Cambio de Arquitecturas de Software en EcoMarket

Ricardo Gaete, Damian, Felipe

Intituto Profesional Duoc UC de Maipú

Abril de 2025

1. Resumen

En este informe se quiere lograr analizar un problema de una compañia actual conocida como EcoMarket SPA, la cual ha tenido problemas de escalabilidad y rendimiendo lo cual ha causado problemas en la efeciencia general de la compañia y la experiencia al usuario. En base a esta problematica, se propone una solución mediante la implementación de un sistema de arquitectura de microservicios, el cual se destaca por la capacidad de escalabilidad, resilencia y adaptabilidad.

Este informe se mostrará la documentación necesaria para llegar a esta solución, como el levantamiento de requisitos funcionales y no funcionales de la empresa mediante las historias de usuario desprendidas de los diversos actores de la empresa, investigación relacionada con los microservicios, plan de migración de un sistema a otro, identificación de posibles fallos y/o riesgos de la migración con su plan de mitigación correspondiente, elección de estrategias a usar en la implementación de los microservicios, consideración de principios eticos en el desarrollo y la explicación de las multiples herramientas empleadas en la elaboración y la ejecución del proyecto.

1. Introducción

EcoMarket SPA es una empresa chilena que ha experimentado un crecimiento sostenido gracias a su enfoque en la comercialización de productos ecológicos y sostenibles. Desde su apertura inicial en el Barrio Lastarria de Santiago, la empresa ha expandido exitosamente sus operaciones con nuevas sucursales en Valdivia y Antofagasta, consolidando así su presencia a nivel nacional. No obstante, este crecimiento ha traído consigo nuevos desafíos tecnológicos, ya que el sistema monolítico que actualmente sustenta sus operaciones ha comenzado a evidenciar problemas críticos de rendimiento, disponibilidad y escalabilidad, afectando directamente la eficiencia interna y la experiencia del cliente.

Ante esta situación, el presente informe tiene como objetivo diseñar una solución tecnológica basada en microservicios, que permita a la empresa adaptarse a sus nuevas necesidades operativas. Esta solución considera el uso de MySQL como motor de base de datos, permitiendo una estructura modular, eficiente y fácilmente escalable, capaz de soportar la evolución del negocio en el corto y largo plazo.

A lo largo del documento se abordarán los distintos elementos clave del proyecto: el análisis de requerimientos funcionales y no funcionales, la evaluación del sistema monolítico actual, el diseño detallado de la nueva arquitectura basada en microservicios, y un plan de migración estratégico. Además, se incluirá una evaluación ética sobre los desafíos relacionados con la privacidad de los datos, la automatización y su impacto social, así como el uso de herramientas colaborativas como Trello, que ha facilitado el trabajo en equipo durante el desarrollo del proyecto.

Este informe busca no solo resolver los problemas actuales de EcoMarket SPA, sino también sentar las bases para una infraestructura sólida, ética y sostenible, que impulse su crecimiento a futuro y fortalezca su compromiso con la innovación y la responsabilidad empresaria.

1. Estrategia de Microservicios a Implementar

Con el fin de modernizar la arquitectura tecnológica de **EcoMarket SPA** y superar las limitaciones del sistema monolítico actual, se propone una estructura basada en **microservicios independientes pero integrados**, cada uno encargado de una funcionalidad específica del sistema. Esta segmentación permitirá una mayor escalabilidad, facilidad de mantenimiento, resiliencia ante fallos y una mejor distribución de la carga de trabajo.

### **Usuarios**: Gestión de autenticación y roles

Encargado de manejar toda la información relacionada con los usuarios del sistema (clientes, empleados, administradores, etc.).

Funciones clave:

- Registro e inicio de sesión de usuarios (clientes y personal).  
- Recuperación y cambio de contraseñas.  
- Control de acceso basado en roles (cliente, vendedor, administrador, soporte, etc.).  
- Gestión de perfiles de usuario (datos personales, direcciones, historial de actividad).

### **Inventario**: Administración de productos y stock

Administra los productos disponibles en la tienda y su cantidad.

Funciones clave:  
- Alta, baja y modificación de productos.  
- Definición de categorías y atributos de productos (nombre, descripción, precio, imágenes).  
- Control de stock (ingreso, egreso, niveles mínimos de inventario).  
Sincronización de stock con el sistema de pedidos y ventas.

### **Ventas**: Procesamiento de transacciones y facturación

Gestiona el proceso de compra y el registro de transacciones.

Funciones clave:  
- Procesamiento de pagos (integración con pasarelas de pago).  
- Emisión de facturas o boletas electrónicas.  
- Registro y seguimiento de cada venta realizada.  
- Aplicación de descuentos, promociones y cupones.

### **Pedidos**: Creación, seguimiento y actualización de órdenes

Controla los pedidos realizados por los usuarios.

Funciones clave:  
- Generación de órdenes de compra tras una venta.  
- Asignación de estado a cada pedido (pendiente, procesado, enviado, entregado, cancelado).  
- Asociación del pedido con el usuario, productos, y dirección de entrega.  
Notificaciones automáticas al usuario sobre el estado del pedido.

### **Logística**: Control de envíos y optimización de rutas

Optimiza y gestiona la entrega física de los productos.

Funciones clave:  
- Coordinación con servicios de entrega (propios o externos).  
- Planificación de rutas para distribución eficiente.  
- Control de entregas fallidas o reintentos.  
Integración con sistemas de seguimiento (tracking).

### **Soporte**: Canal de atención al cliente

Brinda asistencia a los clientes y gestiona sus solicitudes.

Funciones clave:  
- Sistema de tickets para gestionar consultas, quejas o devoluciones.  
- Chat en tiempo real o correo para atención personalizada.  
- Base de conocimientos o sección de preguntas frecuentes (FAQs).  
- Asignación de casos a agentes de soporte.

### **Reportes**: Análisis de datos para administradores y gerentes

Proporciona información para la toma de decisiones basada en datos.

Funciones clave:  
- Reportes de ventas diarias, semanales y mensuales  
- Análisis de stock e inventario  
- Evaluación del rendimiento del personal o del soporte  
- Visualización de métricas clave (KPIs) en dashboards

### **Proveedores**: Control y gestión de relaciones con abastecedores

Administra los productos, contactos y entregas relacionadas a los proveedores.

Funciones clave:  
- Registro y gestión de proveedores (contactos, productos, acuerdos).  
- Seguimiento de órdenes de compra a proveedores.  
- Control de entregas pendientes o retrasadas.  
- Evaluación del desempeño de proveedores.

1. Herramientas a utilizar

Para llevar a cabo la solución tecnológica basada en microservicios para EcoMarket SPA, se utilizarán diversas herramientas que facilitarán tanto el desarrollo como la implementación de la nueva arquitectura. A continuación, se describen las principales herramientas que se emplearán:

### **MySQL**

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) que se utilizará para gestionar la información relacionada con los productos, clientes, ventas, pedidos y demás datos transaccionales. MySQL será la base de datos principal para el sistema, ya que ofrece robustez, alta disponibilidad y eficiencia en la gestión de grandes volúmenes de datos. Además, su integración con los microservicios permitirá una arquitectura eficiente.

### **Postman (para probar APIs)**

**Postman** es una herramienta excelente para probar y depurar APIs. Se puede utilizar para hacer peticiones HTTP a los microservicios y asegurarse de que todo funcione correctamente. Es una herramienta que ayudará a entender cómo los microservicios interactúan a través de las API.

**Función en el proyecto**: Probar las APIs para cada microservicio (usuarios, pedidos, inventario).

### **Git y GitHub**

**Git** es una herramienta de control de versiones que permitirá llevar un registro de los cambios en el código y colaborar con otros miembros del equipo. **GitHub** es una plataforma que permite alojar tu repositorio de Git en la nube y facilitar la colaboración.

**Función en el proyecto**: Gestionar el código de los microservicios y la interfaz frontend, colaborar con otros miembros del equipo y hacer seguimiento de los cambios.

### **Trello (para gestión de tareas y proyectos)**

**Trello** es una herramienta de gestión de proyectos muy visual y fácil de usar. Esta permitirá organizar las tareas relacionadas con el proyecto, desde la creación de microservicios hasta el desarrollo del frontend. Esta es capaz de crear tableros para cada fase del proyecto y asignar tareas a los miembros del equipo.

**Función en el proyecto**: Planificar las tareas, dividir el trabajo en sprints, y hacer seguimiento del progreso del proyecto.

1. **Visual Studio**

**Visual Studio es un IDE (Entorno de Desarrollo Integrado) muy potente, ampliamente utilizado en el desarrollo de aplicaciones, y soporta una variedad de lenguajes como JavaScript, CSS, HTML, Python, entre otros.**

**Función en el proyecto: Crear microservicios para gestionar usuarios, pedidos, inventario, y otras funcionalidades clave del sistema EcoMarket SPA. Visual Studio tiene plantillas para crear APIs RESTful rápidamente, lo que facilita el desarrollo de microservicios.**

### **Django (para desarrollo de microservicios)**

**Django** es un framework de desarrollo web de alto nivel para Python, que fomenta el desarrollo rápido y limpio de aplicaciones. Aunque originalmente está diseñado para aplicaciones monolíticas, Django es perfectamente viable para desarrollar microservicios, especialmente si se utiliza junto con **Django REST Framework** (DRF). DRF permite crear **APIs RESTful** de manera eficiente, lo cual es ideal para una arquitectura de microservicios, donde cada microservicio puede ser accesible a través de una API.

**Función en el proyecto**: Django se puede utilizar para desarrollar microservicios que gestionan diferentes funcionalidades de la tienda EcoMarket SPA, tales como la gestión de usuarios, el manejo de productos e inventarios, o el procesamiento de pedidos. Con el uso de Django REST Framework, cada uno de estos servicios puede exponer sus propios endpoints API, comunicándose con otros microservicios o con el frontend de manera independiente. Además, Django tiene una estructura robusta para la gestión de bases de datos y la autenticación, lo que facilita la creación de servicios seguros y eficientes.

### Enfoques Eticos en el Proyecto

En el desarrollo de una arquitectura de microservicios para EcoMarket SPA, se deben considerar varios aspectos éticos fundamentales para asegurar el funcionamiento responsable y justo de la plataforma, así como la protección de la privacidad y el bienestar de los empleados y clientes. A continuación se describen los principios éticos clave que se deben integrar en el proyecto:

1. **Privacidad de datos**: Se deben implementar buenas prácticas de seguridad, como encriptación de datos, autenticación robusta y cumplimiento de la Ley de Protección de Datos Personales en Chile.
2. **Responsabilidad en el despliegue**: Garantizar que los servicios sean seguros y confiables antes de su puesta en producción, evitando interrupciones que afecten a los clientes.
3. **Impacto en el empleo**: La automatización de procesos con microservicios podría reemplazar algunas tareas manuales, por lo que la empresa debe ofrecer capacitación y reasignación de roles dentro de la organización.
4. **Transparencia y equidad**: Se debe evitar la discriminación algorítmica en los sistemas de recomendación y precios dinámicos.
5. **Analisis del Sistema Monolitico Actual**

EcoMarket SPA ha experimentado un crecimiento que ha llevado a una serie de desafíos asociados con su sistema monolítico. Los problemas identificados incluyen:

1. **Rendimiento Deficiente: Con la expansión de la empresa y el aumento de la demanda, el sistema puede no estar optimizado para manejar el volumen creciente de transacciones, lo que resulta en tiempos de carga lentos y una experiencia de usuario insatisfactoria.**

**B) Problemas de Disponibilidad: Un fallo en cualquier parte del sistema puede afectar a todo el funcionamiento de la empresa. Es decir, si uno de los módulos (por ejemplo, el de inventario) presenta un error, puede hacer que toda la aplicación deje de funcionar.**

**C) Escalabilidad Limitada: Un sistema monolítico es difícil de escalar, ya que cualquier cambio o mejora requiere una implementación completa de la aplicación. Esto puede responder lentamente a las necesidades cambiantes del mercado.**

**D) Dificultad en la Implementación de Nuevas Funcionalidades: La integración de nuevas características requiere cambios en el código existente, lo cual puede ser complejo y arriesgado, aumentando la posibilidad de introducir fallos.**

**E) Falta de Flexibilidad: Personalizar o actualizar partes del sistema se torna complicado, ya que todos los módulos están interconectados. Esto puede llevar a una obsolescencia del software si no se puede adaptar a nuevas tecnologías o métodos.**

**¿Por Qué Estos Problemas Afectan a la Empresa?**

**Los problemas de rendimiento y disponibilidad directamente impactan la experiencia del cliente. En un sector donde la sostenibilidad y la atención al cliente son fundamentales, cualquier deficiencia en el servicio puede resultar en la pérdida de clientes y reputación. Además, el retraso en la implementación de nuevas características puede hacer que EcoMarket no logre aprovechar oportunidades de mercado, limitando su competitividad.**

1. **Plan de Migración**
2. **Análisis y Preparación: Auditoría del Sistema Actual: Evaluar las capacidades, limitaciones y cuellos de botella del sistema monolítico actual. Recopilar datos sobre el uso, el rendimiento y las necesidades de los usuarios.**

**Definición de Microservicios: Con base en la auditoría, confirmar los microservicios identificados:**

**1. Usuarios 2. Inventario 3. Ventas 4. Pedidos 5. Logística 6. Soporte 7. Reportes 8. Proveedores**

**Tecnologías y Herramientas: Seleccionar las tecnologías y herramientas específicas para el desarrollo de microservicios, como bases de datos, lenguajes de programación, frameworks y herramientas de monitoreo.**

1. **Diseño de la Arquitectura**

**Arquitectura de Microservicios: Diseñar la arquitectura de los microservicios, identificando las interfaces de cada uno, los endpoints API y cómo se comunicarán entre sí.**

**Base de Datos: Planificar el esquema de cada base de datos y la estrategia de segregación de datos.**

**Gateway API: Implementar un Gateway API para centralizar el acceso a los microservicios, gestionar la autenticación (por ejemplo, usando JWT) y dirigir las peticiones adecuadamente.**

**C) Desarrollo de Microservicios**

**Implementación Gradual: Comenzar el desarrollo de los microservicios utilizando metodologías ágiles, priorizando aquellos que tienen un impacto significativo en las operaciones diarias (por ejemplo, Inventario, Ventas y Pedidos).**

**Desarrollo de API: Crear APIs RESTful para los microservicios. Asegurarse de que estén documentadas para facilitar futuras integraciones.**

**Pruebas Unitarias e Integración: Realizar pruebas unitarias y de integración en cada microservicio para garantizar su funcionalidad y compatibilidad.**

**D) Estrategia de Migración de Datos**

**Planificación de la Migración de Datos: Definir cómo se migrarán los datos del sistema monolítico a los nuevos microservicios. Esto puede incluir:**

**Exportación de Datos: Extraer datos del sistema monolítico, transformarlos si es necesario, y cargarlos en las bases de datos de los microservicios correspondientes.**

1. **Implementación y Pruebas**

**Despliegue Controlado: Desplegar uno o varios microservicios en la nube o en un entorno de prueba.**

**Pruebas de Usuario: Invitar a un grupo de usuarios finales a probar los nuevos microservicios y recopilar feedback. Esto ayudará a identificar problemas y áreas de mejora antes de un despliegue a gran escala.**

**Monitoreo de Rendimiento: Establecer herramientas de monitoreo (login, métricas de rendimiento, etc.) para realizar un seguimiento de la estabilidad y el rendimiento de los microservicios implementados.**

1. **Despliegue Completo**

**Migración Gradual: Implementar los microservicios restantes en etapas, minimizando interrupciones en las operaciones.**

**Desactivación de Funcionalidades del Monolito: A medida que se implementan nuevos microservicios y se verifica su funcionamiento, desactivar gradualmente las funcionalidades en el sistema monolítico.**

**Capacitación del Personal: Capacitar a los empleados y usuarios finales sobre el uso**

1. **Plan de Mitigación**
2. **Empleado de Ventas:**

**Microservicio de Ventas:**

****Riesgo**:** Errores en el procesamiento de pagos y facturación incorrecta.

****Plan de Mitigación**:** Implementar un sistema de validación de pagos y facturas con auditorías y reportes automáticos.

1. **Gerente de Tienda:**

**Microservicio de Inventario:**

****Riesgo**:** Inconsistencias en el inventario debido a sincronización o errores en el sistema.

****Plan de Mitigación**:** Implementar auditorías periódicas del inventario y mecanismos de reconciliación de datos.

1. **Administrador de Sistema:**

**Microservicio de Usuarios:**

****Riesgo**:** Fallos en la autenticación pueden bloquear el acceso a toda la plataforma.

****Plan de Mitigación**:** Implementar un sistema de autenticación redundante (por ejemplo, autenticación multifactor) y ofrecer métodos de recuperación.

1. **Cliente:**

**Microservicio de Inventario**

****Riesgo**:** Información incorrecta sobre productos, precios o disponibilidad.

****Plan de Mitigación**:** Sincronización continua con el sistema de inventario y validación de la información en tiempo real.

1. **Logística:**

**Microservicio de Pedidos:**

****Riesgo**:** Retrasos o errores en el despacho de productos y rutas incorrectas.

****Plan de Mitigación**:** Integrar servicios de terceros (como APIs de logística) y validar rutas y tiempos de entrega mediante análisis predictivo.

Conclusión

Cómo equipo, se puede considerar que en base a la documentación presentado y las numerosas planeaciones es posible ver este proyecto como algo viable y capaz de ser desarrollado junto con la continua información presentada en clases a medida que progresa el semestre.