

# “INFORME EVALUACIÓN TRANSVERSAL INGENIERÍA DE SOFTWARE”

“Caso perfulandia”

Sección:001D

Docente: Francisco Valdivieso

Integrantes: Javier Albornoz

Vicente Hormazabal

Diego Neira

## Introducción

En este informe se detalla las 3 evaluaciones realizadas en lo largo del semestre , donde se creó un prototipo,se identificaron las etapas del ciclo de vida del software, se realizaron diagramas de Clase de Componentes de Actividad, de Despliegue y Casos de Uso , se identificaron el Alcance y los Objetivos del Proyecto se definieron los roles y responsabilidades del equipo de trabajo y la metodología de trabajo.

## 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 : Miguel Tapia      Cargo : Jefe

Departamento / División :departamento de finanzas

## Gerente de Proyecto

Nombre:Jair Pino    Departamento / División:departamento de logística y operaciones

Cargo:CEO

### Lista de Interesados (stakeholders)

Nombres : Fabian Zamorano, Sebastian Loyola, Miguel Duran, Michael Smith

Tipos: Administrador, CEO, bodeguista , trabajador

Cargos: Usuario, patrocinador , Gerente de la sucursal , jefe de logistica

Departamento / División: Accionista mayor de Perfulandia , Gerente de ventas, administración , logística

### 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 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 utilizara 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.

oauth 2.0: 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 responsive. 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

### 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.

## DIAGRAMAS CREADOS

### FUNDAMENTOS PATRONES DE MODELOS

Los diagramas de casos se tomaron en cuenta los distintos usuarios para poder conectarlo(como: gerente, logística ,cliente, administrador, trabajador y también se agregó el producto para complementar). Dentro de estas búsqueda se decidió crear 2 distintos diagramas de

comunicación la cual permite ver cómo los usuarios cliente y gerente, comunicarán en los diagramas como la web en el sentido de sus acciones dentro de la web. En los diagramas de actividad nos dimos cuenta las situaciones que pueden llegar a pasar dentro de la web y las soluciones que se pueden usar en momentos negativos. Y por último especificamos cómo funcionan las web en distintos usuarios con los diagramas de secuencia.

## IMÁGENES DE LOS DIAGRAMAS

### [Proyecto Software](#)

## PLAN DE PRUEBAS

1-Propósito En este documento tenemos como propósito principal el definir y describir a detalle el plan de pruebas para nuestro sistema ofrecido a perfulandia. Este plan busca asegurar que el nuevo prototipo web cumpla con los requerimientos de funcionalidad, diseño, usabilidad y experiencia de usuario.

### 2-Alcance

En este plan de pruebas se abarca la validación de nuestro prototipo web para la empresa perfulandia. Las pruebas estarán orientadas a verificar los principales flujos de navegación, la experiencia de usuario, la correcta disposición de elementos visuales y sobre todas las cosas, la coherencia general del diseño.

### 3-Descripción

El sistema ofrecido a perfulandia consiste en una nueva plataforma web de comercio electrónico, el objetivo de esto es mejorar la experiencia de los clientes, reducir errores del sistema anterior y fortalecer la presencia digital de la empresa debido a su exponencial crecimiento y expansión

Características clave del sistema :

Navegación clara y estructurada : Nuestro diseño cuenta con un menú de navegación principal visible y ordenado. lo que permite al usuario navegar fácilmente dentro del sitio

Catálogo de productos: Se muestran los productos de manera estructurada, donde se incluye el nombre, la imagen y el precio. Se contempla un sistema de categorías o filtros para que la búsqueda sea más estructurada y fácil.

Buscador de productos : el sistema incluye un campo de búsqueda (simulado) que permite al usuario encontrar productos de manera rápida

Carrito de compras : se incluye la funcionalidad de agregar productos al carrito, con una vista resumen que muestra lo seleccionado, el total a pagar y opciones para continuar el proceso

Configuración: se incluye la sección donde los usuarios podrán enviar consultas o solicitudes mediante formulario.

interacción fluida entre pantallas : todos los botones están conectados por lo cual la navegación será fluida entre distintas secciones del sitio.

### 4- Resumen de las pruebas

En esta sección del informe detallaremos de manera general las pruebas que se le aplicaran al sistema:

4.1 Módulos a probar  
Página de inicio : validación del diseño general, enlaces funcionales y coherencia visual.  
Catálogo de productos : Visualización de detalles e interacciones básicas  
Detalle del producto : Visualización de imagen, precio, descripción y disponibilidad  
4 Login: Validación de “Olvidaste tu contraseña” y “Crear cuenta”  
Carrito de compras : Agregar productos, visualizar resumen del pedido.

#### 4.2 Objetivos específicos de las pruebas

- Confirmar que todos los flujos definidos en el prototipo funcionan correctamente
- Detectar errores de navegación o desconexión entre pantallas
- obtener retroalimentación de usuarios preliminares (testers internos o stakeholders)

#### 4.3 Tipos de pruebas a realizar

Prueba de navegación : Validación de enlaces, botones y flujos de pantallas

Pruebas de usabilidad : Evaluar si los usuarios pueden completar tareas comunes

Pruebas de UI visual : Verificación del diseño: coherencia visual, alineación, etc

#### 4.4 Técnicas de pruebas aplicadas

- Pruebas basadas en escenarios : Validación de tareas típicas como “buscar un producto y agregarlo al carrito ”
- Checklist visual : uso de listas de verificación para validar elementos gráficos y de contenido.
- Feedback de usuario: Recolección de comentarios durante la prueba del prototipo

#### 4.5 Roles y responsabilidades

Diseñador UX/UI : Responsable del diseño del prototipo y ajustes post testeos.

Tester : Ejecutar los casos de prueba, documentar resultados y reportar errores.

stakeholders internos : participan en la validación funcional y proporcionan feedback.

Jefe de proyecto : Registra hallazgos y elabora informes acerca de los resultados.

#### 5. Entorno y Configuración de las Pruebas

En esta sección se describen de manera detallada los requisitos de software y de hardware necesarios para llevar a cabo las pruebas del sistema.

##### 5.1 Requisitos de software Sistema operativo:

Para poder ocupar nuestro sistema no es necesario contar con un sistema operativo en específico, ya que estamos desarrollando una pagina web, solo se necesitará tener un computador (laptop o de escritorio) y que tenga conexión a internet

Navegador web compatible : Google chrome, Microsoft Edge, Brave, Opera Gx

Dependencias adicionales: conexión a internet y tarjeta de débito o crédito para completar el proceso de compra

##### 5.2 Requisitos de hardware

Computadores: Procesador Intel Core i3 o superior, 8GB RAM (recomendación), resolución mínima 1366x768

Conexión a internet estable : Velocidad recomendada 10 Mbps o superior

### 5.3 Entorno de pruebas

El entorno de pruebas es el entorno de prototipado visual interactivo provisto por Figma, donde el equipo de diseño ha definido flujos de navegación simulados.

- Tipo de entorno: Prototipo navegable en Figma
- Acceso: Enlace compartido por el equipo de diseño (editable o sólo visualización, según el caso)
- Permisos: Acceso con cuenta Figma (puede requerir permiso para comentarios)
- Revisión simultánea: Posibilidad de hacer pruebas en equipo o en sesiones colaborativas

## 6. Calendarización de las Pruebas

Revisión inicial del prototipo

Responsable(s): Tester / UX Designer

Fecha de inicio: 02/06/2025

Fecha de término: 03/06/2025

Duración estimada: 1 día

Ejecución de pruebas de navegación

Responsable(s): Tester

Fecha de inicio: 04/06/2025

Fecha de término: 05/06/2025

Duración estimada: 1 día

Validación visual (UI) y de usabilidad (UX)



Responsable(s): Tester + usuarios

Fecha de inicio: 06/06/2025

Fecha de término: 08/06/2025

Duración estimada: 2 días

Pruebas en vista escritorio

Responsable(s): Tester

Fecha de inicio: 10/06/2025

Fecha de término: 11/06/2025

Duración estimada: 1 día

Recopilación de feedback

Responsable(s): Documentador

Fecha de inicio: 12/06/2025

Fecha de término: 13/06/2025

Duración estimada: 1 día

Análisis de resultados y propuesta de mejoras

Responsable(s): Equipo de diseño

Fecha de inicio: 14/06/2025

Fecha de término: 15/06/2025

Duración estimada: 1 día

Elaboración del informe final de pruebas

Responsable(s): Documentador

Fecha de inicio: 17/06/2025

Fecha de término: 18/06/2025

Duración estimada: 1 día

## 7-Resumen de riesgos

El proceso de pruebas puede verse afectado por distintos factores. A continuación, se detallan los principales riesgos, su impacto, probabilidad y estrategias de mitigación:

Riesgo: Acceso limitado al prototipo en Figma o problemas de permisos

Impacto: Alto

Probabilidad: Media

Estrategia de mitigación: Verificar accesos antes del inicio y coordinar con el equipo de diseño.

Riesgo: Cambios de última hora en el diseño del prototipo

Impacto: Medio

Probabilidad: Alta

Estrategia de mitigación: Congelar versión de prueba antes de comenzar y documentar cambios.

Riesgo: Falta de disponibilidad de testers

Impacto: Alto

Probabilidad: Media

Estrategia de mitigación: Programar sesiones con antelación y tener testers de respaldo si es posible.

Riesgo: Equipos de prueba con bajo rendimiento (dispositivos lentos)

Impacto: Medio

Probabilidad: Media

Estrategia de mitigación: Probar previamente en los dispositivos disponibles.

Riesgo: Feedback poco claro o contradictorio

Impacto: Medio

Probabilidad: Alta

Estrategia de mitigación: Estandarizar formato de recolección de comentarios mediante formularios.

Riesgo: Dificultad para simular ciertas funcionalidades no prototipadas

Impacto: Medio

Probabilidad: Media

Estrategia de mitigación: Aclarar limitaciones del prototipo desde el inicio.

Riesgo: Descoordinación entre roles del equipo

Impacto: Medio

Probabilidad: Media

Estrategia de mitigación: Definir claramente responsabilidades y mantener comunicación fluida.

## 8. Condiciones para el Cierre del Proyecto

El plan de pruebas del prototipo web de Perfulandia se considerará finalizado una vez se cumplan las siguientes condiciones :

- Ejecución completa de las pruebas definidas: Todos las pruebas han sido ejecutadas en el prototipo
- Identificación y documentación de hallazgos : Todos los errores han sido detectados a tiempo y documentados con fin de solucionarlos
- recepción de retroalimentación de usuarios : Se recolecta y se aplica feedback de los testers/ usuarios
- propuestas de mejora entregada : Se crea una lista de sugerencias de mejora basada enb los testeos anteriores.
- informe final entregado : Se elabora un informe final con el resumen de pruebas realizadas, hallazgos, observaciones y recomendaciones
- aprobación por parte del equipo : el equipo responsable del proyecto ha validado el cierre de proceso de pruebas.