
Especificación de requisitos de software

Para

**<e-commerce company: cosmo
infinite >**

Versión 1.0 no aprobado

Preparado por <Kathy luz otero Pérez

Ellen Nicole Ordoñez Chávez

Jesús David Julio Hawasly>

<Universidad tecnológica de Bolívar>

<5 de junio de 2025>

Tabla de contenido

1. Introducción.....	1
2.	
Objetivos.....	1
3. Descripción general	2
3.1 Características del producto	2
3.2 Clases de usuarios y características.....	2
3.3 Limitaciones de diseño y aplicación	2
3.4 Documentación del Usuario	2
3.5 Supuestos y dependencias	3
4. Características del sistema	4
4.1 Características de Sistema.....	4
4.1.1 Descripción y prioridad.....	4
4.1.2 Secuencias de estímulo/respuesta.....	4
4.1.3 Requisitos funcionales.....	4
5. Requisitos de la interfaz.....	5
5.1 Interfaces de Usuario.....	5
5.2 Interfaces de hardware	5
5.3 Interfaces de software.....	5
5.4 Interfaces de Comunicación	5
6. Otros requisitos No funcionales	6
6.1 Requerimientos de desempeño.....	6
6.2 Requisitos de seguridad.....	6
6.3 Atributos de calidad del software.....	6
7. Otros Requerimientos.....	7
8. Prueba de interes	8
9. Modelo de analisis	9
10. lista de problemas	10
11. conclusión	11
12. glosario	12
13. referencias	13

Historial de revisiones

Nombre	Fecha	Motivo de los cambios	Versión
MVP	17/03/2025	Establecimiento de la introducción, objetivos y MVP	0.01
Integración del sprint 1	27/03/2025	La integración de nuestro primer esprint	0.02
Integración de nuestro segundo sprint	30/03/2025	La integración de nuestro segundo sprint y además de eso la edición de nuestra pasarela de pagos por una interfaz que muestre un mensaje y el cambio del login	0.03
Integración de nuestro 3 sprint además de la creación de nuestros microservicios e implementación de los servicios CRUR	20/04/2025	Este esprint fue el mas largo y de mas plazo de entrega se hicieron nuestros microservicios y uno se implementó con el servicio CRUD	0.04
Integración del sprint 4 y 5	03/06/2025	Culminación del desarrollo de los microservicios, verificación funcional a través de pruebas unitarias, y logro de una integración efectiva entre el frontend y el backend. Se completó la documentación de las APIs y se consolidó la versión funcional del MVP.	0.05

Tabla 1. Historial

1. Introducción

Este documento presenta la Especificación de Requisitos de Software (ERS) para el sistema e-commerce Cosmo Infinite. Es una tienda en línea especializada en ropa de playa, productos artesanales y alimentos tradicionales colombianos. Nuestro objetivo fue construir una plataforma funcional, escalable y segura que ofrezca una experiencia de compra fluida y culturalmente enriquecedora.

El desarrollo se realizó por sprints utilizando una arquitectura de microservicios implementada con FastAPI, integrando tecnologías como Firebase para autenticación, Stripe (modo simulado) para pagos y Algolia para búsquedas avanzadas. Este documento detalla los requisitos funcionales y no funcionales del sistema, así como las interfaces, limitaciones, usuarios y tecnologías utilizadas durante los sprints.

2. Objetivos

Objetivo general:

Elaborar una plataforma de comercio electrónico (e-commerce) llamada “*Cosmo Infinite*” que garantice una interacción estable, ágil, adaptable al crecimiento del sistema, y culturalmente enriquecida, a través de la integración de tecnologías modernas, una arquitectura de microservicios, y funcionalidades que aumenten la eficiencia de la compra de ropa de playa, artesanías y alimentos de origen colombiano.

Objetivos específicos:

- Desarrollar e implementar una plataforma modular basada en microservicios para una gestión eficiente.
- Implementar una interfaz segura para usuarios y clientes.
- Adaptar autenticación con Firebase, aplicando control de acceso basado en roles “cliente-administrador”.
- Aplicar un control de búsqueda inteligente con filtros personalizados a través de algolia.
- Establecer una gestión efectiva de productos, pedidos, devoluciones y usuario desde un panel administrativo.
- Documentar y estructurar correctamente las funcionalidades, arquitecturas y APIs.
- Realizar el traslado a una base de datos SQL.
- Impulsar la identidad cultural colombiana mediante los productos brindados.

3. Descripción general

Cosmo Infinite es una plataforma de comercio electrónico desarrollada para brindar a los usuarios una experiencia completa y segura en la compra de ropa de playa, artesanías, alimentos típicos colombianos y accesorios. Basada en una arquitectura de microservicios y tecnologías modernas, la plataforma garantiza escalabilidad, eficiencia y una navegación intuitiva.

El sistema permite explorar un catálogo organizado que incluye vestidos de baño para hombres y mujeres, guayaberas, bermudas, sandalias, alpargatas y complementos de moda. Además, presenta una colección única de artesanías que abarca figuras de madera y cerámica, joyería en plata y baño de oro, imanes, llaveros, esmeraldas, y recuerdos típicos. También ofrece una sección dedicada a productos tradicionales como café colombiano, chocolates artesanales y dulces regionales.

La experiencia del usuario está optimizada mediante autenticación segura, panel de usuario personalizado, historial de pedidos, sistema de devoluciones, recomendaciones inteligentes basadas en historial de navegación, y un motor de búsqueda avanzado potenciado por Algolia.

Toda la información es gestionada mediante servicios independientes desarrollados en FastAPI, conectados a una futura base de datos SQL. Las credenciales de los usuarios están protegidas a través de Firebase, y el sistema cuenta con pruebas unitarias para garantizar fiabilidad. Adicionalmente, el sitio está preparado para futuras expansiones, soportando la integración con terceros y nuevas funcionalidades sin afectar su núcleo.

3.1. Características del producto

El sistema de e-commerce de Cosmo Infinite incluye una gama amplia de funcionalidades desarrolladas a lo largo de los sprints, orientadas a la eficiencia operativa y la experiencia del usuario. Entre ellas destacan:

- **Catálogo de productos con filtros avanzados:** Visualización clara con imágenes, descripción, categorías y buscador inteligente.
- **Gestión de productos e inventario:** CRUD completo para administradores, control de stock en tiempo real y alertas de bajo inventario.
- **Carrito de compras dinámico:** Permite agregar, modificar y eliminar productos, mostrando totales actualizados y resúmenes detallados.
- **Procesamiento de pedidos:** Confirmación automática, seguimiento por estado (pendiente, en proceso, entregado), y sistema de notificaciones por correo.
- **Gestión de devoluciones:** Solicitudes por parte del cliente y revisión por parte del administrador con control de estado.

- **Autenticación y gestión de usuarios:** Registro, inicio de sesión y recuperación de cuenta mediante Firebase; asignación de roles cliente/administrador.
- **Pasarela de pago (simulada):** Simulación de pagos mediante Stripe, con validación de datos básicos y generación de comprobantes.
- **Historial de compras y perfil de usuario:** Visualización de compras anteriores, edición de perfil, y almacenamiento de direcciones.
- **Panel administrativo:** Módulo exclusivo para administrar productos, pedidos, usuarios, y visualizar reportes de ventas e inventario.
- **Recomendaciones personalizadas:** Uso del historial del cliente para sugerencias relevantes, conectado a búsquedas mediante Algolia.
- **Notificaciones en tiempo real:** Vía correo electrónico o mensaje sobre estado de pedidos, promociones y cambios importantes.
- **Arquitectura modular con microservicios:** Servicios como `product_service`, `user_service`, `order_service`, `cart_service` y `return_service` completamente separados y escalables.
- **Pruebas unitarias e integración:** Validación de funcionalidades clave con cobertura esperada superior al 80%.
- **Gestión de productos por categoría:** Los productos están organizados en ropa, artesanías, café y más, facilitando la navegación y segmentación.
- **Visualización del catálogo con búsqueda avanzada:** Se incorpora Algolia para realizar búsquedas inteligentes y rápidas con filtros personalizados.

3.2 Clases de usuarios y características

Clientes

- Registro e inicio de sesión
- Agregar productos al carrito
- Realizar pedidos
- Consultar historial y estado de pedidos
- Editar perfil

Administradores

- Gestión de productos y stock
- Visualización y gestión de pedidos y devoluciones

- Control de usuarios
- Generación de reportes
- Gestión de categorías

3.3 Limitaciones de diseño y aplicación

- El sistema no procesa pagos reales, solo simula la operación.
- Autenticación delegada a Firebase.
- Los servicios aún están migrando de almacenamiento local a base de datos SQL.
- Algunos componentes aún están en fase de pruebas unitarias.

3.4 Documentación del Usuario

- Manual del Cliente
- Manual del Administrador
- Guías de API REST (Swagger/OpenAPI)
- Ayuda integrada y FAQ

3.5 Supuestos y dependencias

- Plataforma web basada en FastAPI
- Firebase para autenticación
- Stripe (simulado) para pagos
- Algolia para búsqueda
- Cloudflare para seguridad
- GitHub para versionamiento
- Base de datos relacional en transición a SQL

4. Características del sistema

4.1 Características del Sistema

4.1.1 Descripción y prioridad

Funcionalidad	Prioridad
Gestión de productos e inventario	Alta
Carrito de compras	Alta
Autenticación y gestión de usuarios	Alta
Pedidos y seguimiento	Alta
Reembolsos y devoluciones	Media
Panel de administración	Alta
Búsqueda avanzada y filtros	Alta
Sistema de notificaciones	Media

Tabla 2. Características del sistema

4.1.2 Secuencias de respuesta

- **Producto:** al crear un producto, se almacena en la base de datos y se visualiza en el catálogo.
- **Carrito:** el usuario agrega/elimina productos y el sistema actualiza el contenido dinámicamente.
- **Pedido:** se genera un ID único, se registra y notifica al usuario.
- **Devolución:** el cliente solicita una devolución y el administrador revisa y responde.
- **Inicio de sesión:** Firebase valida credenciales y determina el rol.

4.1.3 Requisitos funcionales

ID	Descripción
RF1	Descomposición en microservicios
RF2	Implementación y exposición de APIs REST
RF3	Gestión de productos (CRUD)
RF4	Gestión de usuarios (registro, roles, autenticación)
RF5	Manejo de carrito de compras
RF6	Gestión de pedidos

RF7	Sistema de notificaciones
RF8	Personalización basada en historial del cliente
RF9	Sistema de devoluciones y reembolsos
RF10	Validación de datos de entrada y manejo de errores

Tabla 3. Requisitos funcionales

5. Requisitos de la interfaz

5.1 Interfaces de Usuario

- Catálogo general con filtros por categoría
- Detalles de producto
- Panel de usuario (perfil, historial)
- Carrito y proceso de pago
- Panel de administración
- Flujo de compra paso a paso

5.2 Interfaces de hardware

- Dispositivos compatibles: PC, laptops, tablets, smartphones
- Requiere conexión a Internet y navegador actualizado

5.3 Interfaces de software

- FastAPI (Backend)
- Firebase (auth)
- Stripe (pagos simulados)
- Algolia (búsqueda)
- SQL (migración de base de datos)
- GitHub (CI/CD)

5.4 Interfaces de Comunicación

- HTTP/HTTPS
- APIs RESTful
- Correo electrónico para notificaciones
- Pruebas vía integración GitHub Actions

6. Otros requisitos No funcionales

6.1 Requerimientos de desempeño

- Tiempo de respuesta: menor a 2 segundos
- Soporte para múltiples usuarios concurrentes
- Arquitectura escalable

6.2 Requisitos de seguridad

- Cifrado HTTPS (SSL)
- Firebase Authentication + JWT
- Validación de datos y control de acceso
- Protección contra inyecciones y CSRF


6.3 Atributos de calidad del software

- **Usabilidad:** interfaz responsive, mobile-first
- **Mantenibilidad:** código modular y documentado
- **Disponibilidad:** uptime mayor a 99%
- **Escalabilidad:** microservicios
- **Tolerancia a fallos:** mecanismos de recuperación

7. Otros Requerimientos

- Internacionalización: soporte multilenguaje
- Cumplimiento con Habeas Data
- Documentación técnica para APIs
- Pruebas unitarias y de integración con cobertura mínima del 80% de coverage

8. Pruebas de estrés




Host
https://super-duper-space-waddle-7v5gxq7jwwr7f9xg...

Status
READY

RPS
0

Failures
0%



Start new load test

Number of users (peak concurrency) *

1

Ramp up (users started/second) *

1


Host

https://super-duper-space-waddle-7v5gxq7jwwr7f9xg-8089.app.github.dev/

Advanced options

START

ABOUT



Host
https://super-duper-space-waddle-7v5gxq7jwwr7f9xg...


Status
STOPPED

RPS
32.7

Failures
0%

NEW

RESET



STATISTICS

CHARTS

FAILURES

EXCEPTIONS

CURRENT RATIO

DOWNLOAD DATA

LOGS

LOCUST CLOUD

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	//productos/	787	0	45	100	160	52.75	37	320	5328	32.7	0
	Aggregated	787	0	45	100	160	52.75	37	320	5328	32.7	0

Type	Name	50%	66%	75%	80%	90%	95%	98%	99%	99.9%	99.99%
%	100% # reqs										
GET	/productos/	5	6	7	9	14	18	23	33	110	11
0	110 1711										
	Aggregated	5	6	7	9	14	18	23	33	110	11
0	110 1711										

Error report	Error
# occurrences	
1711	GET /productos/: gaierror(-2, 'Name or service not known')

9. Modelo de análisis

Problema:

Visualizar los productos del carrito

Contexto:

Este caso de uso permite al usuario ver los productos que ha añadido a su carrito de compras, incluyendo nombre, imagen, cantidad, precio unitario, subtotal y la opción de modificar o eliminar productos antes de proceder al pago.

Actores principales:

- Usuario registrado
- Usuario no registrado (carrito temporal)

Flujo principal:

1. El usuario accede a la sección “Carrito” desde el menú o el ícono correspondiente.
2. El sistema consulta los productos almacenados en el carrito asociado al usuario.
3. Se muestra una lista con:

- Nombre del producto
- Imagen del producto
- Precio unitario
- Cantidad seleccionada
- Subtotal por producto

4. El sistema calcula y muestra el total del pedido.

5. El usuario puede:

- Aumentar o disminuir la cantidad
- Eliminar productos
- Proceder al checkout

Requisitos relacionados:

- **RF-06:** Visualización de productos en el carrito
- **RF-07:** Actualización de cantidades
- **RF-08:** Eliminación de productos del carrito
- **RNF-03:** Tiempo de carga inferior a 2 segundos para mostrar el carrito

Resultado:

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	//carritos/1	606	0	45	97	220	54.39	39	489	5328	5.2	0
DELETE	//carritos/1/productos/1	606	606	15	24	140	18.63	12	341	0	6	6
PUT	//carritos/1/productos/1	632	632	15	24	64	18.16	12	350	0	7.6	7.6
POST	//ordenes/	604	604	15	24	270	20.36	12	393	0	7.3	7.3
GET	//productos/	639	0	45	95	200	53	36	336	5328	6.7	0
Aggregated		3087	1842	17	71	190	33	12	489	2148.8	32.8	20.9

10. Lista de problemas

ID	descripción del problema	Estado	Observaciones
P-01	La integración con el sistema de pagos está en modo simulado	No resuelto	Aún no se ha conectado con entorno real de pagos
P-02	Algunos módulos migran parcialmente a SQL	En desarrollo	Es necesario corregir la conexión y las consultas
P-03	La funcionalidad de búsqueda con filtros todavía está en proceso de mejora	En desarrollo	Algolia requiere más pruebas con categorías personalizadas
P-04	No se ha definido el proceso completo de devolución	Pendiente	Aún no se conecta con historial de pedidos ni almacén
P-05	El manejo de errores es irregular en algunas APIs	No resuelto	Necesario estandarizar respuestas HTTP y validaciones

Tabla 4. Lista de problema

11. Conclusion

Cosmo infinite es un proyecto de comercio electrónico desarrollado con enfoque estratégico y arquitectura contemporánea. A lo largo de la elaboración de los sprints se construyó un sistema confiable basado en microservicios, que permiten la facilidad de la escalabilidad, mantenimiento y futuras integraciones.

Incorpora funcionalidades para administrar productos, usuarios, pedidos, recomendaciones y devoluciones, enfocando en una experiencia de usuario segura y eficiente. Para ello, se utilizaron tecnologías como FastAPI, Firebase, Algolia y Stripe (modo simulado), asegurando una integración efectiva entre frontend y backend.

El proyecto está en una etapa avanzada, con microservicios y pruebas funcionales realizadas. Algunas partes del sistema se están adaptando para funcionar con SQL, pero está preparado para evolucionar a un producto comercial con pocos ajustes. El proyecto se encuentra en etapa estable, con bases sólidas para futuras mejoras

13. Glosario

TERMINO	CONCEPTO
Microservicios	Son una arquitectura de desarrollo de software en la que una aplicación se estructura como un conjunto de servicios pequeños y autónomos que se comunican entre sí.
Firebase	Es una plataforma en la nube desarrollada por Google que proporciona una serie de herramientas diseñadas para ayudar los desarrolladores en la creación, optimización y monetización de aplicaciones web y móviles.
Stripe	Es una plataforma de pagos en línea que se incorpora en una página web y que permite a las empresas y autónomos gestionar los pagos por productos y servicios a través de un sitio web.
Algolia	Es un motor de búsquedas escalable que se integra en un producto, aplicación o sitio web, es accesible a medios, ecommerce y negocios SaaS, quienes podrán acceder a él desde la API.
FastAPI	Es un conjunto de herramientas para los programadores que quieren construir aplicaciones web de manera eficiente y rápida.
CRUD	Es un acrónimo que representa las cuatro operaciones fundamentales que se realizan sobre los datos: Crear, Leer, Actualizar y Eliminar. Estas operaciones permiten a los desarrolladores interactuar con datos persistentes almacenados en bases de datos u otros sistemas de almacenamiento de datos.
API REST	Es una interfaz de programación de aplicaciones (API) que sigue los principios de diseño del estilo de la arquitectura REST. REST significa transferencia de estado representacional y consiste en un conjunto de reglas y recomendaciones para diseñar una API web.
SQL	Es un lenguaje de programación para almacenar y procesar información en una base de datos relacional. Una base de

	datos relacional almacena información en forma de tabla, con filas y columnas que representan diferentes atributos de datos y las diversas relaciones entre los valores de datos.
JWT	Es un objeto de Java script creado con el propósito de transmitir información de forma segura y efectiva entre un determinado emisor y su respectivo receptor.
Responsividad	Es la capacidad que tiene la interfaz de una página web para adaptarse a las configuraciones de distintos dispositivos.
Frontend	Es la parte de una web que conecta e interactúa con los usuarios que la visitan. Es la parte visible, la que muestra el diseño, los contenidos y la que permite a los visitantes navegar por las diferentes páginas mientras lo deseen.
Backend	Es la parte trasera de cualquier página web. Se trata de todo el conjunto del desarrollo que se encarga de que una página funcione y de que lo haga como lo hace, pero que al mismo tiempo es totalmente invisible para el usuario, que solo ve lo visual y gráfico.
MVP (Producto Mínimo Viable)	Es un prototipo del producto o servicio que quieres lanzar, con unas funciones mínimas que te permitan presentarlo a tus clientes potenciales y comprobar su interés por el mismo.

Tabla 5. Glosario

12. Referencias

- [1] Otero Pérez, K. L., Ordoñez Chávez, E. N., & Julio Hawasly, J. D. (2025). Sprint 1: E-commerce Company: Cosmo Infinite [Manuscrito no publicado]. Universidad Tecnológica de Bolívar.
- [2] Otero Pérez, K. L., Ordoñez Chávez, E. N., & Julio Hawasly, J. D. (2025). Sprint 2: E-commerce Company: Cosmo Infinite [Manuscrito no publicado]. Universidad Tecnológica de Bolívar.
- [3] Otero Pérez, K. L., Ordoñez Chávez, E. N., & Julio Hawasly, J. D. (2025). Sprint 3: E-commerce Company: Cosmo Infinite [Manuscrito no publicado]. Universidad Tecnológica de Bolívar.
- [4] Otero Pérez, K. L., Ordoñez Chávez, E. N., & Julio Hawasly, J. D. (2025). Sprint 4: E-commerce Company: Cosmo Infinite [Manuscrito no publicado]. Universidad Tecnológica de Bolívar.
- [5] Otero Pérez, K. L., Ordoñez Chávez, E. N., & Julio Hawasly, J. D. (2025). Sprint 5: E-commerce Company: Cosmo Infinite [Manuscrito no publicado]. Universidad Tecnológica de Bolívar.
- [6] *Algolia para eCommerce: Qué es y cuáles son sus ventajas.* (s/f). Gluo. Recuperado el 5 de junio de 2025, de <https://www.gluo.mx/blog/algolia-para-ecommerce-que-es-y-cuales-son-sus-ventajas>
- [7] Carrasco, C. (2020a, agosto 27). *¿Qué es Back end y para que sirve?* - Neo Wiki. Neoattack.com; NeoAttack. <https://neoattack.com/neowiki/back-end/>
- Carrasco, C. (2020b, diciembre 31). *Front end.* Neoattack.com; NeoAttack. <https://neoattack.com/neowiki/front-end/>
- [8] Castillo, G. (2024, julio 17). *JWT: Qué es, estructura, usos e implementaciones.* Innovación Digital 360. <https://www.innovaciondigital360.com/blockchain/jwt-que-es-estructura-usos-e-implementaciones/>
- [9] *CRUD: Qué es, Para Qué sirve, Cómo Funciona y Ejemplos.* (2023, diciembre 22). EBIS Education. <https://www.ebiseducation.com/crud-que-es-para-que-sirve-como-funciona-y-ejemplos>

- [10] FastAPI: la herramienta definitiva para el desarrollo web. (2023, septiembre 25). *OpenSistemas*. <https://opensistemas.com/fastapi-la-herramienta-para-el-desarrollo-web/>
- [11] Isaac. (2025, febrero 26). *Firebase: Qué es y cómo funciona la plataforma de Google*. Mundobytes. <https://mundobytes.com/firebase-que-es/>
- [12] Nehuén, T. (2024, junio 28). *Microservicios*. Definición.de; Definicion.de. <https://definicion.de/microservicios/>
- [13] *¿Qué es una API de REST?* (s/f). Redhat.com. Recuperado el 5 de junio de 2025, de <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- [14] Quipu, R. D.-C. (2023, junio 7). *Stripe: qué es y cómo funciona esta pasarela de pago*. Autónomos, empresas y asesorías; Quipu. <https://getquipu.com/blog/que-es-y-como-funciona-stripe/>
- [15] Responsividad web: diseñando para una sociedad de múltiples pantallas. (2023, junio 13). *Empresa de Marketing de contenidos* | WeAreContent. <https://www.wearecontent.com/blog/seo/responsividad>
- [16] (S/f-a). Amazon.com. Recuperado el 5 de junio de 2025, de <https://aws.amazon.com/es/what-is/sql/>
- [17] (S/f-b). Eserp.com. Recuperado el 5 de junio de 2025, de <https://es.eserp.com/articulos/producto-minimo-viable/>