

Acta Constitución Proyecto

Análisis de Requerimientos de alto nivel y Propuesta de entrada

Sección:001D

Docente: Viviana Poblete

Integrantes: Javier Albornoz

Vicente Hormazabal

Diego Neira

Índice de Contenidos

| | |
|---|----|
| Información del proyecto | 2 |
| Datos | 2 |
| Patrocinadores | 2 |
| Gerente de Proyecto | 2 |
| Lista de Interesados (stakeholders) | 2 |
| Descripción del proyecto | 3 |
| Objetivos de Negocio | 3 |
| Justificación del proyecto – Contexto | 3 |
| Problema-Necesidad | 4 |
| Requerimientos de alto nivel del Proyecto | 5 |
| Descripción del Proyecto | 6 |
| Análisis Propuesta inicial (Diagnóstico de Alcances) | 6 |
| Redefinición de propuesta de entrada en base al Proyecto | 6 |
| Objetivo del proyecto | 7 |
| Alcances del proyecto | 7 |
| Objetivos del desarrollo | 8 |
| Descripción de la solución | 8 |
| Descripción del sistema ideado en base a los requerimientos y visión del Proyecto | 8 |
| Alcances del Producto - Premisas y restricciones | 9 |
| Especificaciones técnicas de las herramientas de desarrollo | 10 |
| Tipo de Infraestructura de Hardware y Sistemas de implementación y/o servicios a utilizar | 11 |
| Organización del equipo Proyecto | 12 |
| Hitos principales del Proyecto | 12 |
| Requisitos de aprobación de propuesta de entrada del proyecto | 13 |
| Aprobaciones y control de cambios | 13 |

Información del proyecto

Datos

| | | |
|--|------------------------|-----------------------|
| | Empresa / Organización | Perfulandia |
| | Nombre del Proyecto | OMNIA |
| | Fecha de inicio/fin | 27/03/2025-31/07/2025 |
| | Cliente | Perfulandia |
| | Patrocinador principal | line hit SPA |
| | Jefe de Proyecto | Javier Albornoz |

Patrocinadores

| Nombre | Cargo | Departamento / División |
|--------------|-------|--------------------------|
| Miguel Tapia | Jefe | departamento de finanzas |

Gerente de Proyecto

| Nombre | Cargo | Departamento / División |
|-----------|-------|---|
| Jair Pino | CEO | departamento de logística y operaciones |

Lista de Interesados (stakeholders)

| Nombre | Tipo | Cargo | Departamento / División |
|------------------|---------------|------------------------|---------------------------------|
| Fabian Zamorano | administrador | Usuario | Accionista mayor de Perfulandia |
| Sebastian Loyola | CEO | patrocinador | Gerente de ventas |
| Miguel Duran | bodeguista | Gerente de la sucursal | administración |
| Michael Smith | trabajador | jefe de logistica | logistica |

Descripción del proyecto

Objetivos de Negocio

los principales objetivos del negocio son :

- Modernizar la infraestructura tecnológica
- Mejorar la experiencia del cliente
- Facilitar la expansión nacional

Justificación del proyecto – Contexto

Nuestro cliente “Perfulandia” ha presentado un alto crecimiento en los últimos tiempos, pero su sistema presenta fallos visibles los cuales obstruyen la buena experiencia de los clientes , esto para la marca se puede reflejar en pérdidas económicas y que su reputación baje.

Bajo este contexto nosotros como empresa venimos a darle solución a sus problemas en el sistema.

Problema-Necesidad

El sistema actual de nuestro cliente “Perfulandia” tiene problemas de conectividad en momentos de alta demanda, por lo cual se debe de migrar a una arquitectura de microservicios para poder obtener una mayor escalabilidad y mejorar la experiencia del cliente.

Requerimientos de alto nivel del Proyecto

| | |
|----|---------------------------------|
| 1 | GESTIONAR USUARIOS |
| 2 | MONITOREAR SISTEMA |
| 3 | ADMINISTRACIÓN DE SISTEMA |
| 4 | GERENTE DE LA SUCURSAL |
| 5 | EMPLEADO EN VENTAS |
| 6 | LOGÍSTICA |
| 7 | CLIENTE |
| 8 | SEGURIDAD Y PROTECCIÓN DE DATOS |
| 9 | INTERFAZ INTUITIVA Y RESPONSIVE |
| 10 | DISPONIBILIDAD 24/7 |
| 11 | ESCALABILIDAD |

Descripción del Proyecto

Análisis Propuesta inicial (Diagnóstico de Alcances)

Nuestro proyecto llamado “Omnia” trata acerca de la migración del sistema monolítico a un sistema de microservicios con el fin de resolver fallas/problemas como lo son el problema de conectividad y el alto crecimiento de la empresa.

Nuestro cliente es la empresa Perfulandia la cual es una empresa chilena emergente que se ha destacado por ofrecer productos de alta calidad a precios competitivos.

Objetivo del proyecto

Nuestro objetivo es desarrollar e implementar una arquitectura de microservicios para poder reemplazar el sistema actual con el fin de garantizar la disponibilidad continua del sistema y que se pueda soportar el alto crecimiento que está llevando la empresa.

Alcances del proyecto

Uno de los alcances principales del proyecto es lograr la migración de una arquitectura monolítica a una arquitectura de microservicios, para ello necesitaremos una base de datos, una buena gestión de sucursales, un portal web de clientes y una buena capacitación para los empleados y administradores del software que les daremos.

Objetivos del desarrollo

Como principales objetivos de desarrollo tenemos el construir la arquitectura de microservicios, garantizar la mayor disponibilidad del servicio , optimizar los procesos críticos y también mejorar la transmisión de datos, productos por la vía web mejorando el rendimiento, latencia, y disponibilidad de servidores.

Descripción de la solución

Descripción del sistema ideado en base a los requerimientos y visión del Proyecto

En base a los requisitos que recopilamos, podemos crear un software el cual es escalable, cuenta con medidas de seguridad varias, está desarrollado en base a una arquitectura de microservicios, y contará con una disponibilidad del 99.9%.

Alcances del Producto - Premisas y restricciones

Premisas

El sistema contará con las siguientes funciones clave:

- Módulo de administración: Gestión de usuarios, permisos, monitoreo del sistema, y respaldo de datos.
- Módulo de ventas: Registro de ventas, atención a devoluciones y reclamaciones, emisión de facturas y consultas de inventario.
- Módulo de sucursales: Gestión de inventario, creación de reportes, gestión de pedidos y configuración de sucursales.
- Módulo de logística: Gestión de envíos, optimización de rutas de entrega y gestión de proveedores.
- Portal web para clientes : Funcionalidades para que los clientes creen cuentas, naveguen por productos, realicen pedidos, consulten el historial de compras y gestionen su perfil.

Restricciones

Esta subsección describe aquellas limitaciones que se imponen sobre los desarrolladores del producto:

- Políticas de la empresa: Se deberán cumplir medidas de seguridad acorde a la empresa.
- Limitaciones del hardware: Se deberá conocer cuál es el presupuesto del que se dispone para conseguir hardware para llevar a cabo el software.
- Interfaces con otras aplicaciones: El sistema tendrá integrado Apis de terceros como webpays y chilexpress.

- Operaciones paralelas: El software podrá realizar operaciones paralelas.
- Funciones de auditoría.: Se podrá conocer las acciones realizadas por cada usuario.
- Funciones de control: Se podrá pausar por partes o por completo el software en caso de mantenimiento.
- Lenguaje(s) de programación: Java (con Spring Boot), y utilizará tecnologías como Docker, Kubernetes y bases de datos relacionales
- Protocolos de comunicación: El sistema utilizará APIs de tipo Rest.
- Requisitos de habilidad: Todos los usuarios correspondientes a sus áreas deberán tener un básico conocimiento sobre el uso de un computador y dispositivo móvil.
- Criticalidad de la aplicación: El sistema debe ser altamente disponible (99.9% de uptime), con mecanismos de recuperación ante fallos.
- Consideraciones acerca de la seguridad: Implementación de SSL/TLS, autenticación multifactor (MFA), y protección contra ataques como XSS e inyección SQL.

Especificaciones técnicas de las herramientas de desarrollo

Spring Boot Java: es una herramienta que permite desarrollar aplicaciones web, microservicios con Java, siendo una extensión de marcos de trabajo SPRING, la cual facilita el desarrollo de aplicaciones con dificultades más extremas.

Esta herramienta cuenta con ventajas de reducción de desarrollo a largo plazo, aumenta su productividad, facilita la creación de autonomía en aplicaciones y simplifica su configuración.

MYSQL: Es un sistema que permite la gestión de bases de datos de código abierto, basados en álgebra relacional y el almacenaje de datos de servicios web. Content Management System (CMS) utiliza MYSQL.

Servicios Rest: es una infraestructura ligera para diseñar aplicaciones que utilizan HTTP para realizar llamadas. REST utiliza HTTP para realizar operaciones Create, Read, Update y Delete (CRUD) entre cliente y servidor.

Postman: Es una popular herramienta utilizada para probar APIs, permitiendo a los desarrolladores enviar peticiones a servicios web y ver respuestas. Explicaremos cómo Postman puede ser una herramienta esencial para probar APIs eficientemente.

JWT: Es un estándar abierto que permite compartir información de forma segura entre partes, usando objetos JSON. Se usa para autenticar usuarios y transmitir información entre aplicaciones.

oauth2.0: es un protocolo de autorización que permite a las aplicaciones acceder a recursos protegidos sin exponer las credenciales de los usuarios. Es un estándar abierto que se utiliza para compartir información entre sitios web y aplicaciones.

Docker : Es una plataforma que permite empaquetar una aplicación junto con todas sus dependencias (bibliotecas, configuración, etc.) dentro de un contenedor. Esto facilita la implementación y ejecución de aplicaciones de manera aislada y consistente, sin importar el entorno.

Kubernetes :Es un sistema de orquestación de contenedores que gestiona la implementación, escalado y operación de aplicaciones contenerizadas (como aquellas que utilizan Docker). Se usa para asegurar que las aplicaciones funcionen de manera eficiente en un entorno distribuido.

RabbitMQ: Es un broker de mensajes que permite la comunicación entre sistemas de manera asíncrona y desacoplada. Se utiliza para gestionar colas de mensajes y asegurar que los servicios se comuniquen entre sí sin necesidad de estar directamente conectados.

Kafka: Similar a RabbitMQ, Kafka es un sistema de mensajería distribuido, pero con un enfoque en la alta escalabilidad y rendimiento para manejar flujos de datos en tiempo real.

Memcached: Es una solución de caching distribuido que almacena datos en la memoria para reducir la carga de las bases de datos y mejorar la velocidad de acceso.

Lazy Loading: Es un patrón de diseño que retrasa la carga de un objeto hasta el momento en que se necesita por primera vez. Se usa para optimizar el rendimiento al evitar cargar datos o componentes que no se van a utilizar inmediatamente. En el frontend, puede usarse para cargar productos en un catálogo sólo cuando el usuario hace scroll, en lugar de cargar todos los productos de una vez..

Tipo de Infraestructura de Hardware y Sistemas de implementación y/o servicios a utilizar

Servidores Backend:

- Procesador: mínimo 4 vCPU.
- Memoria RAM: 8 GB por instancia de microservicio.
- Almacenamiento: SSD, redundante, mínimo 100 GB por nodo.

Estaciones de trabajo (para operadores administrativos):

- Navegador actualizado (Chrome, Firefox, Edge).

- Conexión a internet estable.
- Impresoras compatibles para facturación (PDF o conexión a servicios de impresión).

Dispositivos móviles (clientes):

- Android e iOS: acceso vía navegador.
- Pantallas táctiles compatibles con el diseño UI responsivo.

Este software se implementará en los servicios de la nube de AWS.

Organización del equipo Proyecto

Javier Albornoz se tendrá que dirigir por los tramos más mortíferos hasta llegar a una democracia contra el cliente con el cual realizar sus pedidos y no sucumbir a una perdición del equipo completo

Diego Neira se verá envuelto en una encrucijada donde tendrá que poner a prueba el funcionamiento del sistema para encontrar los errores, hacer revisiones generales y solicitar cambios en caso de fallas.

Vicente hormazabal empezará su camino con muchos baches de errores por letras en el código del sistema, tendrá que luchar contra cada línea de código para generar un sistema funcional y entregarle al cliente

Aprobaciones y control de cambios

| Versión | Nombre | Rol | Fecha | Firma |
|---------|--------------------|--------------|------------|-------|
| 1.0 | Javier Albornoz | Scrum Master | 03/04/2025 | jalb |
| 1.0 | Vicente Hormazabal | Developer | 03/04/2025 | V |
| 1.0 | Diego Neira | QA | 03/04/2025 | Dn |

Análisis de gestión de tiempo y costo

- Duración total:

16 semanas (4 sprints de 2 semanas).

- Sprints claves:

Sprint 1: usuarios y autenticación

Sprint 2: ventas

Sprint 3: pedidos y envíos

Sprint 4: Clientes web.

- Costo estimado:

entre \$50,000 y 100,000 dólares.

- Recursos:

Buscamos como recursos para el proyecto serían JavaScript, HTML, CSS, MYSQL. necesitaremos un personal humano con el cual se evaluará las capacidades del sistema y revisiones de errores en momentos, con el cual el personal facilitara la inducción del cliente via web para que las tareas sean más conformable. Se impartirá que el sistema o web tenga los productos de rendimiento más actualizados(la mejor latencia, el mejor rendimiento, la mejor seguridad, etc). La web se podrá abrir en cualquier lugar para un trabajo remoto. recursos de infraestructura tendríamos como tal que en sucursales se proporciona un pc/computador para realizar la acciones desde ahí.