Logotipo

Descripción generada automáticamente con confianza media

**Especificación de requisitos de software**

Proyecto: VinoVivo

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Verificado Calidad.** |
| 14/05/2024 |  | Grupo 4 – Digital House – Proyecto Integrador II |  |

**Documento validado por las partes en fecha:**

|  |  |
| --- | --- |
| **Por el proveedor** | **Por el cliente** |
| Digital House | Verónica Valdez |

**Contenido**

[Introducción 4](#_Toc169037044)

[**1.1 Propósito 4**](#_Toc169037045)

[**1.2 Alcance 5**](#_Toc169037046)

[**1.3 Personal 5**](#_Toc169037047)

[**1.4 Definiciones, acrónimos y abreviaturas 5**](#_Toc169037048)

[**1.5 Referencias 5**](#_Toc169037049)

[**1.6 Resumen 5**](#_Toc169037050)

[2. Descripción General 6](#_Toc169037051)

[**2.1. Perspectiva del producto 6**](#_Toc169037052)

[**2.2. Funcionalidad del producto 6**](#_Toc169037053)

[**2.3. Características de los usuarios 6**](#_Toc169037054)

[**2.4. Restricciones 7**](#_Toc169037055)

[**2.5. Suposiciones y dependencias 7**](#_Toc169037056)

[3. Requisitos específicos 7](#_Toc169037057)

[**3.1. Requerimientos Funcionales 7**](#_Toc169037058)

[**3.2. Requerimientos No Funcionales. 10**](#_Toc169037059)

[**3.3. Requisitos comunes de las interfaces 13**](#_Toc169037060)

[**3.3.1. Interfaces de usuario 13**](#_Toc169037061)

[4. Historias de Usuario 13](#_Toc169037062)

[**4.1. Definición de Filtros: 13**](#_Toc169037063)

[**4.2. Creación de Base de Datos: 14**](#_Toc169037064)

[**4.3. Creación de Infraestructura: 14**](#_Toc169037065)

[**4.4. Creación de Backend API: 14**](#_Toc169037066)

[**4.5. Creación de encabezado: 15**](#_Toc169037067)

[**4.6. Creación del cuerpo del sitio: 16**](#_Toc169037068)

[**4.7. Creación de Card y visualizar 4 productos en el Home: 16**](#_Toc169037069)

[**4.8. Creación de Footer páginas: 16**](#_Toc169037070)

[**4.9. Creación de un paginado en las categorías de productos: 17**](#_Toc169037071)

[**4.10. Creación de una Página del concepto del restaurante de cata: 17**](#_Toc169037072)

[**4.11. Creación de una Página de Contacto. 17**](#_Toc169037073)

[**4.12. Planificación y ejecución de los Tests: 18**](#_Toc169037074)

[5. Historias de Usuario SPRINT 2 19](#_Toc169037075)

[**5.1. Ver bloque de detalle: 19**](#_Toc169037076)

[**5.2. Registrar Producto: 20**](#_Toc169037077)

[**5.3. Desplegable de productos por categoría: 20**](#_Toc169037078)

[**5.4. Vista de productos según categoría seleccionada: 20**](#_Toc169037079)

[**5.5. Vista de producto: 21**](#_Toc169037080)

[**5.6. Vista “mi perfil”: 21**](#_Toc169037081)

[**5.7. Vista de “mis productos” (Admin): 21**](#_Toc169037082)

[**5.8. Implementar Infraestructura CI/CD: 22**](#_Toc169037083)

[**5.9. Bucket contenido frontend (imágenes): 22**](#_Toc169037084)

[**5.10. Testing: 22**](#_Toc169037085)

[**5.11. Diseño UX/UI: 23**](#_Toc169037086)

[**5.12. Validación Form Contacto: 23**](#_Toc169037087)

[**5.13. Backend Endpoint Order: 23**](#_Toc169037088)

[**5.14. Backend Endpoint Order\_Details: 24**](#_Toc169037089)

[**5.15. Backend Endpoint Type: 24**](#_Toc169037090)

[**5.16. Backend Endpoint Variety: 24**](#_Toc169037091)

[**5.17. Backend Endpoint Winery: 24**](#_Toc169037092)

[**5.18. Backend Endpoint Microservicio User’s: 25**](#_Toc169037093)

[**5.19. Backend y BBDD: 25**](#_Toc169037094)

[6. Historias de Usuario SPRINT 3 26](#_Toc169037095)

[**6.1. Registro de Usuario (Cliente): 26**](#_Toc169037096)

[**6.2. Desplegable Avatar: 26**](#_Toc169037097)

[**6.3. Cerrar Sesión: 27**](#_Toc169037098)

[**6.4. Editar Producto (Admin): 27**](#_Toc169037099)

[**6.5. Diseño UX/UI: 27**](#_Toc169037100)

[**6.6. Reestablecer contraseña: 27**](#_Toc169037101)

[**6.7. Carrito Desplegable: 27**](#_Toc169037102)

[**6.8. Compra en 3 pasos: 28**](#_Toc169037103)

[**6.9. Registro de Datos de envío: 28**](#_Toc169037104)

[**6.10. Mis compras: 29**](#_Toc169037105)

[**6.11. Conexión de Front con Keycloak: 29**](#_Toc169037106)

[**6.12. Gestión de Roles: 29**](#_Toc169037107)

[**6.13. Implementar Infraestructura: 29**](#_Toc169037108)

[**6.14. Testing: 30**](#_Toc169037109)

[**6.15. Crear microservicio de Usuario: 30**](#_Toc169037110)

[**6.16. Crear microservicio de Gateway: 30**](#_Toc169037111)

[**6.17. Crear Base de Datos Keycloak: 30**](#_Toc169037112)

[**6.18. Seguridad en la conexión del cliente: 31**](#_Toc169037113)

[**6.19. Verificación de cuenta de correo: 31**](#_Toc169037114)

[**6.20. Conexión de Gateway con los demás microservicios: 31**](#_Toc169037115)

[**6.21. Modificar las configuraciones de Gateway para que se pueda interactuar desde front: 31**](#_Toc169037116)

[**6.22. Despliegue de réplicas de los microservicios como Backus ante fallos: 31**](#_Toc169037117)

[**6.23. Implementación de configuración centralizada: 31**](#_Toc169037118)

# Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el Sistema de información, para la gestión de procesos y control de inventarios VinoVivo. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

* 1. **Propósito**

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de un sistema de información web que permitirá gestionar distintos procesos administrativo. Éste será utilizado por dueño de la empresa.

* 1. **Alcance**

Esta especificación de requisitos está dirigida al usuario del sistema, para continuar con el desarrollo de aplicaciones web y para profundizar en la automatización de esta, la cual tiene por objetivo principal el gestionar los distintos procesos administrativos (Inventario, Eventos, Registros).

* 1. **Personal**

|  |  |
| --- | --- |
| Nombre |  |
| Rol |  |
| Responsabilidad |  |

* 1. **Definiciones, acrónimos y abreviaturas**

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Usuario | Persona que usará el sistema para gestionar sus compras |
| VV | VinoVivo (Grupo especialista) |
| ERS | Especificación de Requisitos Software |
| RF | Requerimiento Funcional |
| RNF | Requerimiento No Funcional |
| FTP | Protocolo de Transferencia de Archivos |

* 1. **Referencias**

|  |  |
| --- | --- |
| Título del Documento | Referencia |
| Standard IEEE 830 - 1998 | IEEE |

* 1. **Resumen**

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

# 

# 2. Descripción General

## 2.1. Perspectiva del producto

Vino Vino es una enoteca & wine-bar boutique está emplazado en el corazón de uno de los barrios más exclusivos de Buenos Aires. Como representación de la digitalización de las ventas ahora el dueño desea generar un comercio electrónico que le permita tener un mayor alcance en las ventas de su negocio y potencializar la marca de su negocio.

## 2.2. Funcionalidad del producto

Generar una Web que permita a los usuarios frecuentes y nuevos clientes generar el consumo de los productos característicos de la enoteca, permitiendo al dueño generar la administración del sistema para que de la misma manera para que pueda administrarlo y quitar y colocar productos según la necesidad y la época en la cual se encuentre que sean de mayor beneficio para su público Objetivo.

* 1. **Características de los usuarios**

|  |  |
| --- | --- |
| Tipo de usuario | Admin |
| Persona con este usuario | Dueño |
| Actividades | Control y manejo del sistema en general |

|  |  |
| --- | --- |
| Tipo de usuario | Cliente |
| Persona con este usuario | Usuarios internos y externos |
| Actividades | Da sus datos para poder ser registrado |

*.*

* 1. **Restricciones**
* Interfaz para ser usada con internet.
* Lenguajes y tecnologías en uso:

**Front**

Next, TypeScript

Tailwind

**Back**

Java

Spring Boot

Keycloak

SQL

Swagger

* Los servidores deben ser capaces de atender consultas concurrentemente.
* El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

*.*

* 1. **Suposiciones y dependencias**
* Se asume que los requisitos aquí descritos son estables
* Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta de la misma

# Requisitos específicos

## 3.1. Requerimientos Funcionales

|  |  |
| --- | --- |
| Identificación del  requerimiento: | RF01 |
| Nombre del  Requerimiento: | Gestión de Usuarios. |
| Características: | * Registro de nuevos usuarios con campos como nombre, correo electrónico y contraseña. * Inicio de sesión para usuarios registrados con autenticación segura. * Capacidad para restablecer contraseñas en caso de olvido. * Perfiles de usuario con la posibilidad de actualizar la información personal. |
| Descripción del  requerimiento: | El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| Identificación del  requerimiento: | RF02 |
| Nombre del  Requerimiento: | Reporte. |
| Características: | * Generación de un informe con datos relevantes del sistema o aplicación. * Filtros para personalizar el contenido del informe, como rango de fechas o categorías específicas. * Formato del informe que sea fácil de entender y leer para los usuarios. * Opción para exportar el informe a diferentes formatos (por ejemplo, PDF, CSV). |
| Descripción del  requerimiento: | El sistema debe generar reportes exportables por medio del usuario “Admin” dependiendo del módulo en el cual se encuentre. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| Identificación del  requerimiento: | RF03 |
| Nombre del  Requerimiento: | Transacción. |
| Características: | * Capacidad para realizar una transacción o una serie de transacciones con los datos ingresados por el usuario. * Asegurar que las transacciones sean seguras y estén protegidas contra cualquier tipo de manipulación maliciosa o errores. * Proporcionar confirmaciones visuales o notificaciones de éxito o fracaso de la transacción. |
| Descripción del  requerimiento: | El sistema debe tener sistema de alertas ante usos inadecuados de los campos, de igual forma para los errores para que el usuario siempre este enterado en que proceso de la transacción se encuentra. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| Identificación del  Requerimiento: | RF04 (EXTRA) |
| Nombre del  Requerimiento: | Conexión entre Microservicios |
| Características: | * Generar la conexión entre los microservicios con la finalidad de tener accesibilidad a todas las funcionalidades de la página web. |
| Descripción del  requerimiento: | Como cliente, al utilizar la aplicación que ofrece diversas funcionalidades distribuidas en diferentes microservicios, es crucial que el microservicio 'Gateway' se conecte con los demás microservicios para poder acceder a todas las funcionalidades. |
| **Prioridad del requerimiento:**  Baja | |

|  |  |
| --- | --- |
| Identificación del  requerimiento: | RF05 (EXTRA) |
| Nombre del  Requerimiento: | Configuraciones de Gateway – Frontend |
| Características: | * Se requiere una conexión entre el backend y el frontend para que se puedan dar las funcionalidades de forma correcta dentro del sitio. |
| Descripción del  requerimiento: | Como cliente, al utilizar la interfaz de usuario del frontend de la aplicación para acceder a sus funcionalidades, es necesario que las configuraciones del microservicio 'Gateway' se ajusten adecuadamente para garantizar una interacción fluida, y efectiva entre el backend y el frontend. |
| **Prioridad del requerimiento:**  Baja | |

|  |  |
| --- | --- |
| Identificación del  requerimiento: | RF06 (EXTRA) |
| Nombre del  Requerimiento: | Despliegues de Backups - Redundancia |
| Características: | * Generar un ambiente espejo en el cual tengamos la capacidad de generar un DRP de los sistemas en los cuales se encuentra corriendo la aplicación. * Recrear ambientes en diferentes espacios de repositorio para poder tener respaldo suficiente para poder tener el ambiente productivo activo en todo momento |
| Descripción del  requerimiento: | La aplicación debe tener sistema de redundancia la cual permita volver a retomar el sistema ante un desastre o algún incidente mayor presentado en la aplicación. |
| **Prioridad del requerimiento:**  Media | |

## 3.2. Requerimientos No Funcionales.

|  |  |
| --- | --- |
| Identificación del  requerimiento: | RNF01 |
| Nombre del  Requerimiento: | Interfaz del sistema. (Home) |
| Características: | Sección para venta de vinos desde el inicio de la página principal |
| Descripción del requerimiento: | El sistema debe tener una interfaz de uso intuitiva y sencilla. |
| Prioridad del requerimiento: | |

|  |  |
| --- | --- |
| Identificación del  requerimiento: | RNF02 |
| Nombre del  Requerimiento: | Interfaz del sistema. (Concepto) |
| Características: | Sección sobre el concepto de Vino Vivo con imágenes relativas al lugar y la experiencia |
| Descripción del | La interfaz debe estar complementada con la cata de vinos que tiene actualmente el negocio. |
| Prioridad del requerimiento: | |

|  |  |
| --- | --- |
| Identificación del  requerimiento: | RNF03 |
| Nombre del  Requerimiento: | Interfaz del sistema. (Contacto) |
| Características: | Sección de contacto con formulario para dejar mensajes y mapa incluido para facilitar la comprensión de la ubicación |
| Descripción del requerimiento: | El sistema debe disponer de una sección de contacto en donde permite visualizar información y dejar datos importantes de contacto directo con el negocio. |
| Prioridad del requerimiento: | |

|  |  |
| --- | --- |
| Identificación del requerimiento: | RNF04 |
| Nombre del  Requerimiento: | Interfaz del sistema. (Footer) |
| Características: | Sección en pie de página que incluya las redes sociales que tiene el espacio para multiplicar los espacios de contacto y marketing digital. |
| Descripción del requerimiento: | La interfaz de usuario debe tener las redes sociales del negocio para que pueda generar interacción de contacto por los diferentes medios que están disponibles para contactar al negocio. |
| Prioridad del requerimiento: | |

|  |  |
| --- | --- |
| Identificación del requerimiento: | RNF05 |
| Nombre del  Requerimiento: | Diseño de la interfaz |
| Características: | En el sitio deberá predominar el espacio libre, claro y con formas regulares geométricas simples. |
| Descripción del requerimiento: | Garantizar el diseño uniforme de la interfaz para que sea visualmente agradable a los usuarios y llamativo. |
| Prioridad del requerimiento: | |

|  |  |
| --- | --- |
| Identificación del  requerimiento: | RNF06 |
| Nombre del  Requerimiento: | Diseño de la interfaz |
| Características: | Paleta cromática restringida y en tonos neutros de acuerdo con el espacio físico de Vino Vivo |
| Descripción del requerimiento: | Garantizar el estilo de uso de colores dentro del diseño de la interfaz, así como de su tipografía. |
| Prioridad del requerimiento: | |

|  |  |
| --- | --- |
| Identificación del  requerimiento: | RNF07 |
| Nombre del  Requerimiento: | Imágenes |
| Características: | Las imágenes deben tener una calidad mínima de 72 dpi y tamaño de uno de los lados de 1024px |
| Descripción del requerimiento: | La disposición en cuanto al tamaño de las imágenes es relevante para mantener la coherencia dentro del diseño realizado |
| Prioridad del requerimiento: | |

|  |  |
| --- | --- |
| Identificación del requerimiento: | RNF08 |
| Nombre del  Requerimiento: | Imágenes del Lugar Físico (Negocio) |
| Características: | Se mostrarán imágenes sobre el lugar, los productos y los servicios de datas y menú degustación ofrecidos en Vino Vivo |
| Descripción del  requerimiento: | Garantizar la visualización de las instalaciones a los usuarios para que tengan conocimiento del lugar y sus espacios que tienen para visitar. |
| Prioridad del requerimiento: | |

|  |  |
| --- | --- |
| Identificación del requerimiento: | RNF09 |
| Nombre del  Requerimiento: | Productos. (Home) |
| Características: | Se mostrarán la vista central y completa del producto para poder conocer aquello que se compra |
| Descripción del  requerimiento: | Garantizar la visualización de los productos recomendados por parte del negocio. |
| Prioridad del requerimiento: | |

|  |  |
| --- | --- |
| Identificación del requerimiento: | RNF10 |
| Nombre del  Requerimiento: | Diseño de la interfaz Usuario (Alertas, Errores) |
| Características: | Se utilizarán los colores para indicar estados: matices que vienen del rojo, para atención y/o error, y matices que son verdes para informar al usuario sobre la confirmación y/o validez |
| Descripción del  requerimiento: | Garantizar el conocimiento del usuario en la transacción del cliente en la cual se encuentra en su proceso dentro del sitio Web, con colores visibles y ayudas requeridas. |
| Prioridad del requerimiento: | |

* 1. **Requisitos comunes de las interfaces**

*.*

## Interfaces de usuario

La interfaz con el usuario consistirá en un conjunto de ventanas con botones, listas y campos de textos. Ésta deberá ser construida específicamente para el sistema propuesto y, será visualizada desde un navegador de internet.

# Historias de Usuario

Como cliente, deseo tener filtro para poder buscar los tipos de vino dentro de la página web:

## 4.1. Definición de Filtros:

Va en producto y será por tipo:

* Vino tinto
* Vino blanco
* Vino rosado
* Vino espumoso

Como cliente, deseo poder guardar mis órdenes y tener varios productos con las características de Bodega, tipo de vino y variedad de uva para poder identificarlos dentro del sitio web:

## 4.2. Creación de Base de Datos:

* Diseñar el Modelo de Entidad-Relación (DER) para definir la estructura de la base de datos que respaldará la aplicación.
* Crear la tabla de products, esta tabla deberá contener los siguientes atributos: id, name, description, year, price, image, stock, id\_type, id\_winery e id\_variety.
* Crear la tabla de orders, esta tabla deberá contener los siguientes atributos: id, id\_customer, amount, shipping address y order\_email.
* Crear la tabla de order\_details, esta tabla deberá contener los siguientes atributos: id, id\_order, id\_product, price y quantity.
* Crear la tabla de type, esta tabla deberá contener los siguientes atributos: id y name.
* Crear la tabla de variety, esta tabla deberá contener los siguientes atributos: id y name.
* Crear la tabla de winery, esta tabla deberá contener los siguientes atributos: id y name.
* Realizar la carga inicial de datos en la base de datos para asegurar que la aplicación tenga información relevante para su funcionamiento.

Como cliente, deseo poder tener una página con servicios en la nube para una mayor accesibilidad al ecommerce:

## Creación de Infraestructura:

* Configurar y generar la base de datos en AWS para garantizar la escalabilidad, persistencia y disponibilidad de los datos.
* Realizar un diseño detallado de la infraestructura requerida, para que el proyecto funcione de manera óptima en AWS. Además, es crucial crear un boceto claro que represente la red y sus componentes, como servidores, almacenamiento, red interna y base de datos. Este diseño me permitirá comprender la arquitectura del proyecto y asegurar una implementación exitosa en la plataforma AWS.

Como cliente, deseo poder gestionar todos mis servicios de una forma centralizada:

* 1. **Creación de Backend API:**
* Diseñar el diagrama UML (Unified Modeling Language), el cual proporcionará una representación visual de la arquitectura y las interacciones entre los diferentes componentes de la API.
* Desarrollar el proyecto con Java y Spring Boot para implementar la API que contendrá los endpoints del sistema, estructurando el proyecto con los paquetes model, DTO, repository, service y controller.
* Crear el microservicio de commerce que contendrá toda la lógica de negocio.
* Crear el microservicio de discovery que permitirá conectar los microservicios entre sí.
* Mapear los atributos de la tabla “products” con una clase de nuestro modelo.
* Mapear los atributos de la tabla “orders” con una clase de nuestro modelo.
* Mapear los atributos de la tabla “order\_details” con una clase de nuestro modelo.
* Mapear los atributos de la tabla ”type” con una clase de nuestro modelo.
* Mapear los atributos de la tabla ”variety” con una clase de nuestro modelo.
* Mapear los atributos de la tabla ”winery” con una clase de nuestro modelo.
* Conectar la api de backend a la base de datos para poder realizar las operaciones CRUD.
* Desarrollar el endpoint de products para ser consumida por el frontend. El controller deberá tener los métodos necesarios para: agregar un producto nuevo, listar todos los productos, listar un producto en particular por id, editar un producto en particular y eliminar un producto.

Como cliente, quiero que el header de la aplicación cumpla con las siguientes características:

* 1. **Creación de encabezado:**

**Diseño Responsivo y Fijo:**

* El header debe ocupar el 100% de la pantalla en todas las páginas de la aplicación y debe permanecer fijo en la parte superior incluso al hacer scroll, asegurando así su visibilidad y accesibilidad constante para el usuario.

**Consistencia y Usabilidad:**

* El diseño del header debe ser consistente en todas las páginas de la aplicación para garantizar una experiencia de usuario uniforme y debe ser intuitivo y fácil de usar.

**Optimización para Dispositivos:**

* El header debe adaptarse de manera óptima a diferentes dispositivos y resoluciones de pantalla, utilizando el patrón de diseño de hamburguesa para dispositivos móviles.

**Bloques Específicos: ·**

* Bloque Izquierdo: Debe contener el logotipo y el lema de la empresa. Al hacer clic en estos elementos, el usuario será redirigido a la página principal de la aplicación.
* Bloque Central: Debe incluir enlaces a las diferentes páginas como productos, concepto y contacto, facilitando la navegación del usuario.
* Bloque Derecho: Debe albergar el buscador y los conos de usuario y carrito de compras (sin funcionalidad por ahora), alineados a la derecha del header.
* Estos elementos combinados aseguran un diseño coherente, funcional y adaptable del header que mejora la experiencia general del usuario en la aplicación.

Como cliente, quiero que la página principal de la aplicación cumpla con las siguientes características:

* 1. **Creación del cuerpo del sitio:**
* El body debe tener un color de background que corresponda a la identidad de marca.
* El body debe ocupar el 100 % del alto de la pantalla.
* El body debe ser optimizado para diferentes dispositivos y resoluciones de pantalla.
* Se debe visualizar las tres secciones o bloques: buscador/filtros, categorías y recomendaciones de los productos. (Recordar definir las categorías)

Como cliente, quiero que la página principal de la aplicación tenga información de productos recomendados por parte de la vinoteca:

* 1. **Creación de Card y visualizar 4 productos en el Home:**
* En el home se debe ver una lista aleatoria de productos.
* Garantizar que la lista de productos aleatorios sea verdaderamente aleatoria, sin repetir productos y sin seguir un patrón previsible 4 para desktop 1 para mobile.
* 4 para desktop, 1 para mobile.

Como cliente, deseo visualizar un footer con los derechos del copyrigth en la página web, las políticas de privacidad y la posibilidad de suscribirse a sus noticias, con el fin de estar enterado de sus productos y tener la seguridad de que es una página legítima:

* 1. **Creación de Footer páginas:**
* Se debe crear un footer que ocupe el 100% del ancho de la pantalla y que esté ubicado en el pie de página en todas las páginas de la aplicación.
* Dentro del footer, debe existir un bloque alineado a la izquierda que contenga info importante y el copyright correspondiente.

Como cliente, deseo ver las categorías de los vinos en una nueva página:

* 1. **Creación de un paginado en las categorías de productos:**
* Que se pueda visualizar solo una cantidad limitada de X productos.
* Mostrar un botón o enlace para que el usuario pueda ir hacia atrás o al inicio del listado de productos, es decir, una paginación al final de la página.

Como cliente, deseo visitar la página web de un restaurante especializado en catas para conocer más sobre su historia, ver imágenes de eventos pasados y poder reservar una cata:

* 1. **Creación de una Página del concepto del restaurante de cata:**

**Sección de Historia del Restaurante:**

* Quiero poder leer la historia del restaurante, incluyendo su origen, valores y experiencia en catas. La sección debe contener tanto texto como imágenes para ofrecer una experiencia visual completa y atractiva.

**Galería de Imágenes de Catas:**

* Me gustaría ver una galería de imágenes que muestre eventos pasados de catas realizadas en el restaurante. Las imágenes deben proporcionar una vista realista de la experiencia de las catas.

**Sección de Reserva de Catas:**

* Quiero encontrar una sección específica que me permita reservar una cata en el restaurante. La sección debe incluir:
  + - Una imagen atractiva relacionada con el concepto de la cata, situada en el lateral izquierdo.
    - Un texto motivador en el lateral derecho que invite a reservar una cata y transmita la experiencia única que ofrece el restaurante.
    - Un botón claro y llamativo debajo del texto que me lleve a la página de contacto para realizar la reserva.

Como cliente, deseo ponerme en contacto y saber la dirección del restaurante para poder asistir a catas y eventos:

* 1. **Creación de una Página de Contacto.**

**Sección de Formulario:**

* El formulario solicitará al usuario completar los siguientes campos:
  + - Nombre, Celular, Email, y un campo opcional en el cual se podrá dejar un breve comentario o mensaje para que se pongan en contacto y recibir información del restaurante.

**Sección de ubicación:**

* Quiero encontrar una sección que me lleve a mapas para saber qué tan lejos o cerca se encuentra la vinoteca, además de poder saber sus horarios. La sección debe incluir:
  + - Un texto donde podrá obtener la dirección, teléfono y horarios de la vinoteca.
    - Además del mapa, el cual me mostrará la ubicación y con el cual al clickear me permitirá ingresar y obtener el recorrido hasta el mismo.

Como cliente, deseo que la página web presente la menor cantidad de errores cuando este activa para el ingreso de los usuarios:

* 1. **Planificación y ejecución de los Tests:**
* Planear casos de prueba basados en las historias de usuarios que existen en el Sprint, y luego ejecutarlos.
* Realizar también una prueba exploratoria sobre lo que se desarrolló durante el Sprint.
* Validar el funcionamiento de la API Rest creada por el Back End durante el Sprint. Utilicen Postman para las pruebas y creen los scripts de prueba necesarios.

**SPRINT 2**

# Historias de Usuario SPRINT 2

Como usuario, quiero visualizar un bloque de encabezado y un pie de página según lo descrito en las historias de usuario correspondientes:

## 5.1. Ver bloque de detalle:

* En el cuerpo de la página, deseo ver la información detallada de un vino elegido, dividida en dos secciones: izquierda y derecha. La imagen del vino debe ubicarse a la izquierda de la tarjeta, mientras que los detalles del producto deben mostrarse en el lado derecho de la tarjeta, con los campos definidos claramente.
* La flecha para volver atrás debe estar alineada a la derecha, en la parte inferior de la tarjeta, para facilitar la navegación del usuario.
* La vista debe contar con un título general con la primera palabra del nombre del vino
* La información detallada del vino debe incluir los siguientes campos:
* Nombre del vino
* Descripción
* Tipo de vino
* Bodega
* Variedad de uva
* Año
* Stock disponible
* Precio
* Al final de la información detallada, deben colocarse dos botones de acción:
* "Agregar icono-carrito”, para que los usuarios puedan realizar la acción de compra.
* El botón de volver atrás en la navegación con forma de <

Como administrador, quiero acceder a la página "Agregar producto" para poder ingresar información relevante sobre un nuevo producto.

## 5.2. Registrar Producto:

* La página "Agregar producto" debe incluir campos para ingresar información nombre, bodega, variedad de uva, descripción, precio, año de producción, cantidad de producto idéntico disponible (stock) e imagen del producto.
* El tipo de vino, la bodega y la variedad de uva debe ser un desplegable con las opciones de vino que manejamos
* Una vez que se ingresen los datos del producto, el administrador debe tener la opción de guardar el producto.
* Cuando se dé clic en guardar el producto, este debe ser agregado correctamente a la base de datos del sitio y mostrar un mensaje de éxito.
* Debe contar con la posibilidad de volver atrás con un botón alineado a la derecha, en la parte superior, al lado del título de la página, para facilitar la navegación del usuario.
* Si se intenta agregar un producto, sí éste ya existe, en la base de datos, se debe mostrar un mensaje de error.
* Todos los campos son obligatorios y deben tener validaciones básicas como cantidad de caracteres o tipo de entrada (texto o número) y mostrar al usuario el mensaje de que debería tener cada uno o que este es obligatorio en caso de que intente guardarlo con uno o varios campos vacíos.

Como cliente, deseo visualizar un desplegable en “productos”, donde pueda seleccionarlos por categoría y que estos me lleven a una vista donde pueda visualizar todos los productos clasificados por la misma.

## Desplegable de productos por categoría:

* El usuario debe ver una lista/ desplegable al hacer click en home/navbar/productos con las categorías (tintos, blancos, rosados, espumosos) en los que a su vez pueda hacer click para ver en otra vista los productos clasificados según la categoría seleccionada.

Como cliente, deseo visualizar los productos clasificados/filtrados según la categoría seleccionada y que estos estén paginados para verlos cómodamente.

* 1. **Vista de productos según categoría seleccionada:**
* El usuario debe ver un título que le confirme la categoría de vinos que se encuentran en la vista.
* El usuario debe poder visualizar una cantidad limitada de 8 productos (de la categoría seleccionada desde el desplegable) por página. El paginado debe estar debajo, al finalizar los productos.
* El usuario debe poder ir hacia atrás en la navegación de la página, así, sí inicialmente estaba en home, pueda regresar desde “vinos tintos”, por ejemplo, de nuevo a home; con un botón en la parte superior izquierda.

Como usuarios quiero poder ver todos los productos sin clasificación y paginados.

* 1. **Vista de producto:**
* Estando en home, al hacer click en “mostrar más”, debe dirigir al usuario a una vista donde encuentre todos los vinos sin clasificación y paginados.
* La vista debe tener un título que ubique al usuario en la vista de “productos”.
* El usuario debe poder visualizar cómodamente en la pantalla desktop 8 productos, 4 en una línea y 4 más debajo.
* Los productos deben estar paginados Max 8 x pág.
* El usuario debe poder volver atrás en la navegación con un botón.

Como usuario quiero poder ver mi información personal y poder modificarla e ir desde allí a otras partes.

* 1. **Vista “mi perfil”:**
* El usuario deberá poder modificar cada uno de los campos, a través de un botón que despliegue el formulario con los datos previos y pueda modificarlos, el formulario cuenta con un botón de guardar, al cerrarlo debe actualizarse la vista con la nueva información.
* El usuario puede ver dos botones, un con el que puede navegar desde allí a “mis productos” en caso de ser admin y el otro que por ahora no lleva a ningún lado con el nombre de “mis compras” para el caso de cliente.

Como usuario admin quiero ver todos los productos que he creado en un solo lugar, poder editarlos y eliminarlos. Solo el administrador tendrá acceso a estas funcionalidades.

* 1. **Vista de “mis productos” (Admin):**
* El usuario podrá encontrar un título que le confirme la ruta en la que se encuentra de “mis productos”, donde podrá ver todas las tarjetas de los productos creados con anterioridad, sino tiene ninguno, habrá un mensaje “Aún no tiene productos creados, da click en “crear producto””
* Las cards son como las de “productos” o recomendaciones del home, y hacer click, pero está en vez de “comprar” tendrá el botón de “editar” y en la esquina superior derecha una X o icono de basura para eliminar.
* Al hacer clic en el icono de eliminar sale una ventana emergente que le preguntará si desea eliminar el vino nombre del vino, un botón de confirmar y un botón de cancelar.
* El usuario cuenta con 1 botón para “crear productos”, que lo lleva a la ruta de product/register con el formulario para crearlo.
* Los productos deben ser paginados de a 8 por página y 4 por línea en desktop.

Como cliente quiero poder tener todo integrado para se vea reflejado de forma inmediata dentro de la página:

* 1. **Implementar Infraestructura CI/CD:**
* Desplegar BBDD ecommerce en RailWay.
* Desplegar BBDD users en RailWay.
* Deplegar KeyCloak en RailWay.
* Desplegar microservicio de ecommerce en RailWay.
* Desplegar microservicio de discovery en RailWay.
* Desplegar en FrontEnd en Vercel

Como cliente quiero poder tener mis imágenes en un acceso público el cual pueda administrar fácilmente:

* 1. **Bucket contenido frontend (imágenes):**
* Crear S3 Bucket para almacenar las imágenes de los productos:
* Crear bucket en AWS.
* Almacenar las imágenes de los productos y añadir las URLs a la base de datos.

Como usuario deseo que la página sea testeada con lo realizado para no tener problemas al salir con la página:

* 1. **Testing:**
* Creación de pruebas automatizadas
* Casos de prueba manuales con:

● Agregar a la planilla de casos de prueba sobre las funcionalidades del sprint II.

● Realizar mantenimiento de los casos de prueba manuales generados en el sprint I, en caso de ser necesario.

Como usuario deseo que se me muestren los diseños de cómo está cada página para dar una opinión en cuanto a si se requiere algún cambio:

* 1. **Diseño UX/UI:**
* Crear vista del perfil -> hay indicaciones básicas
* Crear vista del detalle de producto -> ya está
* Crear vista de registrar producto -> hay base
* Crear vista de mis productos -> hay diseño base
* Crear vista del carrito -> sin desarrollo en front

Como usuario quiero poder ingresar a la página de contacto y completar información para que se pongan en contacto.

* 1. **Validación Form Contacto:**
* El usuario deberá proporcionar información precisa y válida, como: - nombre, - celular, - dirección de correo electrónico, y podrá, dejar un mensaje.
* Todos los campos, excepto el de mensaje, son obligatorios y deben tener validaciones básicas de cantidad mínima de caracteres, tipo de entrada (texto, correo, números, etc.) y mostrar mensajes de error acordes al usuario si este intenta enviar el formulario sin rellenar todos los campos obligatorios.
* El usuario deberá ver en la vista un título que le confirme que está viendo el formulario de contacto.
* El usuario debe ver un botón al final del formulario, que diga “enviar” para finalizar dicho proceso; este a su vez dispara las validaciones y en caso de estar todo bien, mostrar un modal, donde se le notificará que el formulario fue enviado y que pronto lo contactaran.

Como usuario deseo poder gestionar las ordenes en el sistema:

* 1. **Backend Endpoint Order:**
* La api debe contar con un método para agregar una nueva orden.
* La api debe contar con un método para listar todas las órdenes.
* La api debe contar con un método para obtener una orden por su ID.
* La api debe contar con un método para editar una orden por su ID.
* La api debe contar con un método para eliminar una orden por su ID.

Como usuario deseo poder gestionar los detalles de las órdenes en el sistema.

* 1. **Backend Endpoint Order\_Details:**
* La api debe contar un método para agregar un nuevo detalle de orden.
* La api debe contar un método para listar todos los detalles de órdenes.
* La api debe contar un método para obtener un detalle de orden por su ID.
* La api debe contar un método para editar un detalle de orden por su ID.
* La api debe contar un método para eliminar un detalle de orden por su ID.

Como usuario deseo poder gestionar los tipos de vino en el sistema.

* 1. **Backend Endpoint Type:**
* La api debe contar un método para agregar un nuevo tipo.
* La api debe contar un método para listar todos los tipos.
* La api debe contar un método para obtener un tipo por su ID.
* La api debe contar un método para editar un tipo por su ID.
* La api debe contar un método para eliminar un tipo por su ID.

Como usuario deseo poder gestionar las variedades de uva del vino desde el sistema.

* 1. **Backend Endpoint Variety:**
* La api debe contar un método para agregar una nueva variedad.
* La api debe contar un método para listar todas las variedades.
* La api debe contar un método para obtener una variedad por su ID.
* La api debe contar un método para editar una variedad por su ID.
* La api debe contar un método para eliminar una variedad por su ID.

Como usuario deseo poder gestionar las bodegas de vino en el sistema.

* 1. **Backend Endpoint Winery:**
* La api debe contar un método para agregar una nueva bodega.
* La api debe contar un método para listar todas las bodegas.
* La api debe contar un método para obtener una bodega por su ID.
* La api debe contar un método para editar una bodega por su ID.
* La api debe contar un método para eliminar una bodega por su ID.

Como usuario admin deseo poder crear, eliminar, editar y listar los usuarios.

* 1. **Backend Endpoint Microservicio User’s:**
* La api debe contar un método para crear un nuevo usuario.
* La api debe contar un método para listar todos los usuarios.
* La api debe contar un método para obtener un usuario por su ID.
* La api debe contar un método para editar un usuario por su ID.
* La api debe contar un método para eliminar un usuario por su ID.

Como usuario quiero acceder a usuarios desde la base de datos y gestionar la información de los usuarios de manera eficiente y segura.

* 1. **Backend y BBDD:**
* La base de datos debe contar con una tabla de usuarios en SQL para que pueda gestionar la información.
* Emplear Keycloak para gestionar la autenticación y autorización de usuarios de manera centralizada.
* Visualizar al menos 10 productos por defecto, para que haya información base y pueda funcionar el paginado de cada vista.

**SPRINT 3**

# Historias de Usuario SPRINT 3

Como usuario quiero poder crear una cuenta en la página para poder hacer compras en el sitio.

## 6.1. Registro de Usuario (Cliente):

* El usuario deberá proporcionar información precisa y válida durante el proceso de registro, como: - nombre, - apellido, - imagen:(opcional, sino avatar por defecto)- dirección de correo electrónico, - contraseña, confirmación de contraseña.
* El campo de confirmar contraseña deberá efectivamente validar que sea igual a la ingresada en el campo “contraseña”
* Todos los campos, excepto imagen de perfil, son obligatorios y deben tener validaciones básicas de cantidad mínima de caracteres, tipo de entrada (texto, correo, números, etc.) y mostrar mensajes de error acordes al usuario si este intenta crear su usuario sin rellenar todos los campos obligatorios.
* El usuario deberá ver en la vista un título que le confirme que está viendo el formulario de crear cuenta o registro de nuevo usuario.
* El usuario debe ver un botón al final del formulario “crear cuenta” que le permita crear su cuenta, este a su vez dispara las validaciones y hace la petición al back para confirmar si el correo existe o no inicialmente y si no existe y se crea correctamente deberá ver un mensaje que le indique que su cuenta ha sido creada con éxito y que ahora puede iniciar sesión. Al cerrarlo, deberá dirigirlo a la vista de inicio de sesión o login del sitio.

Como usuario quiero ver las opciones que tengo dentro de mi perfil para saber que funcionales puedo realizar en el sitio.

## 6.2. Desplegable Avatar:

* El usuario no logueado al hacer clic en el avatar debe poder dirigirse a la página en la cual podrá iniciar sesión, crear cuenta y restablecer contraseña
* El usuario logueado debe poder ver un avatar con sus iniciales, al hacer clic en las mismas debe poder ver las opciones del desplegable. Tanto Admin como Cliente pueden ver “Mi perfil”, “Mis compras” y “Cerrar Sesión”. Adicionalmente si es admin Puede ver “Reportes” y “Productos”
* El carrito de compras se puede visualizar solo si la sesión esta iniciada.

Como usuario quiero poder cerrar sesión de manera segura al hacer clic en “cerrar sesión”.

## Cerrar Sesión:

* Permitir al usuario cerrar sesión de manera segura cuando haya terminado de usar su cuenta al hacer clic en el desplegable del avatar en la opción de “cerrar sesión”.
* Asegurarse de que el cierre de sesión sea efectivo y que el usuario no pueda acceder a las funcionalidades disponibles solo para usuarios autenticados después de cerrar la sesión.
* Que se permita al usuario continuar navegando el sitio web como usuario anónimo después de cerrar la sesión.

Como usuario admin quiero que el botón de editar un producto me dirija a un formulario idéntico al de registrar producto.

* 1. **Editar Producto (Admin):**
* El formulario debe estar prellenado con los datos del producto que se editará.
* Debe tener las mismas validaciones que el formulario de registro.
* Si un campo está vacío no debe dejar hacer la actualización.

Como usuario deseo que se me muestren los diseños de cómo está cada página para dar una opinión en cuanto a si se requiere algún cambio:

## Diseño UX/UI:

* Vista de registrar usuario → modificación por keycloak
* Vista de login -> modificación por keycloak
* Vista página de error personalizada
* Vista de carrito -> modificación por requerimientos

Como usuario quiero tener la posibilidad de recuperar mi contraseña:

## Reestablecer contraseña:

* En el formulario de login debería encontrar un botón o enlace con el mensaje “recuperar contraseña”.

Como usuario quiero ver los productos agregados al carrito en un desplegable lateral donde pueda, además, sumar o quitar productos y decidir entre pagar ahora y seguir comprando en el sitio.

## 6.7. Carrito Desplegable:

* Quiero poder cerrarlo en cualquier momento fácilmente, con un botón x o dando clic fuera del mismo.
* Quiero tener la posibilidad de sumar, restar unidades o eliminar el producto añadido con botones de + o -
* Quiero ver listados los productos con imagen, nombre, variedad de uva y precio.
* Quiero la posibilidad de ir directamente al proceso de compra a través de un botón “comprar ahora”
* Quiero poder ver el costo subtotal(individual) y total de lo agregado al carrito.
* Quiero que al cerrar la sesión mi carrito sea vaciado, en caso de que no haya comprado nada.

Como usuario quiero poder comprar de manera sencilla y rápida, con un camino de compra a seguir claro y simple.

## 6.8. Compra en 3 pasos:

* Quiero ver un título que me confirme en la vista que me encuentro ‘comprar ahora’ o ‘proceso de pago’.
* Quiero un botón para poder volver atrás (en la navegación del sitio) en caso de querer seguir comprando. <
* Visualizar un formulario de 3 pasos, para que no sea tan largo uno solo.
* Los formularios deben contar con botones de “siguiente” o “continuar” para avanzar en el proceso y en el caso del paso 2 y 3 con un botón de “regresar” o “atrás” para corregir datos puestos en el/los formularios anteriores, por lo que la data debe persistir.
* En todos los pasos de compra quiero ver el “Resumen de compra” con los conceptos a pagar y sus respectivos valores, así como la suma total de los mismos.
* En el paso 1 de compra quiero ver el o los productos que quiero comprar y su valor tanto individual como total; además quiero tener la opción de eliminar el producto, de sumar unidades.
* En el paso 2 de compra, quiero ver reflejada mi información de envío si previamente hice una compra y poder modificarlo en caso de ser necesario.
* Al finalizar el proceso de compra quiero ver una confirmación de mi compra, con el código único de la compra, o en su defecto un mensaje de me indique que no fue realizada correctamente la compra.
* Finalmente quiero poder visualizar mi compra en la vista de “mis productos”.

Como usuario quiero realizar mis compras con la opción de envió a domicilio.

## 6.9. Registro de Datos de envío:

* El usuario deberá proporcionar información precisa y válida durante el proceso de cargar datos, como: - número documento de identidad, - teléfono, - provincia, - ciudad, - dirección, - datos de envío adicionales que considere necesario como: características del lugar, quien recibirá el pedido, etc.
* Todos los campos son obligatorios y deben tener validaciones básicas de cantidad mínima de caracteres, tipo de entrada (texto, correo, números, etc.) y mostrar mensajes de error acordes al usuario si este intenta crear su usuario sin rellenar todos los campos obligatorios.
* El usuario deberá ver en la vista un título que le confirme que está viendo el formulario de crear cuenta o registro de nuevo usuario.
* El usuario debe ver un botón al final del formulario que le permita agregar su dirección, este a su vez dispara las validaciones y guarda dicha información en la base de datos.
* Una vez cargado el domicilio se mostrará en pantalla los datos cargados, sumado a un botón de confirmar envío.

Como usuario quiero poder visualizar todos los productos que he comprado con anterioridad listados con detalle, como en el resto de la página.

## 6.10. Mis compras:

* Quiero que el historial a manera de lista con paginado
* Quiero poder ordenar mis compras por antigüedad

Como administrador, es fundamental garantizar la seguridad y protección de los datos de los usuarios/clientes en la página web. Para lograrlo, se integrará Keycloak, una solución de gestión de identidad y acceso de código abierto, que proporciona capacidades seguras de autenticación y autorización para aplicaciones web modernas.

## 6.11. Conexión de Front con Keycloak:

* Integrar Keycloak con el frontend es un paso crucial para garantizar la seguridad y autenticación de los usuarios en la página web.
* A la hora que el usuario se registre y/o inicie sesión de una forma segura, se utilizaran los componentes correspondientes de keycloak.

Como administrador, deseo tener roles con diferentes niveles de permisos (Cliente y Admin) para que solo los administradores puedan realizar modificaciones de los productos:

## 6.12. Gestión de Roles:

* El admin lo crea back directamente en la base de datos.
* Permitir actualizar o cambiar los permisos de los roles existentes en cualquier momento.
* Garantizar que los permisos se apliquen de manera efectiva y que los usuarios solo puedan realizar acciones que estén dentro de su nivel de permisos.
* Solo el admin puede crear productos.

Como cliente deseo poder tener todo integrado en la nube para que se refleje una página actualizada e implementada con las mejores tecnologías:

## 6.13. Implementar Infraestructura:

* Crear pipeline del proyecto donde se ejecute la integración continúa (CI) del código de frontend y de backend.
* Desplegar microservicio de users en RailWay.
* Desplegar microservicio de Gateway en RailWay.

Como usuario deseo que la página sea testeada con lo realizado para no tener problemas al salir con la página:

* 1. **Testing:**
* Creación de pruebas automatizadas
* Casos de prueba manuales con:

● Agregar a la planilla de casos de prueba sobre las funcionalidades del sprint III.

● Realizar mantenimiento de los casos de prueba manuales generados en el sprint II, en caso de ser necesario.

Como usuario, necesito poder obtener mis datos cuando inicio sesión.

## 6.15. Crear microservicio de Usuario:

* Crear dentro del repositorio de Backend el microservicio de usuarios que permite la conexión, con el Gateway.
* Gestionar los usuarios por medio de la conexión de Keycloak, parte de historia 13 para continuar con esa continuidad.
* Crear endpoints de consulta de los usuarios para obtener los datos necesarios desde el front.

Como usuario, quiero poder interactuar naturalmente con todas las funcionalidades de la página.

## Crear microservicio de Gateway:

* Crear dentro del repositorio de Backend el microservicio de Gateway que permite la conexión, con los diferentes microservicios dando una conexión segura a cada funcionalidad de la página.

Como usuario, quiero que mi información sea consistente con todo el sitio.

## 6.17. Crear Base de Datos Keycloak:

* Crear la Base de Datos de Keycloak en la cual se estará almacenando la información de los usuarios.
* Conectar la Base de Datos de los users con Keycloak para que se almacene cualquier registro de usuarios por medio de front.

Como usuario, quiero que el sitio sea seguro y que pueda proteger mi información personal.

## Seguridad en la conexión del cliente:

* Crear configuración de OAUTH2 para la identificación de JWT (Tokens) de conexión segura entre los microservicios que se están comunicando.

Como usuario, quiero poder verificar a través de correo electrónico mi registro para asegurar que mi cuenta sea segura.

## Verificación de cuenta de correo:

* Crear conexión por medio de conexión SMTP para la configuración del servidor de correo con Keycloak y permitir que exista un correo de verificación de cuenta por parte del usuario.

Como usuario, quiero poder interactuar naturalmente con todas las funcionalidades de la página.

## Conexión de Gateway con los demás microservicios:

* Gestionar las conexiones requeridas por cada petición por medio del servicio configurado como Gateway para dar una mayor seguridad y no tener todo publicado en la parte principal dando diferentes rutas o conexiones en diferentes puertos de acceso diferente al 8080.

Como usuario, quiero poder interactuar desde la página del sitio y tener todas las funcionalidades a mi disposición para poder verificar funcionalidades desde el perfil en el cual me encuentre:

## 6.21. Modificar las configuraciones de Gateway para que se pueda interactuar desde front:

* Configurar toda la parte entre el Keycloak y el Front para que por medio del Gateway exista la comunicación intermedia y se puedan comunicar sin problema consumiendo desde la parte del front por medio de los endpoints que fueron creados desde Usuario.

## 6.22. Despliegue de réplicas de los microservicios como Backus ante fallos:

* Realización de nuevos repositorios y nuevos despliegues en Railway para los siguientes microservicios: ms-commerce, ms-users, ms-eureka, ms-gateway.

## 6.23. Implementación de configuración centralizada:

* Implementación de Config Server como metodología para aislar entornos y proteger los datos de acceso a los microservicios.