**
* **Altas tasas de fallos debido a sobrecarga en el servidor.**

### **3.2 Identificación de Problemas**

* **Cuellos de botella en la base de datos centralizada.**
* **Dependencia de un solo servidor.**
* **Baja tolerancia a fallos.**

## **4. Diseño de la nueva arquitectura**

## **4.1 Arquitectura de Microservicios**

La arquitectura de microservicios es un enfoque de diseño de software en el que una aplicación se divide en pequeños servicios independientes que se comunican entre sí a través de APIs. Cada microservicio está centrado en una función específica de la aplicación y puede ser desarrollado, desplegado y escalado de forma independiente.

### **4.2 Explicación**

Es como tener una aplicación que está formada por varios "mini-proyectos" autónomos. Cada uno de estos proyectos se encarga de una tarea específica, como el manejo de usuarios o la gestión de pagos. Cada microservicio puede actualizarse, probarse y escalar de manera separada sin afectar al resto de la aplicación.

### **4.3 Beneficios**

* **Escalabilidad independiente:** Cada microservicio puede escalar de forma independiente según las necesidades de la funcionalidad específica.
* **Desarrollo ágil:** Los equipos pueden trabajar en diferentes microservicios sin interferir entre sí, lo que facilita el desarrollo y la implementación continua.
* **Resiliencia:** Si un microservicio falla, no afecta a toda la aplicación, lo que mejora la disponibilidad general.
* **Flexibilidad tecnológica:** Cada microservicio puede estar construido usando diferentes tecnologías según lo que sea más adecuado para esa parte de la aplicación.

### 

### **4.4 Desafíos**

* **Complejidad en la gestión:** La comunicación entre microservicios y la gestión de sus dependencias puede ser difícil de manejar.
* **Sobrecarga de infraestructura:** Se necesita más infraestructura y herramientas para manejar los microservicios de manera eficiente (como orquestación y monitoreo).
* **Consistencia de datos:** Mantener la coherencia de datos entre diferentes microservicios distribuidos puede ser complicado.

### **4.5 Empresas que la utilizan**

* **Netflix:** Utiliza microservicios para manejar la enorme cantidad de contenido y usuarios, permitiendo escalabilidad y resiliencia.
* **Amazon:** Su arquitectura está basada en microservicios, lo que les permite desarrollar y desplegar nuevas funciones de manera independiente y rápida.
* **Spotify:** Emplea microservicios para gestionar sus diferentes funcionalidades de forma eficiente y escalable.

### **4.6 Estrategia de Microservicios para EcoMarket SPA**

Se propone dividir el sistema en módulos independientes con APIs REST para comunicación:

* **Microservicio de Autenticación y Usuarios:** Manejo de usuarios y roles.
* **Microservicio de Inventario:** Administración de productos, stock y pedidos.
* **Microservicio de Ventas y Facturación:** Registro y procesamiento de compras.
* **Microservicio de Logística y Envíos:** Gestión de envíos y proveedores.
* **Microservicio de Interacción con Clientes:** Gestión de reseñas e historial de pedidos.
* **Microservicio de Reporte y Análisis:** Generación de informes de ventas, inventario y rendimiento.

### **4.7 Herramientas Tecnológicas**

* **Lenguaje:** Java con Spring Boot.
* **Base de Datos:** MySQL.
* **Mensajería:** RabbitMQ para comunicación entre microservicios.
* **Orquestación:** Kubernetes para despliegue y escalabilidad.
* **Seguridad:** JWT para autenticación de usuarios.

### **4.8 Diagramas**

* **Casos de uso:** Representación de interacciones entre usuarios y sistema.
* **Clases:** Modelado de entidades y relaciones.
* **Despliegue:** Arquitectura en la nube y comunicación entre microservicios.

## **5. Planificación de la Migración**

### **5.1 Fases de Migración**

1. **Análisis y diseño**: Definir los microservicios y la infraestructura necesaria.
2. **Implementación gradual**: Desarrollar e implementar cada microservicio por separado.
3. **Pruebas y validación**: Evaluación de rendimiento y estabilidad del nuevo sistema.
4. **Despliegue y monitoreo**: Implementación final y monitoreo para corrección de errores.

### **5.2 Identificación de Riesgos y Mitigación**

* **Incompatibilidad con sistemas heredados** → Implementación de APIs intermedias.
* **Tiempo de inactividad** → Despliegue progresivo y pruebas en entornos de desarrollo.
* **Problemas de seguridad** → Uso de autenticación robusta y cifrado de datos.

## **6. Consideraciones Éticas**

* **Privacidad de datos:** Cumplimiento de normativas de protección de datos personales.
* **Impacto en el empleo:** Capacitar a los empleados en la nueva tecnología.
* **Responsabilidad en el despliegue:** Realizar pruebas de seguridad antes de la implementación.

## **7. Conclusión**

La implementación de una arquitectura de microservicios permitirá a EcoMarket SPA mejorar la escalabilidad, disponibilidad y seguridad de su sistema. La transición gradual minimizará riesgos, asegurando una experiencia fluida para clientes y empleados. Esta transformación digital es un paso clave para el crecimiento sostenible de la empresa

Diagrama De Casos De Uso:



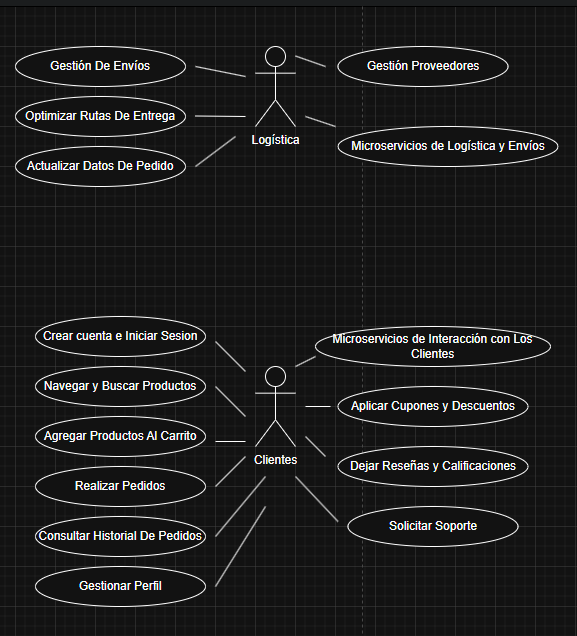


Diagrama De Clases

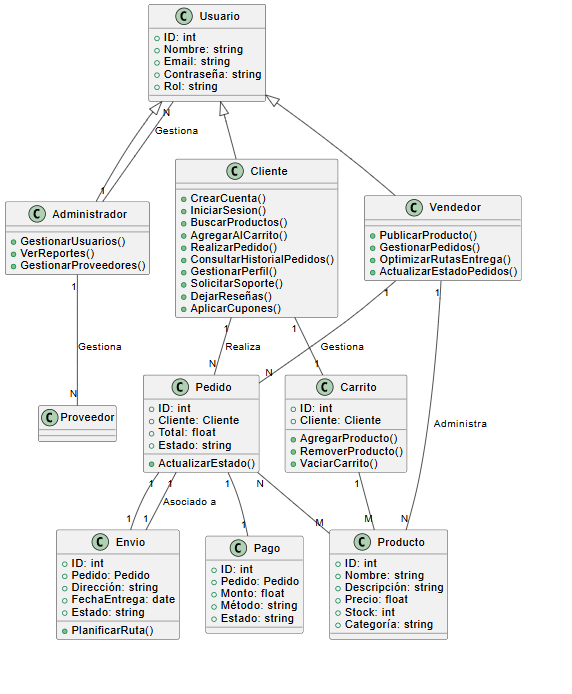


Diagrama De Despliegue:

