Informe de Proyecto: migración de Sistema monolítico a Microservicios

Nombre del Proyecto: MediExpress

Integrantes del equipo: Jarem Cofre, Diego Lara, Carlos Vázquez

Asignatura: Desarrollo de Fullstack

Docente: VICTOR ISIDRO ROSENDO LUGO

Fecha de entrega: 31/03/2025

Definición del Problema:

Contexto de la Empresa

MediExpress Spa es una compañía chilena especializada en la distribución de insumos médicos a hospitales, clínicas y farmacias. Durante el último año, ha experimentado un crecimiento significativo impulsado por la alta demanda del sector.

Problema Central

Este crecimiento exponencial evidenció que la infraestructura tecnológica de MediExpress se ha vuelto obsoleta, lo que limita su capacidad para responder a las exigencias actuales del mercado.

Pruebas Fehacientes del Problema

Los principales desafíos operativos derivados de esta situación son:

Falta de escalabilidad en sistemas actuales.

Deficiencias en la gestión de inventarios y pedidos, con procesos manuales propensos a errores.

Sobrevenda de productos no disponibles por desactualización de stock en tiempo real.

Retrasos recurrentes en entregas por mala planificación logística.

Inconsistencias entre el stock físico y los registros digitales.

Ausencia de integración tecnológica con proveedores, generando descoordinación en la cadena de suministro.

Nuestro objetivo general es brindarle una solución que sea escalable y mantenible en el tiempo, nuestros objetivos del proyecto son los sgt:

Administrador del Sistema

• Gestionar Usuarios: Crear, actualizar, desactivar, activar y eliminar cuentas de usuarios del sistema.

• Configurar Permisos: Asignar y modificar permisos de acceso a diferentes módulos y funciones del sistema.

• Monitorización del Sistema: Visualizar el estado del sistema, recibir alertas sobre posibles fallos y monitorizar el rendimiento.

• Respaldar y Restaurar Datos: Realizar copias de seguridad periódicas y restaurar datos en caso de pérdida o fallo

Gestor de Inventario

• Administrar Productos: Agregar, actualizar y eliminar insumos médicos en el sistema, asegurando que toda la información de disponibilidad esté actualizada.

• Control de Stock: Implementar alertas de inventario bajo para que los proveedores sean notificados de la necesidad de reabastecimiento.

• Gestión de Proveedores: Registrar y actualizar la información de proveedores, permitiendo la automatización de pedidos y la trazabilidad del proceso de adquisición.

• Generación de Reportes de Inventario: Crear reportes detallados sobre la cantidad de productos disponibles, tendencias de consumo y proyecciones de demanda.

Coordinador de Logística

• Registrar Ventas: Procesar transacciones de pedidos en el sistema, aplicando descuentos y ofertas cuando sea necesario.

• Gestión de Entrega: Actualizar el estado de los pedidos en el sistema desde que se recibe la solicitud hasta que se entrega al cliente.

• Consultar Inventario: Verificar disponibilidad de productos en tiempo real y buscar productos en el inventario.

• Generar Facturas: Emitir facturas electrónicas y enviarlas por correo electrónico a los clientes.

Soporte Técnico • Atención de Incidencias: Resolver problemas reportados por usuarios del sistema, tanto empleados como clientes.

• Atender Devoluciones y Reclamaciones: Procesar devoluciones de productos y gestionar reclamaciones de clientes.

• Monitoreo del Funcionamiento del Sistema: Analizar métricas de rendimiento y posibles fallos en la infraestructura para prevenir interrupciones en el servicio.

Acciones de los Clientes vía Web

• Crear Cuenta: Registrarse en la plataforma web proporcionando la

información necesaria de su institución para poder realizar pedidos de

insumos.

• Iniciar Sesión: Acceder a la cuenta utilizando las credenciales de usuario.

• Navegar y Buscar Productos: Explorar el catálogo de productos, utilizando

filtros y la barra de búsqueda para encontrar productos específicos.

• Agregar Productos al Carrito: Seleccionar productos y añadirlos al carrito de

compras.

• Realizar Pedidos: Completar el proceso de compra, proporcionando detalles

de pago y seleccionando opciones de envío.

• Consultar Historial de Pedidos: Ver el historial de compras anteriores y el

estado actual de los pedidos en curso.

• Gestionar Perfil: Actualizar información personal, direcciones de envío y

detalles de pago.

• Solicitar Servicio Técnico: Enviar consultas o problemas a través de un

formulario de contacto o chat en línea.

• Dejar Reseñas y Calificaciones: Evaluar productos comprados y dejar

comentarios en el sitio web.

Estas acciones aseguran que el sistema cubra todas las necesidades operativas de

MediExpress SPA, tanto a nivel interno como en su interacción con los clientes,

facilitando una gestión eficiente y una experiencia de usuario satisfactoria

Estrategia de Microservicios

Elección :Dominio

¿Por qué?

La estrategia de microservicios por dominio se enfoca en dividir el sistema en unidades que corresponden directamente con las áreas del negocio,

facilitando la comprensión tanto para desarrolladores como para stakeholders.

Ventajas:

Escalabilidad y Aislamiento:

Los microservicios por dominio permiten escalar de manera independiente según las necesidades del negocio.

Si un dominio específico enfrenta mayor demanda, solo ese componente necesita ser escalado, sin afectar el resto del sistema.

Esto mejora la eficiencia en la gestión de recursos.

Desarrollo y Despliegue Independiente:

Cada dominio es autónomo, lo que facilita que los equipos trabajen de manera independiente.

Se pueden realizar despliegues y cambios en un dominio sin que estos afecten a otros, lo que acelera la entrega de nuevas características o la corrección de errores.

Tecnologías Específicas por Dominio:

Esta estrategia permite la flexibilidad de elegir tecnologías más adecuadas para cada dominio, optimizando el rendimiento del sistema en su conjunto.

Por ejemplo, se podría elegir un tipo de base de datos específica o un sistema de mensajería adecuado para un dominio particular, lo que no impacta a los otros.

Mejor Mantenimiento a Largo Plazo:

La autonomía de cada dominio hace que el mantenimiento sea más sencillo.

Los cambios realizados en un dominio son aislados, lo que minimiza el riesgo de que los errores se propaguen a otras partes del sistema.

Esto facilita la gestión del sistema a medida que crece.

Herramientas Utilizadas

Lenguaje: java (JavaScript es un lenguaje de programación popular y versátil que tiene muchos beneficios, lo elegimos por su versatilidad, su rapidez y su simplicidad)

Framework: Spring Framework es un framework de código abierto para desarrollar aplicaciones y páginas web en Java. Es uno de los entornos de desarrollo más usados en programación backend.

Base de datos: Java Database Connectivity (JDBC) es una API de Java que permite a las aplicaciones Java conectarse y operar con bases de datos. JDBC es una especificación estándar que define cómo los programas Java pueden acceder a bases de datos

Plataforma Cloud: Google Cloud es una plataforma de computación en la nube que ofrece servicios de almacenamiento, análisis de datos, aprendizaje automático, y más

Funcionalidades

JavaScript: En conjunto, JavaScript sería un componente clave en el desarrollo de la solución tecnológica para MediExpress SPA, tanto en el frontend como en el backend, asegurando que la empresa pueda superar las limitaciones de su sistema actual y soportar su crecimiento futuro.

Spring: La elección de Spring para el desarrollo de la solución tecnológica de MediExpress SPA está plenamente justificada debido a sus características clave como escalabilidad, flexibilidad, seguridad, facilidad de integración y desarrollo rápido.

Java Database Connectivity:

En el contexto del caso de MediExpress SPA, donde el sistema debe manejar grandes cantidades de datos (como pedidos, inventarios, clientes y transacciones), JDBC proporciona un conjunto de funcionalidades esenciales para asegurar la conectividad, gestión eficiente de datos y operaciones sobre bases de datos.

Google cloud:

puede ser utilizado para mejorar la infraestructura y facilitar el crecimiento continuo del sistema, especialmente considerando las necesidades de gestión de datos, escalabilidad, rendimiento y seguridad.

Justificación

JavaScript:

a elección de Java como lenguaje de programación para desarrollar la solución tecnológica para MediExpress SPA está justificada por varias razones clave que responden tanto a las necesidades específicas del proyecto como a las ventajas inherentes que Java ofrece en el contexto de aplicaciones empresariales y sistemas en constante crecimiento

Spring:

La elección de Spring como framework para el desarrollo de la solución tecnológica para MediExpress SPA está justificada por varias razones clave que responden a las necesidades del proyecto, las características específicas del sistema y las ventajas que ofrece Spring en el contexto de aplicaciones empresariales. A continuación, se detallan los motivos por los cuales Spring es una opción adecuada para este caso

Java Database Connectivity:

La elección de Java Database Connectivity (JDBC) para la solución de MediExpress SPA se justifica por una serie de razones que responden a las necesidades específicas de la empresa y las características del sistema que se está desarrollando. JDBC es una API de Java que permite la interacción directa con bases de datos, lo que lo convierte en una herramienta poderosa para garantizar que la plataforma sea eficiente, escalable y fácil de integrar con bases de datos relacionales.

Google cloud:

La elección de Google Cloud para la solución tecnológica de MediExpress SPA se justifica por una serie de razones que responden tanto a las necesidades actuales de la empresa como a sus proyecciones de crecimiento. Google Cloud ofrece una infraestructura robusta, escalable, segura y fácil de gestionar, que facilita la creación de aplicaciones modernas y altamente disponibles.

Administrador del Sistema

1. Gestionar Usuarios: Crear, actualizar, desactivar, activar y eliminar cuentas de usuarios del sistema.

o Herramienta: Spring Security (para la gestión de autenticación y autorización de usuarios) y JDBC (para interactuar con la base de datos y realizar operaciones CRUD sobre usuarios).

2. Configurar Permisos: Asignar y modificar permisos de acceso a diferentes módulos y funciones del sistema.

o Herramienta: Spring Security (para gestionar los roles y permisos de los usuarios en el sistema) y JDBC (para persistir la configuración de permisos en la base de datos).

3. Monitorización del Sistema: Visualizar el estado del sistema, recibir alertas sobre posibles fallos y monitorizar el rendimiento.

o Herramienta: Google Cloud Monitoring (para monitorear el estado y el rendimiento de la infraestructura de la nube) y Google Cloud Alerts (para recibir notificaciones de fallos o problemas).

4. Respaldar y Restaurar Datos: Realizar copias de seguridad periódicas y restaurar datos en caso de pérdida o fallo.

o Herramienta: Google Cloud Storage (para almacenar copias de seguridad) y Google Cloud SQL (para gestionar las bases de datos con opciones de respaldo y restauración automáticas).

Gestor de Inventario

1. Administrar Productos: Agregar, actualizar y eliminar insumos médicos en el sistema, asegurando que toda la información de disponibilidad esté actualizada.

o Herramienta: Spring Boot (para desarrollar la lógica de la aplicación y manejar las solicitudes CRUD de productos) y JDBC (para la interacción con la base de datos y la gestión de los productos).

2. Control de Stock: Implementar alertas de inventario bajo para que los proveedores sean notificados de la necesidad de reabastecimiento.

o Herramienta: Spring Boot (para gestionar las alertas de stock bajo) y Google Cloud Pub/Sub (para enviar notificaciones en tiempo real a los proveedores).

3. Gestión de Proveedores: Registrar y actualizar la información de proveedores, permitiendo la automatización de pedidos y la trazabilidad del proceso de adquisición.

o Herramienta: Spring Boot (para gestionar los registros de proveedores) y JDBC (para interactuar con la base de datos y actualizar la información de proveedores).

4. Generación de Reportes de Inventario: Crear reportes detallados sobre la cantidad de productos disponibles, tendencias de consumo y proyecciones de demanda.

o Herramienta: Spring Boot (para generar los reportes) y Google Cloud BigQuery (para realizar análisis de grandes volúmenes de datos y generar reportes).

Coordinador de Logística

1. Registrar Ventas: Procesar transacciones de pedidos en el sistema, aplicando descuentos y ofertas cuando sea necesario.

o Herramienta: Spring Boot (para gestionar la lógica de ventas) y JDBC (para almacenar las transacciones en la base de datos).

2. Gestión de Entrega: Actualizar el estado de los pedidos en el sistema desde que se recibe la solicitud hasta que se entrega al cliente.

o Herramienta: Spring Boot (para gestionar los estados de los pedidos) y JDBC (para actualizar los registros en la base de datos sobre el estado de los pedidos).

3. Consultar Inventario: Verificar disponibilidad de productos en tiempo real y buscar productos en el inventario.

o Herramienta: Spring Boot (para gestionar las consultas de inventario) y JDBC (para consultar la base de datos en tiempo real sobre la disponibilidad de productos).

4. Generar Facturas: Emitir facturas electrónicas y enviarlas por correo electrónico a los clientes.

o Herramienta: Spring Boot (para generar las facturas) y Google Cloud Storage (para almacenar las facturas generadas). Además, se puede usar Google Cloud SendGrid o SMTP para enviar las facturas por correo electrónico.

Soporte Técnico

1. Atención de Incidencias: Resolver problemas reportados por usuarios del sistema, tanto empleados como clientes.

o Herramienta: Spring Boot (para gestionar las incidencias y notificaciones de los usuarios) y JDBC (para registrar las incidencias en la base de datos).

2. Atender Devoluciones y Reclamaciones: Procesar devoluciones de productos y gestionar reclamaciones de clientes.

o Herramienta: Spring Boot (para gestionar devoluciones y reclamaciones) y JDBC (para almacenar información sobre devoluciones y reclamaciones).

3. Monitoreo del Funcionamiento del Sistema: Analizar métricas de rendimiento y posibles fallos en la infraestructura para prevenir interrupciones en el servicio.

o Herramienta: Google Cloud Monitoring (para monitorizar métricas de rendimiento y recibir alertas sobre fallos en la infraestructura).

Acciones de los Clientes vía Web

1. Crear Cuenta: Registrarse en la plataforma web proporcionando la información necesaria de su institución para poder realizar pedidos de insumos.

o Herramienta: Spring Boot (para la gestión de registros de usuarios) y Spring Security (para la autenticación y autorización de los usuarios).

2. Iniciar Sesión: Acceder a la cuenta utilizando las credenciales de usuario.

o Herramienta: Spring Security (para gestionar la autenticación y autorización de usuarios) y Google Cloud Identity Platform (para servicios de autenticación adicionales).

3. Navegar y Buscar Productos: Explorar el catálogo de productos, utilizando filtros y la barra de búsqueda para encontrar productos específicos.

o Herramienta: Spring Boot (para gestionar la lógica de búsqueda) y JDBC (para realizar consultas a la base de datos sobre productos).

4. Agregar Productos al Carrito: Seleccionar productos y añadirlos al carrito de compras.

o Herramienta: Spring Boot (para gestionar el carrito de compras de los usuarios) y Google Cloud Firestore (para almacenar de manera eficiente el estado del carrito de compras).

5. Realizar Pedidos: Completar el proceso de compra, proporcionando detalles de pago y seleccionando opciones de envío.

o Herramienta: Spring Boot (para gestionar las transacciones de pedidos) y Google Cloud Payments o Google Cloud Pay (para procesar pagos).

6. Consultar Historial de Pedidos: Ver el historial de compras anteriores y el estado actual de los pedidos en curso.

o Herramienta: Spring Boot (para gestionar el historial de pedidos) y JDBC (para almacenar y consultar el historial de pedidos).

7. Gestionar Perfil: Actualizar información personal, direcciones de envío y detalles de pago.

o Herramienta: Spring Boot (para gestionar la actualización del perfil) y Spring Security (para asegurar que la información se gestione de manera segura).

8. Solicitar Servicio Técnico: Enviar consultas o problemas a través de un formulario de contacto o chat en línea.

o Herramienta: Spring Boot (para gestionar las solicitudes de soporte técnico) y Google Cloud Chat API o Google Cloud Contact Center AI (para gestionar las interacciones con los usuarios).

9. Dejar Reseñas y Calificaciones: Evaluar productos comprados y dejar comentarios en el sitio web.

o Herramienta: Spring Boot (para gestionar las reseñas y calificaciones de productos) y JDBC (para almacenar las reseñas en la base de datos).

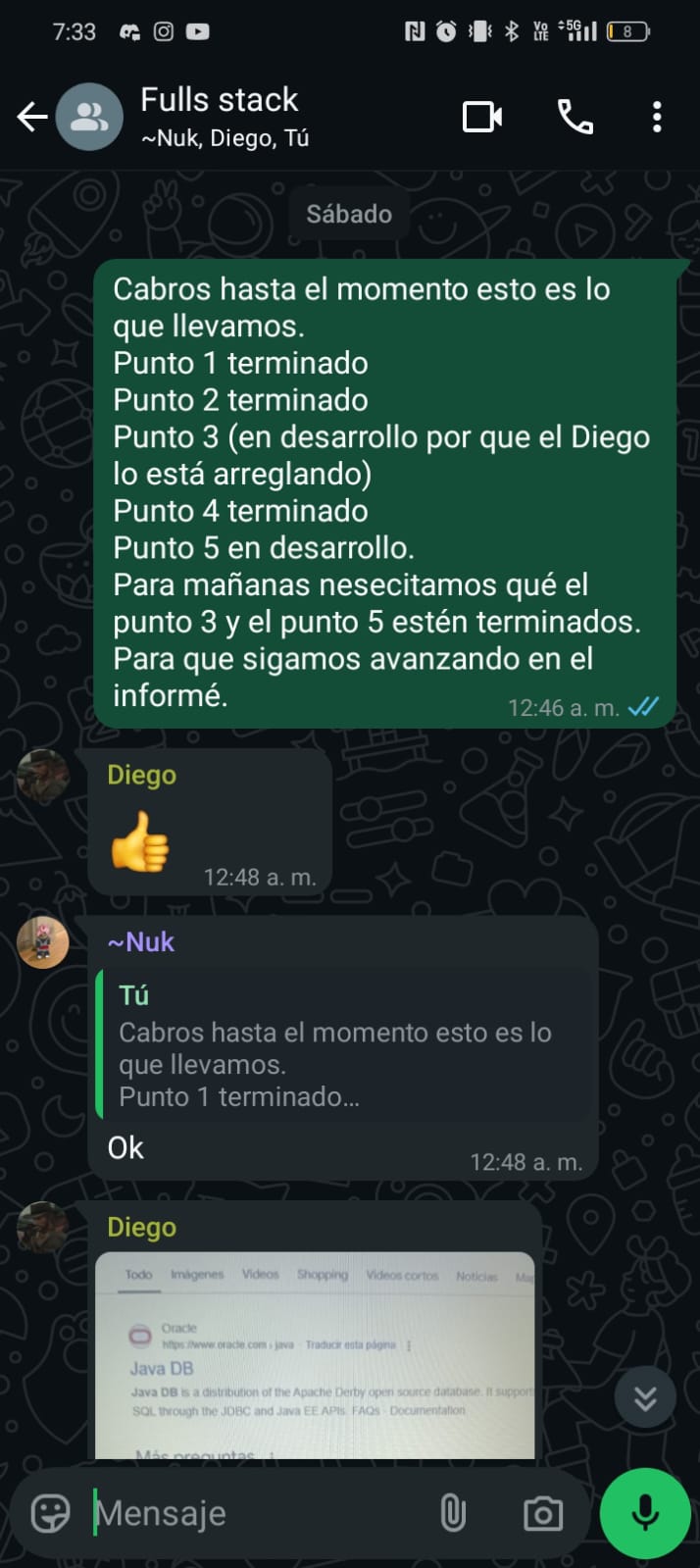
Herramientas de Trabajo Colaborativo

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

<https://trello.com/invite/b/67e767c8cf3e703e17140ae0/ATTI99030cbaea0bee1157b2d1e59f95b608B460D327/caso-de-estudiomediexpress-spa>

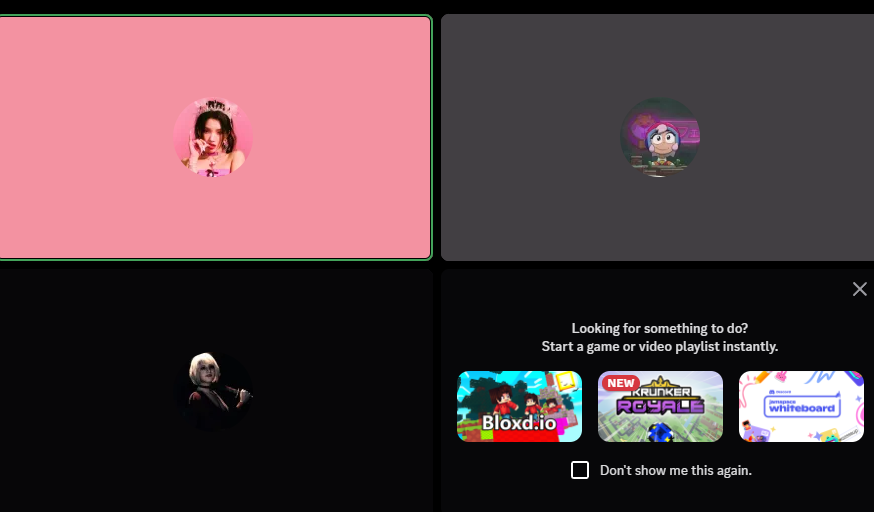
Capturas de organización por puntos del equipo:



Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Llamada de equipo:



Enfoque Ético en el Desarrollo:

Privacidad de los Datos:

Desafío: MediExpress SPA manejará datos sensibles, incluyendo información personal de pacientes y detalles médicos,

lo que plantea un riesgo significativo si se filtran o manejan de manera inapropiada.

Solución: Es crucial implementar medidas robustas de protección de datos, como el cifrado de datos tanto en reposo como en tránsito.

Además, se debe garantizar que los accesos a los datos sean restringidos y auditados.

También se pueden adoptar tecnologías de amonificación y pseudonimización para minimizar los riesgos.

2. Seguridad:

Desafío: La empresa debe proteger sus sistemas contra ciberataques, vulnerabilidades o fugas de información,

ya que cualquier brecha podría comprometer tanto la operación como la confidencialidad de los datos.

Solución: Es recomendable implementar una estrategia de ciberseguridad integral, que incluya firewalls avanzados, autenticación de múltiples factores,

y pruebas de penetración periódicas. Además, la empresa debería proporcionar formación continua sobre buenas prácticas de seguridad a todos los empleados.

3. Responsabilidad en el Despliegue:

Desafío: El cambio hacia un sistema más automatizado podría generar impactos imprevistos, tanto en los usuarios finales como en la empresa,

que podrían no estar preparados para la transición.

Solución: La implementación debe realizarse de manera gradual y controlada.

Además, es importante contar con un plan de contingencia en caso de fallos del sistema,

y asegurar que los usuarios reciban formación continua sobre el uso del nuevo sistema.

4. Impacto en los Puestos de Trabajo:

Desafío: La automatización podría generar la preocupación de que algunos trabajos sean reemplazados por sistemas, afectando la moral de los empleados.

Solución: Se debe trabajar en la capacitación y actualización de los empleados para que puedan adaptarse a la nueva tecnología,

promoviendo una cultura de trabajo colaborativo entre humanos y máquinas.

La automatización debe estar orientada a tareas repetitivas, dejando a los empleados tareas de mayor valor añadido.

Soluciones y Buenas Prácticas Adoptadas

Privacidad y Seguridad: Para cumplir con las regulaciones locales como la Ley de Protección de Datos Personales (Ley 19.496 en Chile),

es fundamental aplicar políticas claras de privacidad, usar tecnologías de protección de datos y garantizar que los usuarios tengan acceso a la gestión de sus datos.

Cumplimiento Normativo: Se debe garantizar el cumplimiento de la Ley de Protección de Datos Personales de Chile, que regula el tratamiento de datos personales.

La empresa debe realizar auditorías periódicas para garantizar que se cumplen las obligaciones legales en materia de protección de datos.

Al implementar estas prácticas éticas, MediExpress SPA podrá abordar los riesgos y mejorar la confianza de sus usuarios y empleados en el sistema.

=======================================================================================================================================================================

Ley 19.496: Protección de los Derechos de los Consumidores

Esta ley establece una serie de derechos para los consumidores, incluyendo la protección de la información personal de los clientes.

Los principios clave de esta ley incluyen:

Información clara y precisa sobre el uso de los datos personales.

Consentimiento explícito por parte de los consumidores para el tratamiento de sus datos.

Derecho a la eliminación o rectificación de la información personal.

En el caso de MediExpress SPA, se debe asegurar que:

La empresa tenga políticas claras sobre cómo recoge, usa y comparte los datos personales de los consumidores.

Los consumidores sean informados explícitamente de cómo se usarán sus datos antes de ser recopilados.

Se permita a los consumidores ejercer sus derechos de acceso, rectificación, y eliminación de sus datos personales.

2. Ley 19.628: Protección de la Vida Privada

Esta ley regula la recolección, almacenamiento, y uso de los datos personales en Chile. Específicamente, la ley establece que:

El tratamiento de datos personales debe ser realizado con el consentimiento informado del titular.

Los datos deben ser adecuados, pertinentes, y no excesivos en relación con la finalidad para la cual se recogen.

Las personas tienen derecho a conocer y corregir sus datos personales.

Seguridad en el manejo de la información personal, para evitar accesos no autorizados.

Para cumplir con esta ley, MediExpress SPA debe:

Garantizar que todos los datos personales se recojan de manera legal y transparente, con el consentimiento adecuado de los usuarios.

Implementar medidas de seguridad como cifrado de datos, autenticación y control de accesos para proteger los datos personales.

Establecer un proceso claro para que los usuarios puedan ejercer sus derechos de acceso y rectificación de sus datos.

3. Cumplimiento de Normas de Ciberseguridad:

Dado que la ley exige la protección de la información personal,

MediExpress SPA debe garantizar que su infraestructura tecnológica cuente con medidas adecuadas de ciberseguridad para proteger los datos contra accesos no autorizados. Esto incluye:

Uso de firewalls, cifrado de datos, y otras herramientas de protección para salvaguardar la integridad y privacidad de los datos.

Capacitación continua de los empleados sobre buenas prácticas de seguridad informática.

4. Responsabilidad en el Despliegue:

A la hora de implementar la nueva solución tecnológica,

MediExpress SPA debe asegurarse de que el proceso sea transparente y se alinee con las normativas chilenas en cuanto al manejo de datos personales,

especialmente en cuanto a la recolección y uso de datos sensibles relacionados con la salud.

Conclusión:

El enfoque propuesto para la privacidad y seguridad de los datos, así como para el cumplimiento normativo de las leyes chilenas,

está alineado con las regulaciones locales, como la Ley 19.496 y la Ley 19.628. Implementando las prácticas descritas,

MediExpress SPA puede cumplir adecuadamente con estas leyes, asegurando la privacidad y la seguridad de los datos de los consumidores.

Análisis de Requerimientos para MediExpress SPA

A continuación, realizamos un análisis detallado de los requisitos funcionales y no funcionales para el sistema propuesto, así como la síntesis de necesidades para cada perfil de usuario descrito en el caso.

1. Requisitos Funcionales

Los **requisitos funcionales** son aquellos que definen las funciones y características específicas que el sistema debe cumplir para satisfacer las necesidades del negocio. En este caso, los requisitos funcionales abarcan las siguientes áreas clave:

Administrador del Sistema:

1. Gestionar **Usuarios**:
   * Crear, actualizar, desactivar, activar y eliminar cuentas de usuarios.
   * Asignar roles y permisos a cada usuario.
2. ConfigurarPermisos:
   * Asignar permisos de acceso a diferentes módulos (Gestión de Inventario, Logística, etc.).
   * Modificar permisos de acceso a funcionalidades específicas según los roles de usuario.
3. MonitorizacióndelSistema:
   * Visualizar el estado y salud del sistema.
   * Recibir alertas sobre posibles fallos o problemas en el sistema.
   * Monitorizar métricas de rendimiento.
4. RespaldaryRestaurarDatos:
   * Realizar copias de seguridad periódicas.
   * Restaurar datos en caso de pérdida o fallo.

Gestorde Inventario**:**

1. Administrar Productos:
   * Agregar, actualizar y eliminar productos del inventario.
   * Gestionar descripciones, precios, y disponibilidad.
2. Control de Stock:
   * Implementar alertas automáticas cuando el stock de un producto esté bajo.
   * Notificar a los proveedores para reabastecer productos automáticamente.
3. Gestión de Proveedores:
   * Registrar y actualizar información de proveedores.
   * Generar pedidos automáticos a los proveedores cuando el inventario sea bajo.
4. Generación de Reportes de Inventario:
   * Generar reportes detallados de inventario.
   * Crear reportes sobre tendencias de consumo y proyecciones de demanda futura.

Coordinador de Logística:

1. Registrar Ventas:
   * Procesar transacciones de pedidos (con descuentos y ofertas si es necesario).
2. Gestión de Entrega:
   * Actualizar el estado de los pedidos a medida que avanzan en el proceso de entrega.
   * Optimizar las rutas de entrega.
3. Consultar Inventario:
   * Verificar la disponibilidad de productos en tiempo real.
   * Consultar y buscar productos en el inventario.
4. Generar Facturas:
   * Emitir facturas electrónicas y enviarlas a los clientes por correo electrónico.

Soporte Técnico:

1. Atención de Incidencias:
   * Gestionar y resolver problemas reportados por usuarios (empleados y clientes).
2. Atender Devoluciones y Reclamaciones:
   * Gestionar devoluciones de productos y reclamaciones de los clientes.
3. Monitoreo del Funcionamiento del Sistema:
   * Monitorear métricas del sistema para detectar posibles fallos en la infraestructura y prevenir interrupciones en el servicio.

Clientes (vía Web):

1. Crear Cuenta:
   * Registrar cuenta en la plataforma proporcionando la información de la institución.
2. Iniciar Sesión:
   * Acceder a la cuenta utilizando credenciales de usuario.
3. Navegar y Buscar Productos:
   * Explorar el catálogo de productos usando filtros y barra de búsqueda.
4. Agregar Productos al Carrito:
   * Seleccionar productos y añadirlos al carrito de compras.
5. Realizar Pedidos:
   * Completar la compra con detalles de pago y opciones de envío.
6. Consultar Historial de Pedidos:
   * Ver historial de compras previas y el estado de pedidos en curso.
7. Gestionar Perfil:
   * Actualizar información personal, direcciones de envío y detalles de pago.
8. Solicitar Servicio Técnico:
   * Enviar consultas o problemas a través de un formulario de contacto o chat.
9. Dejar Reseñas y Calificaciones:
   * Evaluar productos y dejar comentarios en el sitio web.

2. Requisitos No Funcionales

Los **requisitos no funcionales** se refieren a las características del sistema que no están directamente relacionadas con las funcionalidades específicas, pero son esenciales para el desempeño y la calidad del sistema. Estos requisitos son cruciales para garantizar que el sistema sea eficiente, seguro y confiable.

1. Rendimiento:
   * El sistema debe ser capaz de manejar grandes volúmenes de transacciones (pedidos, consultas de inventario) sin afectar su rendimiento.
   * Las respuestas a las consultas de productos y pedidos deben ser rápidas, incluso cuando el volumen de usuarios y transacciones crezca.
2. Escalabilidad:
   * El sistema debe ser capaz de escalar horizontalmente para manejar un número creciente de usuarios, pedidos y datos a medida que la empresa crece.
   * Utilizar tecnologías basadas en microservicios, como **Spring Boot** y **Google Cloud**, para asegurar que los componentes individuales puedan ser escalados de manera independiente.
3. Seguridad:
   * El sistema debe garantizar la protección de los datos sensibles, como la información de las cuentas de los clientes y los detalles de pago, utilizando medidas de seguridad como cifrado de datos y autenticación robusta.
   * El control de acceso debe estar bien definido y gestionado, especialmente para las funciones de administración y la gestión de inventarios.
4. Disponibilidad:
   * El sistema debe ser altamente disponible, con tiempos de inactividad mínimos para evitar pérdidas de ventas y deterioro en la experiencia del usuario.
   * La infraestructura en **Google Cloud** debe estar configurada para asegurar la alta disponibilidad y la recuperación ante desastres.
5. Usabilidad:
   * El sistema debe ser fácil de usar para todos los perfiles, desde los administradores hasta los clientes.
   * La interfaz debe ser intuitiva y amigable para facilitar la adopción del sistema por parte de los usuarios.
6. Mantenimiento y Actualización:
   * El sistema debe ser fácil de mantener y actualizar sin afectar la operación del negocio. Se deben poder hacer actualizaciones regulares sin tiempos de inactividad largos.
7. Compatibilidad:
   * El sistema debe ser compatible con los principales navegadores web (Chrome, Firefox, Safari, etc.) y dispositivos (PC, tabletas, smartphones).

3. Síntesis de Necesidades de Cada Perfil

1. Administrador del Sistema:

* Necesidades:
  + Control total sobre el sistema, usuarios y permisos.
  + Herramientas para la monitorización del sistema y gestión de incidencias.
  + Funcionalidades de respaldo y restauración para proteger la integridad de los datos.

2. Gestor de Inventario:

* Necesidades:
  + Acceso a funcionalidades para gestionar productos y proveedores.
  + Herramientas para generar reportes detallados y notificaciones de bajo stock.
  + Capacidad para integrar pedidos y reabastecer el inventario de forma eficiente.

3. Coordinador de Logística:

* Necesidades:
  + Herramientas para gestionar el flujo de ventas y pedidos.
  + Capacidades para gestionar la entrega de productos y optimizar las rutas.
  + Funciones de facturación para emitir y gestionar facturas electrónicas.

**4.** Soporte Técnico:

* Necesidades:
  + Herramientas para gestionar incidencias y problemas reportados.
  + Funcionalidad para gestionar devoluciones y reclamaciones de clientes.
  + Capacidades de monitorización para anticiparse a posibles fallos del sistema.

**5.** Clientes (vía Web):

* Necesidades:
  + Interfaz de usuario sencilla para navegar y comprar productos.
  + Funcionalidad para gestionar su cuenta, historial de pedidos y perfil.
  + Herramientas para contactar con soporte y dejar reseñas de productos.

Análisis del Sistema Actual de MediExpress SPA

En este apartado, se realiza un análisis del sistema monolítico actual utilizado por MediExpress SPA para gestionar sus operaciones. El objetivo es identificar sus características, su arquitectura, así como los puntos débiles que afectan el desempeño, la escalabilidad, y la facilidad de mantenimiento del sistema.

1. Arquitectura General del Sistema Monolítico

El sistema actual de MediExpress SPA se basa en una arquitectura monolítica, lo que significa que todas las funcionalidades de la empresa (gestión de inventarios, pedidos, clientes, administración, etc.) se agrupan en un único bloque de código que interactúa con las distintas partes del sistema. En términos generales, esta arquitectura se caracteriza por:

1. Componentes Fuertemente Acoplados:

o Todos los módulos del sistema están interrelacionados y dependen del mismo bloque de código, lo que dificulta realizar cambios o actualizaciones en un área específica sin afectar a otras.

2. Base de Datos Centralizada:

o El sistema utiliza una base de datos única para almacenar la información de usuarios, inventarios, pedidos, transacciones, y más. Esta base de datos se accede de manera centralizada y es compartida por todas las funcionalidades.

3. Desarrollo y Despliegue Únicos:

o La aplicación monolítica se desarrolla y despliega como una única unidad. Cualquier cambio o nueva característica requiere una nueva versión completa del sistema, lo que genera tiempos de inactividad y posibles errores.

4. Escalabilidad Limitada:

o Debido a que el sistema está integrado en un solo bloque de código, escalarlo es un desafío. Si el sistema necesita más recursos para manejar más usuarios o transacciones, se debe escalar toda la aplicación, lo que puede ser costoso y complicado.

5. Interfaz de Usuario Común:

o La interfaz de usuario (UI) no está diseñada específicamente para cada tipo de usuario (administrador, gestor de inventario, cliente, etc.), lo que dificulta la personalización de la experiencia según el perfil.

6. Desempeño y Respuesta:

o Cuando el número de usuarios o de operaciones aumenta, el rendimiento de la aplicación se ve afectado debido a la sobrecarga de recursos en el servidor. Este problema es especialmente notorio cuando el sistema realiza operaciones de lectura y escritura intensivas en la base de datos.

2. Puntos Débiles Detectados en el Sistema Actual

Los siguientes puntos débiles han sido identificados en el sistema monolítico actual de MediExpress SPA:

a. Cuellos de Botella y Desempeño

· Acceso Centralizado a la Base de Datos:

o Debido a la estructura monolítica, todas las solicitudes de acceso a datos deben pasar por un único punto de acceso (la base de datos centralizada). Esto genera cuellos de botella, ya que el sistema no puede distribuir las cargas de trabajo de manera eficiente entre varios recursos. En periodos de alta demanda, el sistema puede volverse lento o incluso caer debido a la sobrecarga.

· Falta de Escalabilidad:

o El sistema no está diseñado para escalar fácilmente. Si la demanda aumenta, por ejemplo, al gestionar más pedidos o usuarios, se requiere escalar toda la aplicación en lugar de poder escalar de manera modular los componentes individuales. Esto implica tiempos de inactividad prolongados y costes más altos.

· Falta de Optimización de Rutas de Entrega:

o No existe una integración eficiente que permita optimizar las rutas de entrega ni una correcta gestión de los tiempos de entrega. Esto se traduce en retrasos en los pedidos, lo que afecta la satisfacción del cliente y la eficiencia de las operaciones.

b. Mantenimiento y Actualización Dificultosos

· Acoplamiento Estrecho entre Componentes:

o Dado que todos los módulos están interconectados en un solo bloque, cualquier cambio en una funcionalidad afecta a otras áreas del sistema. Esto hace que el proceso de mantenimiento sea complejo y arriesgado. Las actualizaciones pueden generar errores imprevistos o afectar otras funcionalidades que dependen de ese componente específico.

· Dependencia de una Base de Datos Única:

o Tener una base de datos centralizada para todo el sistema implica que si algo sale mal, como una falla en el servidor de la base de datos, puede afectar todo el sistema. Además, la administración de la base de datos se vuelve más difícil con el tiempo, ya que el volumen de datos y la complejidad de las transacciones crecen.

· Despliegues Lentos y Costosos:

o La aplicación monolítica requiere que toda la plataforma se despliegue a la vez cuando se realizan cambios, lo que genera tiempos de inactividad prolongados. Esto puede ser problemático en un entorno de producción, ya que cualquier error puede causar interrupciones en las operaciones comerciales.

c. Falta de Flexibilidad y Personalización

· Manejo de Roles y Permisos Limitado:

o El sistema actual no permite un control granular de los permisos. La asignación de permisos a usuarios de diferentes roles (administrador, coordinador de logística, gestor de inventario, etc.) es más difícil de manejar. Cualquier modificación en los permisos requiere cambios a nivel de la base de datos y el código del sistema, lo que incrementa el riesgo de error.

· Experiencia de Usuario No Personalizada:

o Dado que la interfaz de usuario está integrada para todos los perfiles, no se pueden ofrecer experiencias personalizadas para los distintos roles dentro del sistema. Esto puede resultar en una experiencia de usuario menos eficiente, ya que los administradores, técnicos de soporte y clientes deben interactuar con las mismas interfaces, sin adaptaciones específicas a sus necesidades.

d. Integración con Proveedores y Proceso de Actualización del Inventario Deficiente

· Desactualización del Inventario:

o El sistema no está integrado con los proveedores, lo que dificulta la actualización automática de los niveles de inventario. Esto genera desincronización entre el inventario real y el que el sistema muestra a los usuarios, lo que provoca situaciones como la sobreventa de productos no disponibles.

· Falta de Automatización en los Pedidos:

o La falta de integración con proveedores también implica que el proceso de reabastecimiento del inventario se realiza de forma manual, lo que incrementa el riesgo de errores y reduce la eficiencia. Las alertas de stock bajo no están totalmente automatizadas, lo que puede generar demoras en el proceso de adquisición.

3. Resumen de los Puntos Débiles del Sistema Monolítico

1. Desempeño Deficiente:

o El sistema presenta cuellos de botella por el acceso centralizado a la base de datos y la falta de escalabilidad.

2. Mantenimiento Complejo:

o Los cambios en un área afectan otras partes del sistema, dificultando las actualizaciones y el mantenimiento. Los despliegues son lentos y costosos.

3. Falta de Flexibilidad:

o El sistema no permite personalizar la experiencia para diferentes tipos de usuarios ni manejar los permisos de forma eficiente.

4. Problemas en la Gestión del Inventario:

o No hay integración con los proveedores, lo que genera desactualización del inventario y procesos manuales ineficientes.

5. Falta de Optimización:

o No hay optimización en el proceso de gestión de pedidos y rutas de entrega, lo que afecta la eficiencia y la satisfacción del cliente.

Diseño de la nueva Arquitectura

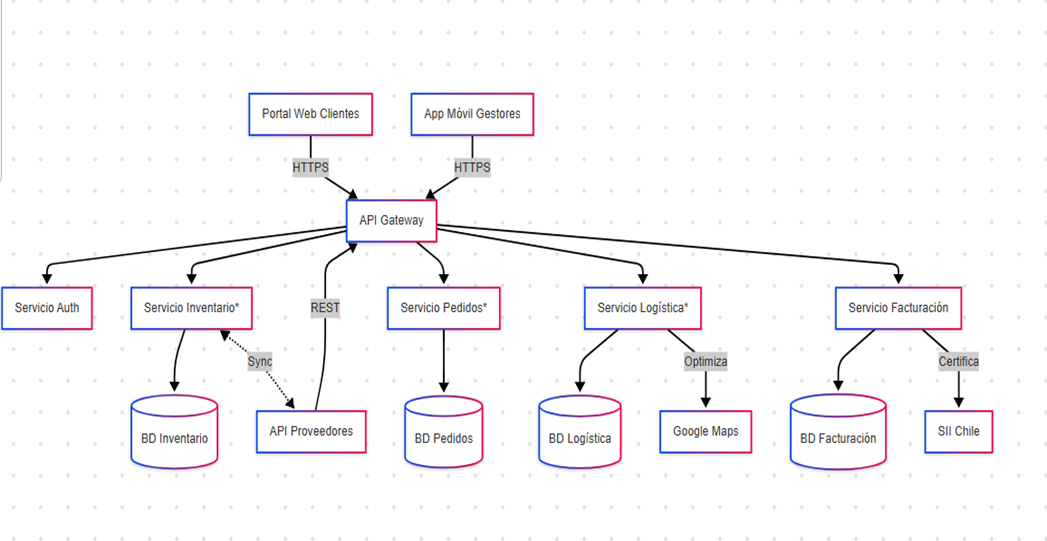
Fases de migración

Diagrama de Actores

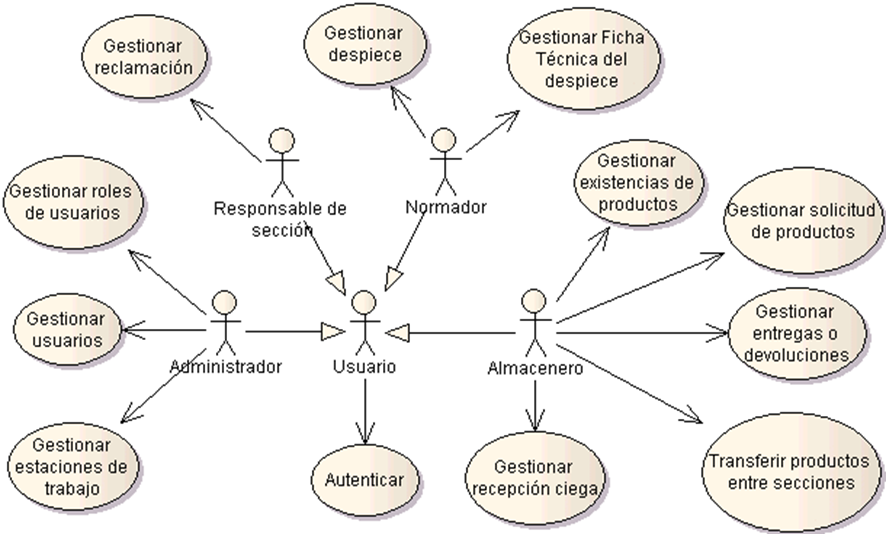


Diagrama de Casos de Uso

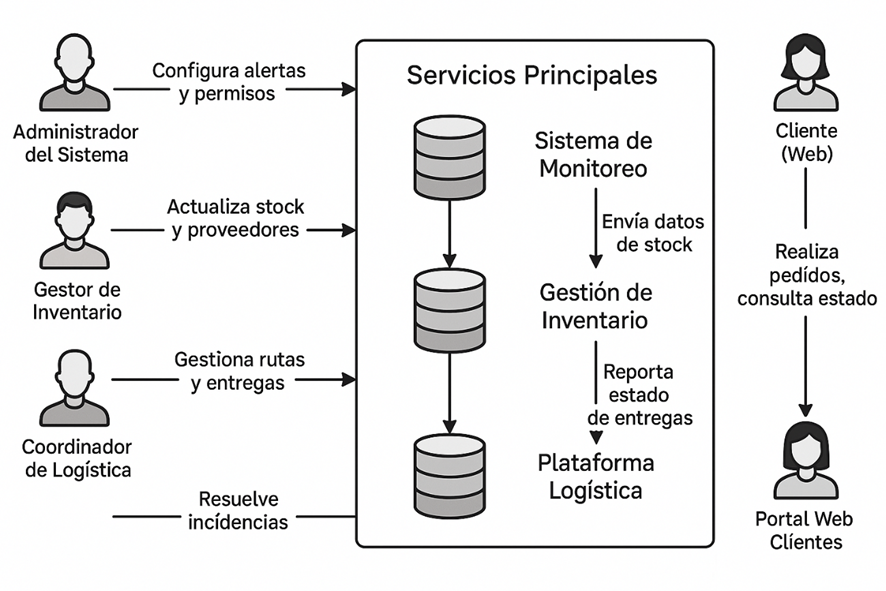
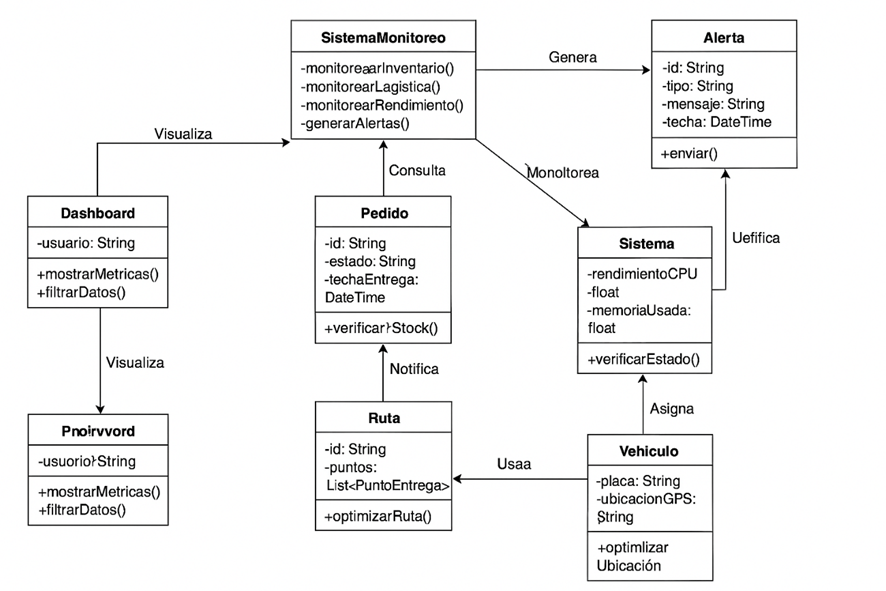


Diagrama de Clases



Planificación Ágil y Cronograma

Metodología Scrum

Adoptamos Scrum como marco ágil para este proyecto, permitiendo:

Avance iterativo e incremental mediante Sprint de 2 semanas.

Flexibilidad para reajustar prioridades según retroalimentación.

Distribución eficiente de tareas entre integrantes con roles definidos.

Roles del Equipo

Rol Responsable Funciones Clave

Product Owner Jarem Jeremy Cofré Díaz - Gestiona el backlog del proyecto.

- Prioriza tareas y define criterios de aceptación.

- Supervisa el cumplimiento de plazos.

Scrum Master Diego Lara - Organiza ceremonias (daily meetings, sprint review).

- Elimina impedimentos del equipo.

- Garantiza adherencia a Scrum.

Scrum Team Carlos Vázquez - Desarrollo técnico de funcionalidades.

- Implementación de microservicios.

- Pruebas unitarias y documentación

Carta Gantt

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Etapas** | **Marzo** | | | | **Abril** | | | | **Mayo** | | | | | **Junio** | | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | | **2** | **3** | **4** |
| Etapa 1 - Creación de Usuarios |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| Etapa 2 – Gestor de inventario |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| Etapa 3 – Gestor de inventario |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| Etapa 4 – Soporte técnico |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| Etapa 5 – Acciones clientes via web |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| Entrega proyecto |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |

Conclusiones y Recomendaciones

• Lecciones aprendidas

En este proyecto hemos podido aprender cómo trabajar en equipo y desarrollar de forma eficiente y flexible un informe completo y detallado sobre la empresa MediExpress SPA. Logramos avanzar mucho y formar una sólida base para seguir trabajando con esta empresa. Somos fieles creyentes de que MediExpress SPA tiene mucho más que ofrecer y mucho más que mejorar.

• Potencial escalabilidad del sistema

El nuevo sistema propuesto para MediExpress SPA, basado en microservicios, Google Cloud y bases de datos distribuidas, tiene un alto potencial de escalabilidad. A medida que el volumen de operaciones crezca, el sistema podrá adaptarse de forma flexible y eficiente sin los límites que presenta la arquitectura monolítica actual.

La capacidad de escalar de manera horizontal, realizar actualizaciones independientes y gestionar los recursos de manera eficiente ayudará a MediExpress SPA a soportar su continuo crecimiento y a optimizar la distribución de insumos médicos en el futuro.

Posibles Mejoras Futuras

Integración de Inteligencia Artificial (IA) y Machine Learning

Predicción de demanda avanzada: Implementar modelos predictivos para anticipar necesidades de insumos médicos por región o temporada, optimizando el stock y reduciendo costos.

Chatbots inteligentes: Usar IA para automatizar el 80% de las consultas de soporte técnico, mejorando tiempos de respuesta.

Blockchain para trazabilidad de productos

Registrar cada movimiento de insumos médicos (desde proveedores hasta clientes) en una red descentralizada, garantizando transparencia y cumplimiento normativo.

Aplicación móvil nativa

Desarrollar una app para que clientes realicen pedidos, consulten entregas en tiempo real (con GPS) y reciban notificaciones personalizadas.

IoT en gestión de almacenes

Instalar sensores para monitorear condiciones críticas (temperatura, humedad) en tiempo real, asegurando la calidad de insumos médicos sensibles.

Autenticación biométrica

Reforzar seguridad con reconocimiento facial o huella digital para accesos administrativos y transacciones críticas.

Plataforma B2B para proveedores

Permitir que proveedores gestionen pedidos, stock y facturas directamente en el sistema, reduciendo intermediarios y errores manuales.

Business Intelligence en tiempo real

Integrar herramientas como Google Data Studio para dashboards interactivos que apoyen decisiones estratégicas basadas en métricas actualizadas.

Expansión regional con CDN

Usar redes de entrega de contenido (CDN) para reducir latencia en regiones remotas, mejorando la experiencia de usuarios fuera de centros urbanos.

Beneficios clave:

Mayor eficiencia: Automatización de procesos críticos.

Escalabilidad: Adaptación a mercados nuevos sin sobrecargar el sistema.

Competitividad: Alineación con tendencias tecnológicas globales en el sector salud.

Anexos

[L1B3SB3BTH/trabajo: Trabajo informe fullsstack](https://github.com/L1B3SB3BTH/trabajo)

[punto 11 en progreso en Caso de Estudio:MediExpress SPA | Trello](https://trello.com/c/lhwYXuHI/21-punto-11-en-progreso)

<https://www.planttext.com/>

<https://gemini.google.com/app/37a69e21fa47f935?hl=es>

<https://www.mermaidchart.com/app/projects/87fe8172-16bd-435d-973d-f6fb04732aad/diagrams/05ea37da-af04-4063-822f-0dd65fdbedb1/version/v0.1/edit> <https://chatgpt.com/c/67eaca96-ad38-800f-a4a1-7fdc8a9364da> Herramientas utilizadas