

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO — CHILE



**DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA
MARKETPLACE PARA EMPRENDEDORES DE PRODUCTOS
ELABORADOS:
ENFOQUE EN ARQUITECTURA DE SOFTWARE, ENTORNO DE
DISEÑO Y BACKEND**

AQUILES ALONSO VIZA VERGARA

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN TELEMÁTICA

PROFESOR GUÍA: JOSÉ MANUEL MARTÍNEZ VERDUGO
PROFESOR CORREFERENTE: NICOLÁS ALONSO JARA CARVALLO

25 de septiembre de 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☐ Tesis de Postgrado;

Título del trabajo: Diseño e implementación de una plataforma marketplace para emprendedores de productos elaborados: Enfoque en arquitectura de software, entorno de diseño y backend

Nombre del candidato(a): Aquiles Alonso Viza Vergara

Carrera / Grado: Ingeniería Civil Telemática

Campus: Casa Central Valparaíso ; **Departamento:** Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, José Manuel Martínez, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

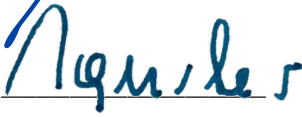
Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 02/09 ; **Firma:** 

Estudiante o Candidato(a):

Fecha: 27/08 ; **Firma:** 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradecimientos

Culminar la escritura de esta memoria ha resultado ser un proceso extenso, acompañado de forma concurrente por varios proyectos en donde tomé responsabilidad y compromiso, buscando una realización personal que no encontré durante mi época universitaria. Esto conllevó a la extensión del proceso académico, lo que tuvo consecuencias indeseadas en muchos ámbitos. Sin embargo, puedo decir que me siento realizado, totalmente satisfecho por haber vivido, compartido y trabajado con tanta gente maravillosa.

Por sobre todas las personas, a quienes más agradezco es a mis padres. El pilar sobre el que me construí como persona, recibiendo constantemente los frutos de su trabajo y amor, brindándome cada una de las herramientas que he necesitado para desenvolverme en el mundo que he recorrido. Gracias por tantos viajes por los paisajes del interior de Arica y las vacaciones en la casa de los abuelos. Por el amor inculcado hacia la naturaleza, la música y a los perritos. Estoy muy agradecido con ambos por el constante esfuerzo que han realizado conmigo y mis hermanas y hermanos. Ahora que podré trabajar como ingeniero no me caben las ganas de retribuir a la familia el apoyo incondicional que me han entregado. Gracias por darme una familia tan bonita, un Mateo cariñoso y tierno, una Amayita chistosa de carácter fuerte, un Marcelo gigante y un trío de perros papiche que son muy bonitos cuando no están ladrando a cada cosa que se mueva.

A mi tata Gastón y Abuelita Marta, que abrieron su casa a mi persona cuando comencé los estudios universitarios y se preocuparon de que no faltara comida ni abrigo en Llay-Llay, donde siempre hubo cariño cada fin de semana que estuvimos juntos. Tatita, espero que además de los genes de flaco, también me herede la lucidez con que usted vive cada día de su cuarta infancia. Gracias a Bandida, por acompañar al abuelo en esa casa tan grande durante estos últimos años.

Querida Inger, conocerte es uno de los motivos que da sentido a toda esta travesía. Agradezco el tiempo que me has dado y la paciencia que tienes con la excesiva tranquilidad con que me desenvuelvo. Gracias por enseñarme acerca del puerto y acompañarme a tantas aventuras. Por ayudarme a ser mejor persona y definitivamente agradezco la insistencia que tuviste para que terminara esta memoria. Espero ser el apoyo que tú necesitas, así como tú me apoyas a mí.

Gracias Erick y Pablo por acompañarme durante gran parte de este periodo. Por invitarme a vivir con ustedes pese a no ser tan cercanos en un inicio y haber convivido todos esos años con esas tardes de estudio y monos chinos. Por tantas juntas con los cabros, celebraciones de todo tipo y las discusiones innecesarias como estimar si llegábamos a tiempo a la universidad si tomábamos una u otra micro. Asimismo, agradezco a Lili y Erick por ofrecerme un nuevo espacio donde estar luego de que acabó el tiempo en el departamento. Gracias a ustedes y su invitación, muchas de las cosas que logré durante este tiempo fueron posibles.

Agradezco a todas las personas con quien pude compartir durante la carrera, especialmente al grupo de amigos con el que compartimos en el *d.epa* tantas tardes y noches. Cada sesión intensiva de estudio y después la celebración cuando acababa el periodo de pruebas. Cuando venían los telemáticos de Santiago u ocurría algún

cumpleaños. Atesoro cada una de esas juntas que terminaban con todos apretados, tratando de dormir en la diminuta pieza que me correspondía.

Agradezco al profesor José Manuel Martínez por darme consejos “más de amigo que de profesor”. Por su eterna paciencia ante las eventualidades de este proyecto. Y por motivarme a mantener mi interés en la senda del bajo nivel cada vez que me resigné y volví al software. Es gracias a la charla que tuvimos durante la primera reunión del PMM lo que me encaminó en la serie de eventos que han ocurrido en estos años.

Gracias a Alonso Rodríguez y Gonzalo Carrasco, por confiar en mis capacidades y permitirme trabajar en el laboratorio, experimentando de primera mano lo que era trabajar como ingeniero. Ustedes definen el estándar de profesional al que apunto y espero llegar a ese nivel.

Gracias a Jorge Marín por darnos a varios la posibilidad de adentrarnos en el nicho de la microelectrónica, por enseñarme con paciencia sobre los conceptos y técnicas y por permitirme ocupar un puesto donde aportar a esta comunidad. Gracias a Christian Rojas por entregar al área una perspectiva técnica distinta, por apoyar continuamente a los estudiantes que toman esta senda y por tener tan buen gusto musical. Agradezco a todos quienes han aportado en la promoción de la microelectrónica en Chile y se han organizado para generar eventos, talleres y charlas. Entre ellos, a Alfonso Cortés, Seba Sánchez, Dani, Mario, A ambos Vicente, A Fran, Nelson, Maxi y todo aquel que ha dado vida a la iniciativa ChipUSM dedicando su tiempo en la misión de difundir y acercar a la gente esta área del conocimiento.

Muchas gracias a quienes han enriquecido el panorama porteño y han levantado espacios culturales, dándole otra faceta más a esta ciudad. Agradezco la entrega y compromiso increíble que ha tenido Sebastián García con “Sesiones 21” y Lourdes Montenegro con “Slam Valparaíso”, lugares que han sido foco de música y poesía emanada de la misma gente de este hermoso puerto.

Agradezco a mis amigos del Conjunto Folklórico por recibirme con los brazos abiertos y darme la posibilidad de conocer tanta música de esta larga y angosta faja de tierra. Gracias a los integrantes con quienes compartí múltiples ensayos, presentaciones y salidas al Liberty y especialmente a Víctor Riquelme y Jimi Molina por su entrega a la hora de representar comunidades y culturas de nuestro país para acercarlo a la gente de esta región. Y no puedo estar más agradecido de conocer a Diego Cárdenas, por tu infinita motivación y esmero hacia la cultura andina que existe en Valparaíso, por difundir y registrar estos eventos y por la sinceridad con la que te desenvuelves día a día y el compromiso a tus proyectos.

Durante gran parte del periodo universitario me enfoqué únicamente en los estudios, en conocer herramientas y técnicas avanzadas del rubro y recolectar conocimientos como curiosidades que nunca pude ocupar ni compartir. En este último tiempo me abrí a otras vivencias, exploré lugares de música, cultura, poesía y me puse a disposición para ser un aporte en aquellos espacios donde me sentí integrado. Si no fuera por todas las personas que me acompañaron durante este periodo, aún estaría inquieto.

Gracias totales.

Resumen

Si un emprendedor desea digitalizar su negocio, existe una amplia gama de opciones a las que puede optar. Sin embargo, las soluciones existentes presentan varios impedimentos que entorpecen su transformación digital, como los altos costos de entrada, las comisiones elevadas, la escasa promoción y la pérdida de identidad de marca. En esta memoria se propone el desarrollo de un marketplace que aborda estas dificultades, proporcionando una solución efectiva y centrada en las necesidades de los emprendedores. El trabajo hace hincapié en la arquitectura del software, el desarrollo del backend y la caracterización de las necesidades del usuario, en el contexto del mercado nacional MIPYME de productos elaborados.

Este trabajo comienza analizando el contexto, las necesidades y los desafíos específicos que enfrentan los emprendedores en el mercado digital chileno. Luego, se revisa el estado del arte de generadores web, e-commerce y marketplaces para recopilar las características principales de estas soluciones y comprender de qué manera resuelven los problemas de sus usuarios objetivo. Con esta base, se lleva a cabo el diseño de la solución, que incluye la determinación de requisitos funcionales, el análisis de tecnologías y el establecimiento de la arquitectura de la plataforma.

Una vez finalizado el diseño, se implementan los requisitos del backend de manera estructurada, lo que incluye la generación de diagramas de flujo y de secuencia entre los componentes lógicos, junto con la descripción de limitaciones técnicas. Posteriormente, se realiza una demostración completa que recorre el “Happy Path” del usuario, mostrando que el producto mínimo viable cumple con los requisitos principales de usabilidad. Además, a través de retroalimentación obtenida en encuestas y entrevistas, se busca comprender mejor las expectativas de los usuarios y su perspectiva respecto al producto. Con esta información, se realiza un análisis general del trabajo, se establecen las conclusiones y se propone el trabajo futuro para las siguientes iteraciones de desarrollo.

En resumen, esta memoria investiga los problemas que enfrentan los emprendedores al adentrarse en el mundo digital, analiza el estado del arte y plantea una hipótesis de solución. Luego, se establecen los objetivos, se diseñan los diagramas de solución, arquitectura y componentes, y se implementa el sistema cumpliendo los requisitos. La solución se valida mediante encuestas y entrevistas, se presentan conclusiones generales y se sugiere trabajo futuro.

Palabras clave: Marketplace, MIPYME, PYME, E-Commerce, Backend

Abstract

If an entrepreneur wishes to digitize their business, there is a wide range of options they can choose from. However, existing solutions come with difficulties that hinder their digital transformation, such as high entry costs, elevated commissions, limited promotion, and the loss of brand identity. Given this scenario, this thesis proposes the development of a marketplace that addresses these difficulties by providing an effective solution focused on the needs of entrepreneurs. The work emphasizes software architecture, backend development, and the characterization of user needs within the context of the Chilean MSME market for original products made by entrepreneurs.

This work begins by analyzing the context, needs, and specific challenges entrepreneurs face in the Chilean digital market. Then, it reviews the state of the art of web builders, e-commerce platforms, and marketplaces to gather the main characteristics of these solutions and understand how they solve their target users' problems. Based on this, the solution design is carried out, which includes determining functional requirements, analyzing technologies, and establishing the platform's architecture.

Once the design is finalized, the backend requirements are implemented in a structured manner. This includes generating flowcharts and sequence diagrams between the logical components, along with a description of technical limitations. Subsequently, a complete demonstration is performed by walking through the user's Happy Path, showing that the minimum viable product meets the core usability requirements. Furthermore, through feedback obtained from surveys and interviews, the goal is to better understand user expectations and their perspective on the product. With this information, a general analysis of the work is conducted, conclusions are drawn, and future work for the next development iterations is proposed.

In summary, this thesis investigates the problems entrepreneurs face when entering the digital world, analyzes the state of the art, and proposes a solution hypothesis. Then, the objectives are established, the solution, architecture, and component diagrams are designed, and the system is implemented to meet the requirements. The solution is validated through surveys and interviews, general conclusions are presented, and future work is suggested.

Keywords: Marketplace, MIPYME, PYME, E-Commerce, Backend

Índice general

1	Introducción	11
1.1	Primeras incursiones del mercado en Internet	11
1.2	Web 2.0 y las redes sociales	11
1.3	Definición del problema	12
1.3.1	Caracterización del mercado chileno	12
1.3.2	MIPYMES y el mercado online nacional	13
1.3.3	Retos de la incursión en el mercado online	14
1.3.4	Hipótesis	14
1.3.5	Objetivos	15
1.3.6	Estructura de la memoria	15
2	Estado del Arte	17
2.1	Creadores de sitios web	18
2.2	Creadores de tiendas	18
2.3	Ventas en redes sociales	19
2.4	Servicios de consultoría y apoyo	20
2.5	Servicios mixtos	21
2.5.1	Marketplaces nacionales	23
2.6	Ventajas y desventajas de las alternativas	24
3	Diseño de la Solución	27
3.1	Análisis	27
3.1.1	Alternativas de solución	27
3.1.2	Acercamiento a la solución	28
3.2	Diseño	29
3.2.1	Requisitos del sistema	29

3.2.2	Diagrama de contexto	31
3.2.3	Diagrama de arquitectura	32
3.2.4	Descripción de módulos	33
3.2.5	Diagrama de componentes	34
3.2.6	Descripción de componentes	35
3.2.7	Análisis de tecnologías	35
4	Implementación de la Solución	43
4.1	Diseño de interfaz web del backend	43
4.1.1	Esquema de seguridad	43
4.1.2	Manejo de versiones	50
4.1.3	Descripción de API HTTP	51
4.2	Desarrollo por requisitos	52
4.2.1	Requisito Funcional 1: Registro de usuarios	53
4.2.2	Requisito Funcional 2: Recuperación de contraseña	61
4.2.3	Requisito Funcional 3: Autenticación por terceros	62
4.2.4	Requisito Funcional 9: Gestión del producto	64
4.2.5	Requisito Funcional 10: Estilo de la tienda	68
4.2.6	Requisito Funcional 13: Carrito de compras	73
4.2.7	Requisito Funcional 14: Filtro de productos	76
4.2.8	Requisito Funcional 15: Visualización de productos por usuario no autenticado	78
4.2.9	Requisito Funcional 16: Compra de carrito por usuario autenticado	79
4.2.10	Requisito Funcional 23: Gestión de direcciones de comprador	80
4.2.11	Requisito Funcional 24: Registro de envío de producto	83
4.2.12	Requisito Funcional 26: Mensajería posventa	84
4.3	Requisitos funcionales no implementados	86
4.3.1	Requisitos de la gestión de usuarios	86
4.3.2	Requisitos de gestión de tienda	86
4.3.3	Requisitos del marketplace	86
4.3.4	Requisitos de ventas	86
4.3.5	Requisitos de envío	87
4.3.6	Requisitos de analíticas	87
4.3.7	Requisitos de soporte	87
4.3.8	Requisitos de festividades y campañas	87
4.4	Demostración del sistema	87

4.4.1	Happy Path 1: Registro de vendedor y gestión de tienda	89
4.4.2	Happy Path 2: Registro de vendedor y su tienda	92
4.4.3	Happy Path 3: Registro de vendedor y su tienda	96
5	Validación	99
5.1	Validación de usabilidad	99
5.1.1	Vendedores	99
5.1.2	Compradores	100
5.2	Verificación funcional del backend	101
6	Conclusiones	103
	Glosario	107
	Bibliografía	113

Índice de figuras

3.1 Diagrama de Contexto	32
3.2 Diagrama de Arquitectura	33
3.3 Diagrama de Componentes	34
4.1 Diagrama de secuencia de autorización con estrategia “local”	45
4.2 Diagrama de secuencia de autorización con estrategia “jwt”	46
4.3 Diagrama de secuencia de validación de token de refresco con estrategia “refresh-jwt”	47
4.4 Diagrama de secuencia de autorización con estrategia “google-jwt”	48
4.5 Diagrama de secuencia de verificación del rol del usuario	49
4.6 Implementación de versionado por cabecera personalizada	51
4.7 Diagrama de flujo de autenticación e ingreso	53
4.8 Diagrama de flujo de registro de usuarios	54
4.9 Registro de Comprador	55
4.10 Registro de vendedor: Tipo de tienda	55
4.11 Registro de vendedor: Información y credenciales	56
4.12 Registro de vendedor: Confirmación	56
4.13 Acceso a cuenta del vendedor	57
4.14 Secuencia de autenticación de usuario	58
4.15 Secuencia de refresco de tokens	59
4.17 Secuencia de autenticación de usuario OAuth2 con Google	63
4.18 Diagrama de gestión de productos	64
4.20 Tienda sin productos	66
4.21 Interfaz de registro de producto	66
4.22 Registro exitoso de producto	67
4.23 Lista de productos registrados	67
4.24 Diagrama de flujo sobre la configuración de la tienda y la vista de los productos	68
4.25 Diagrama de secuencia sobre la personalización de una tienda	69

4.26	Formulario de personalización de imágenes y colores	70
4.27	Formulario de personalización de descripción y plantilla	71
4.28	Ejemplo de tiendas personalizadas	72
4.29	Diagrama de flujo de filtrado de productos	73
4.30	Diagrama de secuencia de sincronización y validación del carrito	74
4.31	Carrito de compras	75
4.32	Diagrama de flujo de filtrado de productos	76
4.33	Diagrama de secuencia de consulta sobre productos filtrados	77
4.34	Filtrado en tienda por categoría de producto	77
4.35	Carrito de compras cuando la sesión no está iniciada	79
4.36	Diagrama de flujo de gestión de direcciones de comprador	80
4.38	Panel de datos de envío antes de proceder a la compra	82
4.39	Resumen de la compra y datos de facturación	82
4.40	Opciones de envío	83
4.41	Vista del chat de comprador	84
4.42	Vista del chat de vendedor	85
4.43	Diagramas de flujo que demuestran la operación del sistema <i>Madeby</i>	88
4.44	Registro de vendedor	89
4.45	Personalización de tienda	90
4.46	Publicación de un producto	91
4.47	Landing page de <i>Madeby</i>	92
4.48	Vista de tienda y de producto	93
4.49	Notificación de producto agregado al carrito	93
4.50	Formulario de acceso o registro de comprador	94
4.51	Proceso de compra de carrito	95
4.52	Órdenes de compra del vendedor	96
4.53	Chat del Vendedor	97
4.54	Chat del comprador	98
5.1	Salida del test de extremo a extremo	101

Índice de tablas

1.1	Criterios de categorización según la ley 20416 y el código del trabajo	12
1.2	Estadísticas laborales y económicas del mercado chileno en función de la categoría	13
2.1	Características de la competencia	25
3.1	Descripción de Componente: Base de Datos SQL	35
3.2	Descripción de Componente: Backend	36
3.3	Descripción de Componente: Backoffice	36
3.4	Descripción de Componente: Marketplace	37
4.1	Comparativa de formas de especificar la versión	50
5.1	Rango etario de emprendedores entrevistados	100

Capítulo 1

Introducción

1.1 Primeras incursiones del mercado en Internet

En las primeras iteraciones del Internet no se considera la adición de protocolos de seguridad principalmente porque todos los nodos eran conocidos. A medida que la red crece, esta deja de ser patrimonio únicamente de empresas y las universidades y se da lugar a nuevas oportunidades para la población general.

Sin embargo, para habilitar las transacciones electrónicas se necesita transmitir información sensible, como datos bancarios y contraseñas. Como la infraestructura existente carece de protocolos de encriptación, la información transmitida en cada paquete estaba en texto claro y no permite estos casos de uso.

Es en 1994 que se logra realizar la primera transacción en Internet por medio del uso de software criptográfico [1] [2] [3]. Y en 1995 se incorpora el protocolo SSL a un cliente web [1] [4], lo que facilita la adhesión del usuario común de la red.

El comercio electrónico se establece con empresas como Amazon y Ebay. Ambas plataformas, que subsisten hasta el día de hoy, causaron una revolución en la economía en general y la forma en que se utiliza internet. Son la piedra angular del E-Commerce actual.

1.2 Web 2.0 y las redes sociales

En concepto de “Web 2.0” referencia al cambio de enfoque que se da al contenido de Internet, pasando del recurso estático a aquel generado por personas y comunidades [5]. Ahora el individuo es capaz de interactuar con las redes sociales por medio de la subida de videos, fotos o publicaciones en blogs o tableros.

Algunas plataformas de E-Commerce creadas a partir de estas tendencias son Wordpress, Joomla, Shopify y PrestaShop, entre muchas otras. La característica principal de los sitios mencionados es que permiten a los usuarios crear tiendas personalizables, en algunos casos sin requerir una infraestructura informática existente, lo

Tabla 1.1: Criterios de categorización según la ley 20416 y el código del trabajo

Tamaño de empresa	Ingresos Anuales (UF)	Cantidad de trabajadores
Micro	0 a 2400	1 a 9
Pequeña	Entre 2400 y 25000	10 a 49
Mediana	Entre 25000 y 100000	50 a 199
Grande	Entre 100000 y más	200 o más

cual puede ser una ventaja competitiva considerablemente al reducir los gastos operacionales. Este es el inicio de la “Democratización del E-commerce”.

Las aplicaciones que surgen gracias a la apertura del mercado electrónico no se limitan a la compra y venta de productos, sino que permiten a los usuarios rentar propiedades u ofrecer servicios, tal es el caso de Uber y Airbnb. En este contexto es que tanto el emprendedor como el usuario son beneficiados. El primero ve reducido los costos operacionales al no ser necesario invertir en infraestructura y mantenimiento para implementar su idea de negocio. Mientras que el segundo puede acceder a una gran cantidad de sitios y aplicaciones web ya sea como cliente o como un proveedor de la misma.

El CaaS surge como un modelo de negocio en que una plataforma ofrece la capacidad de implementar a servicios de comercio en la nube. Ofrece elementos como inventarios, pedidos, gestión de productos entre otras funcionalidades específicas, lo que permite a los usuarios enfocarse en el negocio y su crecimiento sin necesidad de desarrollar o mantener la infraestructura web de E-Commerce. El mayor exponente del mercado es “Shopify”, que con cerca de 4 millones de tiendas ha concretado 175 mil millones de dólares en ventas online durante 2021 [6].

1.3 Definición del problema

1.3.1 Caracterización del mercado chileno

Hay dos criterios para definir el estatus de un emprendimiento. Según la ley 20.416 conocida como “Estatuto Pyme”, la clasificación se basa en los ingresos anuales por ventas, servicios, entra otras actividades. Mientras que para el código del trabajo esta definición se basa en la cantidad de trabajadores. En la tabla 1.1 se resumen ambos criterios.

La tabla 1.2 recopila estadísticas nacionales del impacto que tienen las distintas categorías de empresa según la Encuesta Nacional de Empleo del INE y el Servicio de Impuestos Internos [7]. Las Micro, Pequeñas y Medianas Empresas (MIPYMES) representan en Chile el 98.6 % de las empresas del país y concentran el 65.3 % de los empleos formales para 2022 [8]. Aun así, realizan 7 veces menos ventas que las grandes empresas, pese a que estas solo constituyen un 1,4 % del mercado nacional.

Tabla 1.2: Estadísticas laborales y económicas del mercado chileno en función de la categoría

Tamaño de empresa	Porcentaje de empresas	Empleos generados	Concentración de ventas
Micro	75,5 %	37,3 %	13,1 %
Pequeña	23,1 %	13,1 %	
Mediana		14,9 %	
MIPYME	98,6 %	65,3 %	13,1 %
Grande	1,4 %	34,6 %	86,9 %
Total	1.020.115	7.432.926	704.790.476

1.3.2 MIPYMES y el mercado online nacional

La tendencia del comercio online y la digitalización han ido en aumento gracias al mayor acceso a internet de la población. Los distintos gobiernos han impulsado planes para mejorar el desarrollo digital en Chile desde los años 90, entendiendo el efecto positivo que una buena infraestructura digital tiene sobre la economía y la población general. A finales de 2015 se publica La Agenda Digital 2020 cuyo objetivo radica en la reducción de las brechas socioeconómicas por medio del uso de las tecnologías, el emprendimiento y la innovación. Entre las medidas propuestas se encuentra la implementación de la Fibra Óptica Austral, leyes regulatorias de datos personales y apoyo a la transformación digital de las PYMES [9].

El proceso de digitalización en Chile estuvo liderado por las grandes empresas, quienes en sus procesos de innovación siguen las tendencias vistas en otros países con empresas como Walmart, IBM y Amazon, entre otras. Según el Índice de Transformación Digital de Empresas de la Cámara de Comercio de Santiago y la consultora PMG [10] publicado en 2018, las grandes empresas tienen un avance considerable en los indicadores de transformación digital por sobre las PYMES, remarcando aspectos como la integración de herramientas digitales y una visión y estrategia que apunte a la digitalización e innovación.

Si bien la integración de las PYMES al mercado digital ha tenido un crecimiento sostenido durante los años, es a raíz del confinamiento y las restricciones causadas por la crisis sanitaria que el uso de herramientas digitales y la apuesta por la innovación y optimización de procesos empieza a percibirse como un paso fundamental para la reactivación y el crecimiento de estas.

Durante la pandemia de Covid-19, las MIPYMES fueron duramente afectadas por las medidas para prevenir el contagio. Del universo de empresas cuyas ventas cayeron a nivel nacional, las microempresas representan el 63,1 % de estas, además de reducir en promedio un 21,2 % su dotación de trabajadores [11]. Según el estudio Impacto Pandemia de la Asech [12] [13], el 50,2 % de los encuestados declara que su emprendimiento fue afectado gravemente o catastróficamente, lo que significa que se vieron en la necesidad de reducir personal o, en algunos casos, suspender indefinidamente las operaciones de la empresa. Además, se indica que los riesgos que más preocupan a los emprendedores son la falta de liquidez por baja venta, las deudas con entidades financieras y

los retrasos en pagos de clientes o usuarios deudores.

En 2020 el crecimiento de más de un 100 % de las ventas por internet contribuyó a que la contracción de ventas minoristas no fuera tan abrupta [14], aunque hay que reconocer el aporte de los bonos estatales y los retiros previsionales. Una encuesta realizada a PYMES por MercadoLibre, que supone tener cierto grado de digitalización, indica que 8 de cada 10 ventas realizadas en este periodo fueron a través de internet [8].

1.3.3 Retos de la incursión en el mercado online

La adopción de un modelo de negocios basado en el E-Commerce es un proceso con varias aristas, ya que este mercado impone requisitos adicionales como la compatibilidad con dispositivos móviles y presencia en redes sociales. Estas características suelen representar costos y conocimientos adicionales que son una barrera de entrada a pequeñas y medianas empresas. Ante esto, opciones con precios de entrada menores suelen presentar una serie de dificultades y restricciones:

- **Elevadas comisiones por venta:** Si el emprendedor no busca un modelo de negocio de pago mensual, existen alternativas que cobran comisiones por cada venta que realice, estas comisiones suelen ser de hasta el 2 % por compra si se busca el plan más económico. Al no ser un sistema dinámico que se regule según la cantidad de ventas que realice, puede llegar a causar problemas en los ingresos del negocio.
- **Poca o nula promoción:** Si el emprendimiento implementa únicamente su negocio online y no se preocupa de integrarlo a un marketplace o a redes sociales, no tendrá un punto de descubrimiento para que personas ajenas al negocio lo puedan encontrar.
- **No incluir venta online:** Existen servicios que solo muestran tiendas y los productos de estas, pero carecen de integración para generar la compra. Esto implica que la tienda debe gestionar la venta con sus propios medios o con servicios de terceros.
- **Pérdida de identidad de marca:** Un marketplace, por lo general, prioriza la venta de productos según las especificaciones indicadas por los compradores, ignorando aspectos como la identidad de los negocios que disponen de estos.

Tomando en consideración lo expuesto, los problemas identificados son las barreras de entrada al E-Commerce y la pérdida de la identidad de marca del negocio.

1.3.4 Hipótesis

Una plataforma de E-Commerce enfocada en el mercado MIPYME que disponga de una tienda donde el emprendedor plasme la identidad de su negocio y el valor del producto elaborado, sumado a un marketplace que dé

énfasis al emprendimiento y permita conectar a los consumidores con los vendedores va a aportar positivamente al proceso de digitalización, el crecimiento y la capacidad de innovación de las PYMES.

1.3.5 Objetivos

Objetivo general

Desarrollar una plataforma E-Commerce orientada a la venta de productos de fabricación local, que permita al emprendedor gestionar su tienda digital de manera sencilla, eficiente, a un costo accesible mientras que permita al comprador acceder a un mercado que resalte la identidad única de cada negocio y el valor del producto local.

Objetivos específicos

1. Desarrollar un validar una propuesta de solución por medio del análisis de problemáticas del desafío y la información existente sobre el contexto
2. Diseñar una plataforma E-Commerce con técnicas de arquitecturas limpias, mantenibles y basadas en el viaje del usuario y sus interacciones
3. Implementar el mínimo producto viable (MPV), desplegado por medio de plataformas de computación en la nube
4. Validar el MPV por medio de encuestas y entrevistas a usuarios para obtener métricas de la utilidad percibida

1.3.6 Estructura de la memoria

El presente documento se estructura de la siguiente manera: en el segundo capítulo se realiza un análisis del estado del arte, proporcionando un marco teórico y contextual respecto a las alternativas existentes en el mercado, tipos de aplicaciones y sus características. Esta información es utilizada en el tercer capítulo para abordar el desarrollo de la solución, detallando los requerimientos funcionales y no funcionales del sistema, además del enfoque adoptado para llegar a la solución final. Una vez definidos los requisitos del sitio web se presenta el procedimiento de diseño, implementación y validación en el cuarto capítulo, Finalmente, se concluye con una reflexión sobre los resultados obtenidos y las conclusiones generales del proyecto.

Este proyecto se enmarca en el contexto de memorias multidisciplinarias del año 2022, y ha sido realizado a partir de un problema propuesto por la empresa ITAXXION.

Capítulo 2

Estado del Arte

Para establecer un negocio digital y alcanzar estabilidad y crecimiento, existen varios requisitos que deben ser cubiertos en mayor o menor medida. En el mercado del E-Commerce existen muchas alternativas que se encargan de aspectos particulares como las transacciones por internet, el alcance a los consumidores o establecer una tienda online.

En particular, esta investigación se enfoca en caracterizar las soluciones existentes en cuanto a la cobertura de los requerimientos de un negocio digital:

- Crear una tienda online
- Publicar avisos de venta
- Gestionar la venta del producto
- Acceder a un marketplace
- Personalizar la tienda
- Acceso a métricas de negocio
- Capacitación sobre el mercado digital
- Énfasis en productos elaborados
- Venta de productos
- Venta de Servicios
- Costo de entrada accesible

Esta investigación resultó en cinco categorías que agrupan las funcionalidades de la competencia, estas serán explicadas en profundidad en las siguientes secciones. Las categorías mencionadas son: creadores de sitios web, creadores de tiendas, sistemas de ventas en redes sociales, servicios de consultoría y mixtos:

2.1 Creadores de sitios web

Plataformas que ofrecen herramientas para crear y mantener sitios web de E-Commerce en internet, altamente personalizables y con capacidad para realizar transacciones y hacer marketing por email. Suelen cobrar suscripciones mensuales, y cada característica implica costos adicionales.

Algunos sistemas en esta categoría:

- “GoDaddy” (1997): Compañía de hosting de servicios web y venta de registros de dominio. Creada como registro de dominios en la red, se expande al “Web Hosting” al ofrecer alternativas para levantar sitios web de forma rápida, según la capacidad del cliente. El nivel de personalización del sitio se limita al alcance del negocio y las habilidades de los desarrolladores. Al no disponer de un servicio Marketplace ni promover la venta del producto elaborado, se considera que cubre el 81 % de las necesidades del emprendedor.
- “Wix” (2006): Plataforma de creación de páginas Web. Ofrece un amplio número de plantillas para personalizar la página, sin limitarse al E-Commerce, y dispone de servicios como sistemas de compra y disponibilidad de métricas. Con interfaces intuitivas y un gran número de complementos, un usuario puede crear sitios web sin necesidad de escribir una línea de código. Sin embargo, se ha reportado de fallas en el rendimiento por la saturación de elementos en la página. Cubre la necesidad expuesta en un 81 %.

Este modelo de negocio no es compatible con empresas que están comenzando a digitalizarse, ya que requieren de una gran inversión de recursos con una largo periodo de retorno de inversión. Como estas soluciones no cuentan con alternativas de marketplace, el vendedor queda apartado de los mercados donde concurren potenciales compradores, lo que implica un costo adicional en promoción.

2.2 Creadores de tiendas

Estas plataformas tienen un enfoque exclusivo de tiendas electrónicas, y su modelo de negocio se sostiene en el cobro de comisiones por venta. Esta característica puede ser conveniente a los emprendedores, ya que la barrera de entrada se ajusta a la capacidad de venta

- “Shopify” (2006): La plataforma de creación de tiendas e-commerce más popular. Fundada en 2006 con el objetivo de proporcionar a empresas y vendedores la capacidad de gestionar tiendas en línea de manera fácil y eficiente. Ofrece una amplia gama de servicios, entre los que se destaca el sistema de pagos, mentorías y

capacitaciones, campañas de marketing y gestión de envío. En su oferta, considera planes especializados para las necesidades de distintas categorías de negocio, lo que puede ser Dentro de su catálogo de servicios no ofrece puntos de encuentro entre vendedores y compradores que sirvan de marketplace, por lo que cumple el 90 % de las necesidades postuladas.

- “JumpSeller” (2010): Alternativa al uso de “Shopify” orientada al mercado latino de creación de tiendas online. Su propuesta es similar, pero su enfoque al mercado latino permite integraciones con pasarelas de pago específicas y un servicio al cliente con mayor conocimiento de las PYMES en Chile y latinoamérica. Por ofrecer el mismo tipo de servicio que “Shopify”, también cumple con los objetivos requeridos en un 90 %.
- “TuShop” (2021): El E-Commerce para tiendas de Instagram. Nace con el fin de ofrecer un carrito de compras y la capacidad de manejar transacciones de productos a usuarios de Instagram. Apunta al mercado de los emprendedores que utilizan redes sociales y aprovecha el alcance de estas para promover a los negocios, esto puede ser una alternativa al marketplace. Hoy en día “tushop.cl” no existe como dominio en la red y se asume que no presenta actividades como empresa. Se destaca que cumple con varios de los requisitos planteados, con un rango de cobertura del 90 %

Las tiendas hechas sobre estas plataformas suelen incluir características específicas del E-Commerce como inventarios de productos, conexión a una pasarela de pago, administración posventa, entre otros. En general carecen de integración a marketplace, lo que requiere gestionar la promoción por medio de servicios de terceros o redes sociales. No incorporar mecanismos de promoción a una tienda online dificulta la captura de clientes, por lo que esta característica es necesaria.

Por otra parte, el enfoque al E-Commerce simplifica la personalización y permite reducir tiempos y costos de desarrollo, lo que hace de estas alternativas una opción viable para muchas PYMES.

2.3 Ventas en redes sociales

Redes sociales como Facebook o Instagram se han convertido en parte importante de la interacción entre personas, teniendo un uso diario cercano a las dos horas y media en 2022 [15]. Hoy en día es común que un emprendedor inicie su proceso de digitalización en estas aplicaciones y es válido su uso para incrementar el alcance sobre el mercado objetivo.

Las redes sociales a destacar son las siguientes:

- “Facebook” (2004): Surge como un punto de encuentro en donde compartir experiencias e interactuar con otras personas. Hoy en día es la red social con mayor cantidad de usuarios a nivel mundial, con 2.91 millones de usuarios a nivel mundial [15]. En 2016 se publica “Facebook Marketplace”, una experiencia

que permite a las personas comprar y vender artículos dentro de Facebook. Esta herramienta permite al vendedor acceder a un marketplace, pero se desliga del proceso de capacitación sobre gestión del negocio, personalización de tienda online y métricas de vistas a los productos. Tampoco cuenta con capacidad de gestionar la venta, por lo que no entrega información sobre transacciones. Se determina un 63 % de objetivos cumplidos.

- “Instagram” (2010): Red Social enfocada en compartir fotografías y dibujos, permitiendo usar filtros preestablecidos y etiquetas arbitrarias para categorizar el contenido de la imagen, lo que facilita la llegada al público objetivo. Actualmente permite la subida de videos de duración reducida, historias que se mantienen públicas por un día y realizar directos, siguiendo las tendencias de otras aplicaciones como Snapchat y TikTok. A la hora de publicar productos y servicios, la principal limitación es la falta de métodos de pago para concretar las transacciones. Recientemente se ha implementado la herramienta “Shops”, una nueva característica que acerca la red social al E-Commerce al agregar un menú propio de tienda para cuentas con fines comerciales. Esto simplifica la búsqueda de un usuario para encontrar algo que le interesa y lo redirige directamente al sitio de la tienda del vendedor en la página del producto seleccionado. Al igual que Facebook, Instagram cumple con los objetivos en un 63 %.

El uso de redes sociales para mejorar la visibilidad de una tienda entrega ventajas como la reducción de costos en publicidad y también se puede apuntar a un nicho de consumidores específico por medio de analíticas de terceros.

Si bien publicar productos y servicios no tiene un costo, estas aplicaciones no cuentan con un modelo de negocio orientado al E-Commerce, por lo que carecen de funcionalidades como ventas en línea, carritos de compra y personalización de un espacio. Esto fuerza a los usuarios a ocupar otros medios para realizar las transacciones, lo que puede dar lugar a que ocurran estafas o malentendidos. Además, al no contar con mecanismos de personalización de la tienda, se dificulta la tarea de imprimir una identidad de marca que destaque en el mercado.

Tomando en consideración la influencia de las redes sociales en la actualidad, cualquier propuesta de valor o idea de negocio debe tenerlas en cuenta y aprovechar su uso al momento de captar nuevos clientes.

2.4 Servicios de consultoría y apoyo

Las empresas dedicadas a la capacitación y orientación tienen como objetivo principal impulsar el crecimiento y desarrollo de los emprendedores en el mercado digital. Existen muchos tipos de empresas que cumplen con este criterio, pero con fines de ejemplificar se mencionan las siguientes:

- “Mutatis” (2018): Consultora certificada por MercadoLibre para alfabetizar a emprendedores sobre el mundo digital, enseñando a vender por internet, entregando estrategias para mejorar la performance y

acompañando en etapas de seguimiento. Además del servicio de consultoría, en su sitio web disponen de un blog donde se entregan recomendaciones y explicaciones respecto a dudas comunes.

- “ApoyameAquí” (2020): Nace como iniciativa de la cámara de comercio de Santiago en respuesta a los efectos de la pandemia y la tendencia del E-Commerce. Da orientación sobre que alternativas existen para dar inicio al proceso de digitalización, como plataformas de creación de tienda, acceso a varios marketplace y pasarelas de pago existentes en el mercado nacional. Al estar vinculado a la Cámara de Comercio de Santiago, también permite acceder a sus talleres y asesorías en tópicos como tendencias y proyecciones económicas, indicaciones sobre el servicio de impuestos internos y buenas prácticas.
- “Neuronal” (2018): Su propósito es facilitar la digitalización a los negocios, estableciendo planificaciones de transformación en función de las características de la empresa. Ofrece pruebas gratuitas para determinar el nivel de digitalización del emprendimiento y un blog con información relacionada con la digitalización y estrategia de mercado.

Por lo general las empresas de esta categoría ofrecen servicios de consultoría especializada, con información obtenida investigaciones de mercado y revisión de revistas y diarios financieros, además de estudios sobre las herramientas existentes a la hora de digitalizar un negocio. Esta información se presenta en función de las necesidades específicas de los emprendimientos y se utiliza para determinar estrategias y planes de mercado.

No se contempla el servicio de creación de tiendas o de marketplace, sino que hay una dedicación al educar y capacitar al emprendedor sobre el panorama económico, las aplicaciones que puede ocupar y estrategias para crecer en el mundo digital. Se aprovecha la existencia de empresas cuyo rubro es la infraestructura del E-Commerce.

Dadas las características mencionadas, se determina una cobertura del 9 % sobre los objetivos planteados principalmente porque no incluyen servicios para publicitar o realizar ventas, sino que se basan en tecnologías existentes y metodologías de trabajo informadas para guiar a los negocios.

2.5 Servicios mixtos

Aquellos que integran características de otras categorías vistas, como la integración con un marketplace, herramientas de gestión de negocio y ventas, creación y personalización de tienda, entre otras funcionalidades.

- “MercadoShops” (2010): Herramienta para el diseño y gestión de tiendas online que no tiene costo de activación ni de mantenimiento. Al ser un producto del ecosistema de “MercadoLibre” es fácil de integrar en el marketplace, lo cual es una gran ventaja al ser ocupado del 90 % de los chilenos que compra por internet [16]. También es altamente integrable con servicios como “MercadoADS” o “MercadoPago”. Ser parte de un marketplace tan grande implica, a su vez, mayor competencia entre las MIPYMES y las

grandes empresas y multinacionales, lo que dificulta captar consumidores en ciertos mercados. Se destaca el nivel de simpleza de la interfaz, servicio al cliente y métricas de tienda que el usuario puede acceder.

- “Etsy” (2005): Surge como un apoyo a la comunidad de creadores y artesanos que venden productos únicos y personalizables, además de promover estilos de vida sostenibles y conscientes. Dispone de un marketplace y un gestor de tienda con los que satisfacen varias de las necesidades del comercio digital. Reducir el mercado objetivo permite diferenciarse de la competencia y atacar necesidades específicas de un nicho de consumidores, en este caso de aquellos que buscan productos únicos y distintivos. Presenta altas barreras de entrada y no considera la oferta de servicios. En consecuencia, cumple los objetivos en un 81 %.
- “Marketplace de Empresas Retail”: El mercado chileno está dominado por tres grandes empresas: Cencosud, Falabella y Ripley. El crecimiento de las PYMES y las tendencias de la economía nacional instaron la implementación de soluciones que permitan la venta de productos de pequeños negocios en el mismo sitio web, siendo el origen de “Paris Marketplace”, “Falabella Marketplace” y “Ripley Marketplace”.

Para el emprendedor, significa acceder a una plataforma que tiene gran visibilidad a nivel nacional sin requerir inversión en desarrollo web. También puede utilizar la logística de entrega y el soporte técnico, además de cursos de capacitación sobre digitalización y el mercado digital.

Las desventajas que esta solución implica radican en la mayor competencia con grandes marcas, lo que puede reducir la atención del consumidor. Además de la pérdida de identidad de marca del negocio dado que el enfoque de los marketplace está en la venta de productos, ignorando otros aspectos.

Los marketplace de empresas de retail cumplen con los objetivos en un 64 %.

- “Fiverr” (2010): Plataforma orientada a la oferta de servicios profesionales que nace para conectar a freelancers con clientes de todo el mundo por medio de un marketplace. Cubre una amplia gama de categorías como lo son el diseño gráfico, doblaje de voz, marketing digital, redacción de documentos, entre un amplio abanico y con un rango de precios según la calidad y experiencia del freelancer. El sistema cuenta con la gestión de una tienda de “gigs” donde cada usuario describe lo que ofrece, los posibles precios y ejemplos de trabajos anteriores, y un mecanismo de pago que requiere de la entrega del producto, lo que da mayor seguridad a los compradores.

Un aspecto a destacar es el sistema de chat entre el freelancer y el comprador que cuenta con capacidad de crear transacciones de venta, ajustes al pedido y verificar que el servicio se realizó de forma aceptable.

Dado que los objetivos para la resolución del problema principal requieren de la venta de productos, se considera que esta aplicación cumple con los objetivos en un 81 %

2.5.1 Marketplaces nacionales

Existe un subgrupo de emprendimientos que buscan habilitar un sistema Marketplace dedicado a las PYMES, cada uno con particularidades pero compartiendo el propósito de unificar a los emprendimientos locales en un solo lugar.

1. “contigopyme.cl” (2020): Impulsado por la cámara de comercio de Valparaíso, consiste en un marketplace para PYMES sin costo en donde se pueden registrar productos y ser comprados por medio de una pasarela de pagos. Realiza eventos Cyber tanto a nivel regional como nacional y no tiene límite para subir ofertas a la plataforma.

Cabe mencionar que la plataforma no ofrece personalización a cada tienda que se registra en el sistema, tampoco capacitaciones a los vendedores sobre como gestionar el negocio, por ende cumple con los objetivos en un 63 %

2. “Apanio” (2021): Plataforma de apoyo a las PYMES que permite acceder a canales de venta, publicidad y marketplace de emprendedores locales entre otras herramientas. Al registrar una tienda se define el tipo de producto o servicio a ofrecer y se entregan métricas de las ventas generadas.

No ofrece ningún tipo de capacitación al vendedor para desenvolverse en el mercado digital, por lo que cumple los objetivos en un 81 %.

3. “MercadoProPyme” (2020): marketplace enfocado en el mercado chileno de PYMES que ofrece creación y personalización de tienda online, capacidad de realizar transacciones y realización de campañas de marketing para apoyar la visibilidad de los negocios.

La gestión de la entrega de los productos se realiza entre vendedor y cliente, ya que no integra herramientas de gestión de envío. Además, no realiza capacitaciones a los vendedores, por lo que cumple un 81 % de los objetivos

4. “Sumar.cl” (www.sumar.cl): Busca implementar un “mall virtual” en que los emprendedores publiquen su tienda online y registren sus productos. Dado que fue diseñado por los creadores de la pasarela de pagos “FLOW”, las transacciones se realizan con esta herramienta con la ventaja de tener comisiones más bajas.

No realiza capacitaciones, cumple los objetivos en un 81 %.

5. “Cuponatic” (2010): Busca ser un punto centralizado donde encontrar promociones y descuentos exclusivos en distintos servicios ofrecidos por empresas en un amplio rango de rubros como horas en saunas y spas, tickets para ir al cine o tours en lugares turísticos.

Dentro de sus prioridades no está la promoción de las PYMES ni vender productos, sino que obtener precios bajos para los servicios que estas puedan ofrecer. Se determina una cobertura del 64 % de los objetivos.

6. “TDiseña” (2019): Venta de artículos y servicios para el diseño de interiores y mueblería. Permite a emprendedores ofrecer sus servicios de interiorismo o paisajismo y también vender los materiales y productos utilizados para tal fin.

Este marketplace permite la venta de productos y servicios, cobrando comisiones por venta y entrega métricas sobre las ventas. Tiene una cobertura del 81 % sobre los objetivos establecidos.

El problema del ecosistema de marketplace nacional es la amplia presencia de emprendedores que revenden productos industriales de bajo costo y calidad, siendo coloquialmente asociados a la industria China. Esto reduce el valor que estos pueden ofrecer tanto a emprendedores como a los consumidores, porque los primeros verán competencia en productos de bajo costo, mientras que los últimos dispondrán de una oferta inundada de elementos de baja calidad. marketplace como AliExpress, Temu y MercadoLibre no implementan acciones para evitar este fenómeno.

De las empresas analizadas, se destaca TDiseña por su enfoque a un segmento específico del mercado, lo que permite al comprador buscar emprendimientos que dedican tiempo y esfuerzo a la creación de productos innovadores y originales.

2.6 Ventajas y desventajas de las alternativas

A partir del análisis de las empresas revisadas, se presenta una tabla comparativa que resume el alcance de cada una sobre los objetivos específicos considerados.

En la tabla 2.1, las columnas “a” hasta la “k” indican las capacidades que una plataforma web debe tener para ayudar al vendedor en el camino de digitalizar su negocio:

- a: Crear una tienda online
- b: Publicar avisos de venta
- c: Gestionar la venta del producto
- d: Acceder a un marketplace
- e: Personalizar la tienda
- f: Acceso a métricas de negocio
- g: Capacitación sobre el mercado digital
- h: Énfasis en productos elaborados
- i: Venta de productos

Tabla 2.1: Características de la competencia

Alternativas	a	b	c	d	e	f	g	h	i	j	k
MercadoLibre	NO	SI	SI	SI	NO	SI	NO	NO	SI	SI	SI
MercadoShops	SI	SI	SI	NO	SI	SI	NO	NO	SI	SI	SI
Wix	SI	SI	SI	NO	SI	SI	SI	NO	SI	SI	SI
GoDaddy	SI	SI	SI	NO	SI	SI	SI	NO	SI	SI	SI
Facebook Marketplace	SI	SI	NO	SI	NO	SI	NO	NO	SI	SI	SI
Mutatis	NO	NO	NO	NO	NO	NO	SI	NO	NO	NO	NO
Etsy	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO
TuShop	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI
Marketplace de Retails	SI	SI	SI	SI	NO	SI	SI	NO	SI	NO	NO
Fiverr	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	SI
Shopify	SI	SI	SI	NO	SI	SI	SI	NO	SI	SI	SI
JumpSeller	SI	SI	SI	NO	SI	SI	SI	NO	SI	SI	SI
ApoyameAquí	NO	NO	NO	NO	NO	NO	SI	NO	NO	NO	NO
contigopyme.cl	SI	SI	SI	SI	NO	SI	NO	NO	SI	NO	SI
Apanio	SI	SI	SI	SI	NO	SI	SI	NO	SI	SI	SI
MercadoProPyme	SI	SI	SI	SI	NO	SI	SI	NO	SI	SI	SI
Sumar.cl	SI	SI	SI	SI	NO	SI	SI	NO	SI	SI	SI
Cuponatic	SI	SI	SI	SI	NO	SI	NO	NO	NO	SI	SI
TDiseña	SI	SI	SI	SI	NO	SI	NO	SI	SI	SI	SI
Neuronal	NO	NO	NO	NO	NO	NO	SI	NO	NO	NO	NO

■ j: Venta de Servicios

■ k: Costo de entrada accesible

Según el análisis realizado, la solución conjunta de “MercadoLibre” y “MercadoShops” tienen una amplia cobertura de las necesidades identificadas, además de incorporar características importantes como la logística de distribución y el servicio al cliente. Sistemas como Shopify y JumpSeller destacan por la simpleza al momento de crear e imprimir una marca la tienda online, a diferencia de los marketplace de retail, cuya reducida personalización y el hecho de ser un competidor más impactan en la identidad y alcance de cada emprendimiento. La europea Etsy y las nacionales TDiseña y Apanio establecen mercados de productos originales y de calidad, acotando el nicho de usuarios objetivo con una propuesta de valor que se distancie de competencia como MercadoLibre o AliExpress.

Según el análisis realizado, se concluye que ninguna solución existente puede satisfacer por si sola todas las necesidades de las PYMES. Sin embargo, es posible extraer características de interés que puedan ser utilizadas para cumplir los objetivos.

Capítulo 3

Diseño de la Solución

En este capítulo, se realiza un análisis de los conceptos y términos esenciales para luego diseñar una solución que integre los puntos identificados.

3.1 Análisis

3.1.1 Alternativas de solución

Se han identificado algunos conceptos como necesarios y/o deseables para la solución. A continuación se exponen dichas características.

Marketplace

El formato de Marketplace muestra varios productos de distintas tiendas a los usuarios, facilitando la búsqueda de algún artículo deseado. El problema es que los vendedores compiten por los mismos clientes, por lo que nuevas empresas se ven desfavorecidas por la guerra de precios que se crea en este tipo de plataformas.

Creador de tienda

Ofrecer a las MIPYMES una plataforma en la cual puedan iniciar su digitalización mediante la creación de su propia tienda web personalizable. Se utilizarán elementos que permitan la fácil elaboración del sitio con variadas posibilidades para personalizar el aspecto de la página, como lo serían plantillas, selección de colores y tipografías, y la capacidad de arrastrar y soltar elementos gráficos, entre otras.

Gestión de envíos de paquetes

Sitios como MercadoLibre son reconocidos por su alcance a la hora de realizar entregas, esto se logra al disponer de una infraestructura de bodegas dispersas en puntos estratégicos.

Blogs

Un blog permite publicar noticias y artículos relacionados con el negocio, como la llegada de nuevos productos,

eventos en donde habrá un stand de la tienda, entre otros. El problema que puede presentar esta funcionalidad es que puede ser ignorada por los clientes, o ser abandonadas y transmitir al comprador que la tienda no tiene actividad.

Interfaz con otros Marketplace

Incluir la capacidad de conectarse a otras plataformas Marketplace, como Instagram Shops y Facebook Marketplace. Esto habilitaría a los vendedores a incursionar en mercados ya establecidos.

3.1.2 Acercamiento a la solución

Utilizar un formato de Marketplace facilita la conexión entre varios compradores y tiendas. Sin embargo, se dará la opción para que cada tienda pueda personalizar su contenido con su identidad de marca. Esto permite que los compradores conozcan la empresa que les vendió el producto y no únicamente la plataforma donde se realiza la transacción.

Respecto a las fuentes de ingresos, las comisiones se adecuan de mejor manera al contexto de los nuevos emprendedores, ya que les permite comenzar a publicar sus productos sin riesgo. El modelo de suscripciones no se descarta totalmente, ya que se podría aplicar para ofrecer nuevas funcionalidades en versiones futuras.

3.2 Diseño

En esta sección se realiza un diseño preliminar de una solución que pueda ser implementada en el transcurso de la memoria, comenzando con la lista de requisitos del sistema y concluyendo con las herramientas necesarias para su operación.

3.2.1 Requisitos del sistema

En primer lugar se definen los requisitos funcionales necesarios para responder a las necesidades de los actores:

Requisitos de gestión de usuarios

RF 1 Un usuario, vendedor o comprador, puede realizar un registro básico con un correo electrónico y contraseña

RF 2 Un usuario puede recuperar su contraseña

RF 3 Un usuario puede autenticarse utilizando sitios de terceros

RF 4 Un usuario puede autenticarse con doble factor

RF 5 Un usuario puede eliminar su cuenta

RF 6 Un comprador puede indicar sus intereses, hobbies y elementos que permitan hacer recomendaciones

RF 7 Un comprador puede ver su historial de compras

Requisitos de gestión de tienda

RF 8 El vendedor puede registrar y editar su tienda con información del negocio

RF 9 El vendedor puede definir el estilo de su tienda con elementos como el logo, esquema de colores y categorías asociadas

RF 10 El vendedor puede publicar productos y editarlos

RF 11 El vendedor puede editar la presentación de sus productos ante los compradores

RF 12 El vendedor puede asignar otros vendedores con diferentes roles de administración en su tienda

Requisitos del Marketplace

- RF 13** Un comprador puede seleccionar productos de distintas tiendas en su carrito
- RF 14** Un comprador puede filtrar los productos según el nombre, categorías, precios y otros criterios de interés
- RF 15** Un comprador puede hacer preguntas respecto a un producto y un vendedor puede responderlas
- RF 16** Un usuario no autenticado puede ver los productos del marketplace y agregarlos a su carrito
- RF 17** Un usuario no autenticado debe registrarse para acceder a la compra de productos

Requisitos de ventas

- RF 18** Un comprador puede ocupar distintos métodos de pago
- RF 19** El vendedor puede aplicar ofertas a sus productos
- RF 20** Luego de una transacción, tanto comprador como vendedor pueden enterarse del estado de la compra
- RF 21** En caso de detectar un posible error en precios o descuentos, el vendedor será notificado
- RF 22** El comprador puede reseñar cada producto que compra

Requisitos de Envío

- RF 23** El comprador puede establecer múltiples direcciones de entrega preferidas
- RF 24** Luego de establecer al menos una dirección de entrega, el comprador puede recibir recomendaciones de tiendas y productos cercanos
- RF 25** El vendedor debe registrar los datos de envío necesarios después de acordarlo con el cliente

Requisitos de analíticas

- RF 26** El vendedor puede acceder a un panel de estadísticas e información relevante sobre la actividad de su tienda

Requisitos del chat

- RF 27** Luego de concretar una compra, un comprador puede acceder a un canal de mensajería instantánea con el vendedor para coordinar la entrega del producto

Requisitos de soporte

RF 28 Vendedores, compradores y administradores pueden notificar al sistema ante el uso inapropiado

RF 29 El vendedor será notificado sobre productos inapropiados y tendrá un plazo determinado para realizar los cambios correspondientes

RF 30 Un usuario puede registrar un ticket con problemas técnicos de la plataforma

Requisitos de festividades y campañas

RF 31 Los usuarios deben ser notificados de eventos y campañas de emprendedores

RF 32 Los vendedores que participan en eventos pueden asociar productos u ofertas a este para mejorar su alcance

Por otro lado, es necesario definir los requisitos no funcionales para saber las preferencias del cliente

Requisitos no funcionales

Se relacionan al rendimiento de la aplicación y características que no se perciben directamente por los usuarios, como la seguridad

RNF 1 Cada usuario debe estar autenticado de manera segura, confiable y sus contraseñas deben estar cifradas

RNF 2 La interfaz debe tener un diseño atractivo e intuitivo para los compradores, de tal forma que lo incentive a navegar por la tienda

RNF 3 Cada acción relevante de un comprador debe ser monitorizada e integrada en las estadísticas internas

Al saber los puntos específicos que se deben cumplir, es posible generar un diagrama de contexto que nos permite entender de mejor manera quienes son las entidades y actores que participan en el sistema y como interactúan entre sí.

3.2.2 Diagrama de contexto

La figura 3.1 representa la interacción del sistema marketplace con el entorno externo, en este caso los elementos principales que interactúan con el sistema son los compradores, vendedores, la pasarela de pagos y el repositorio de imágenes.

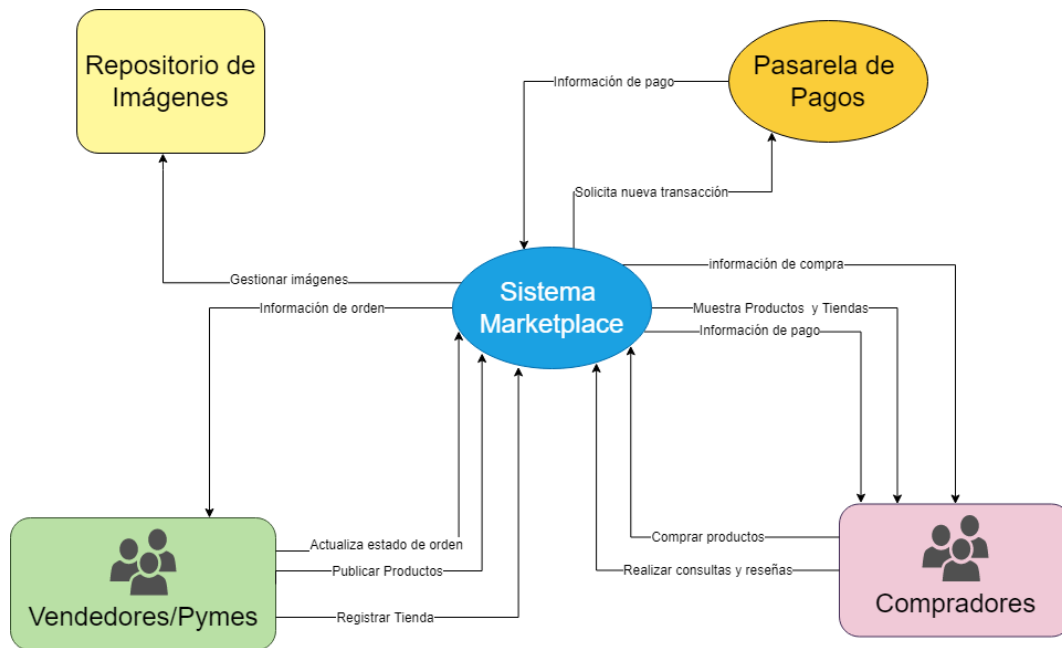


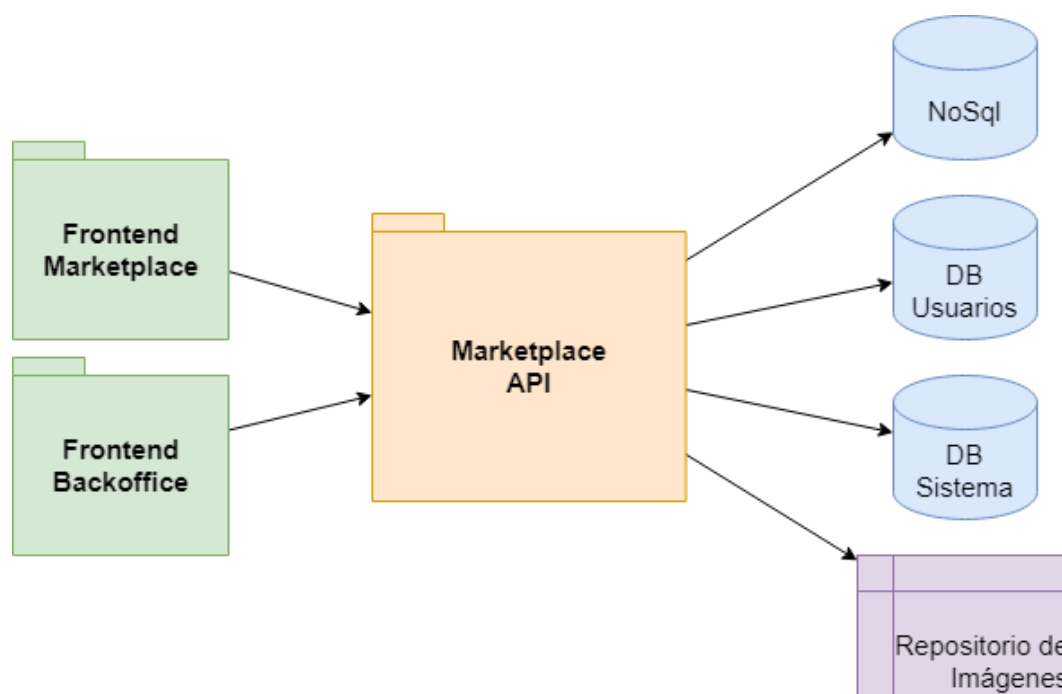
Figura 3.1: Diagrama de Contexto

3.2.3 Diagrama de arquitectura

El núcleo de la plataforma sigue una arquitectura “headless” en que se separa la lógica de la interfaz de usuario (frontend) de la lógica de negocio (backend). Del lado del frontend, se implementa un servicio orientado al flujo de trabajo de los compradores y otra dedicada a los vendedores y la administración de tiendas, denominados **Frontend Marketplace** y **Frontend Backoffice**. El backend, **Marketplace API**, centraliza el acceso a la información del e-commerce, dispone de una API REST para realizar consultas e implementa los procesos del negocio.

Se requiere además de integrar servicios como bases de datos relacionales y no relacionales, repositorios para imágenes y documentos y un web server para operar como proxy reverso y gestionar el enrutamiento de peticiones, manejo de versiones y certificados SSL (HTTPS).

El diagrama 3.2 muestra la arquitectura de la plataforma e-commerce Madeby.

**Figura 3.2:** Diagrama de Arquitectura

3.2.4 Descripción de módulos

Frontend Marketplace

Este módulo maneja todas las vistas asociadas a la lógica de negocio asociada a la compra y venta de productos elaborados, además exhibe los productos de las diferentes tiendas y es capaz de filtrarlos según las categorías pertinentes. También permite el registro y proceso de compra para un usuario, otorgando la posibilidad de agregar productos al carrito, modificar la cantidad de elementos que desea, finalizar su compra redirigiendo al comprador a una pasarela de pagos y gestionar el proceso de envío de sus productos adquiridos mediante información directa y canales de comunicación con el vendedor.

Frontend Backoffice

El módulo de administración consiste en otorgar todas las funcionalidades que necesitan los vendedores para registrarse, inscribir su tienda, agregar sus productos, gestionar sus órdenes de compra, obtener información de valor a través de analítica y manejar los canales de comunicación con los compradores mediante vistas accesibles con un menú general que contenga todas las opciones.

Marketplace API

Los módulos de frontend acceden a la lógica contenida en el servidor mediante una interfaz de programación de aplicaciones (API) según el protocolo HTTP(S). Cada petición es comprobada según distintos criterios de autenticación, filtrado y transformación de los datos que cada ruta pueda implementar en función de la lógica de negocio.

3.2.5 Diagrama de componentes

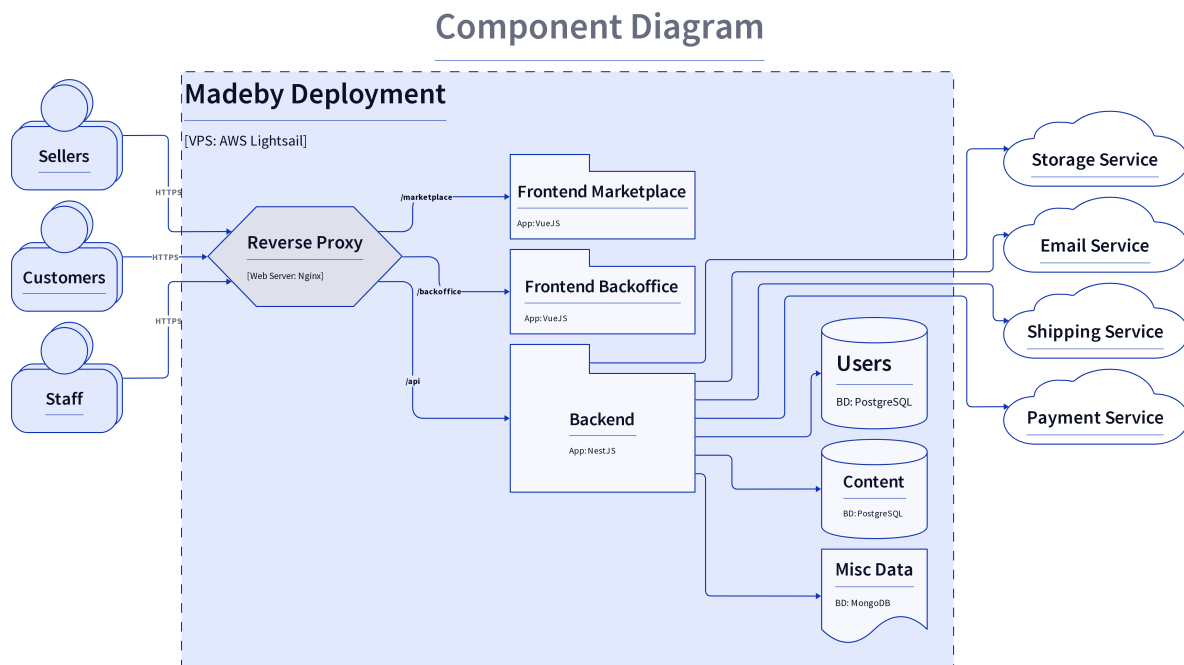


Figura 3.3: Diagrama de Componentes

Como se puede apreciar en la figura 3.3, los elementos señalados en el diagrama de arquitectura son detallados según los servicios internos y externos que afectan al sistema. En este sentido, tanto los sistemas frontend como el backend y las bases de datos serán parte del sistema interno a desarrollar, y este dependerá de servicios externos como la interacción de los usuarios y los siguientes sistemas:

Sistema de Envío

Aquel que se encarga de todos los envíos de productos a los compradores, se externaliza porque depende de los servicios de courier disponibles en cada lugar.

Sistema de Pagos

Comprende los sistemas de Pasarela de Pagos que permiten a los compradores utilizar distintos métodos de pago para realizar sus compras.

Repositorios

Sistemas de almacenamiento para imágenes

Sistema de Correo

Su función cumple realizar campañas de marketing y anuncios personalizados para cualquiera que se suscriba al newsletter, así como también notificar a los vendedores información importante respecto a sus ventas.

Sistema de Autenticación

Una vez el proyecto esté en una fase de mayor madurez, se optará por externalizar el servicio de autenticación, para así simplificar los procesos de desarrollo y abaratar costos.

Para más detalles de cada componente, véase **Descripción de Componentes**.

3.2.6 Descripción de componentes

Tabla 3.1: Descripción de Componente: Base de Datos SQL

Descripción General	Según la definición de la organización de las tablas y relaciones, permite el almacenamiento de datos de transacciones, usuarios, carritos de compra, etc.
Interacción	Backend
Implementación	Se configura sobre una instancia de PostgreSQL, que se ejecuta dentro de un contenedor Docker.
Elementos Tecnológicos	En producción, es almacenado en la Nube de Amazon Web Services.
Funciones	Almacenar los datos relacionales de la aplicación. Interpretar y responder a solicitudes realizadas por el Backend.

Al decidir priorizar un mínimo producto viable, los componentes “Sistema de Autenticación”, “Repositorios”, “Sistema Email”, “Sistemas de Pago” y “Sistemas de Envío” no se desarrollaron en este proceso.

3.2.7 Análisis de tecnologías

En función de los componentes descritos y sus características, se plantea una serie de puntos relacionados con la infraestructura y las herramientas a utilizar para implementar el MPV. Esta sección comenta distintas alternativas y selecciona según las necesidades del proyecto y las capacidades del equipo de trabajo.

Tabla 3.2: Descripción de Componente: Backend

Descripción General	Sistema que contiene la lógica bajo la que el Marketplace va a operar.
Interacción	Base de Datos, Frontend, Sistema de Autenticación, Repositorios, Sistema de Pagos, Sistema Email
Implementación	El backend es desarrollado con tecnologías y bibliotecas basadas en Node.js. Algunas bibliotecas: ■ NestJS: Definir la arquitectura de la aplicación ■ Passport: Autenticación de usuarios. ■ Prisma: Capa de abstracción sobre la base de datos. ■ Swagger: Documentación de API REST.
Elementos Tecnológicos	En producción, es levantado en AWS sobre un contenedor Docker basado en una imagen que contiene las dependencias básicas de Node.js.
Funciones	En producción, es almacenado en la Nube de Amazon Web Services. Ejecuta la lógica de negocio del sistema, que es accedida mediante una API REST

Tabla 3.3: Descripción de Componente: Backoffice

Descripción General	Sistema frontend que permite la gestión de las tiendas.
Interacción	Backend, Aplicación Web
Implementación	Se configura un proyecto de frontend SPA el cual es levantado en un servidor Cloud
Elementos Tecnológicos	Amazon Web Service, Vue3.
Funciones	■ Gestionar la creación de usuarios vendedores y sus tiendas ■ Administrar la creación de productos ■ Gestionar las órdenes de las ventas ■ Contactarse con un comprador

Tabla 3.4: Descripción de Componente: Marketplace

Descripción General	Plataforma web que se encarga de exhibir los productos, conducir al comprador en el flujo de compra a través de múltiples interfaces.
Interacción	Backend, Aplicación Web
Implementación	Se configura un proyecto de Frontend SPA el cual es levantado en un servidor Cloud. Se habilitan características de renderizado por servidor para mejorar las vistas del e-commerce
Elementos Tecnológicos	Amazon Web Service, Vue 2, Nuxt 2
Funciones	■ Gestionar la compra de un producto ■ Administrar las cuentas de compradores ■ Gestionar el pedido de un producto ■ Generar reseñas de compras ■ Contactarse con un vendedor

Infraestructura Cloud

Se decide que el e-commerce Madeby se despliegue en la nube. Existen varias plataformas para alojar aplicaciones en internet evitando los costos de instalación y mantenimiento de infraestructura “On-Premise”. Entre los proveedores existentes se destacan:

- **Amazon Web Services (AWS):** Tiene una alta disponibilidad a nivel global, y ofrece un ecosistema muy amplio de servicios e integración con soluciones empresariales. Es el proveedor más maduro y con mayor cuota de mercado.
- **Google Cloud Platform (GCP):** Si bien ofrece servicios para una amplia variedad de aplicaciones, se especializa en el análisis de datos, el aprendizaje automático y el Big Data.
- **Microsoft Azure:** Ofrece una buena integración con los servicios empresariales ofrecidos por Microsoft como Windows Server, SQL Server y Office 365, y planes que aplican descuentos en las licencias.

Para alojar el proyecto se opta por utilizar los servicios que ofrece **Amazon Web Services (AWS)** dado su alto estándar de disponibilidad, escalabilidad elástica, gama de servicios y precios que se adaptan al tráfico del sitio.

Prototipos Visuales

Se utiliza la plataforma **Figma**, ya que ofrece un sistema de colaboración en tiempo real que permite al equipo trabajar de forma continua, junto con un sistema de prototipado interactivo, reutilización de componentes y una

amplia gama de integraciones como sistemas de íconos. Este conjunto de ventajas permite acelerar la creación de los prototipos a diseñar.

Framework Frontend

Con el fin de implementar una plataforma Marketplace que cuente con características como la personalización de una tienda sobre otra, se opta por utilizar el framework Nuxt basado en Vue3.

Vue3 es un framework de javascript para crear interfaces de usuario. Al tener un sistema de renderizado basado en componentes permite mayores velocidades en términos de rendimiento para un sistema E-Commerce con personalizaciones en gran escala. También se debe destacar su comunidad la cual está constantemente mejorando el framework añadiendo nuevas características y versiones. Finalmente, el peso del compilado de dependencias es menor al de ReactJs, la competencia directa, lo cual puede resultar en una carga más rápida de la aplicación, dando una mejor experiencia a los usuarios.

Nuxt se basa en Vue y extiende sus funcionalidades integrando características necesarias en cada proyecto. Como puede ser la integración de sistemas como la gestión de rutas, una estructura de proyecto simple de manejar y de gestionar, ideal para proyectos que manejan una gran variedad de componentes.

Se destaca su funcionalidad de SSR (Server Side Rendering), ya que es fundamental para situaciones en las que se deba cargar algo de gran tamaño. Al ser el servidor el que renderice la página y no el cliente, se acelera la velocidad inicial del sitio y mejora el SEO al proporcionar contenido direccionable por los motores de búsqueda. Esto quiere decir que los motores de búsqueda como Google darán una mayor puntuación al sitio, subiéndolo en categoría de recomendación.

Framework Backend

La implementación de una API REST puede realizarse con una gran cantidad de frameworks y bibliotecas en distintos lenguajes de programación.

- **ExpressJS (JavaScript):** Biblioteca de JavaScript minimalista y flexible, lo que permite utilizarla en muchos casos de uso. No cuenta con una gran cantidad de abstracciones porque busca ser simple.
- **NestJS (JavaScript, TypeScript):** Se basa en la modularidad por medio de controladores, servicios, módulos y el uso de la inyección de dependencias. Insiste en el uso de arquitecturas limpias y los principios SOLID de diseño de software. Tiene soporte para microservicios, aplicaciones de línea de comandos y al estar basado en TypeScript, permite usar tipado estático para detectar algunos errores tiempo de compilación.
- **Spring Boot (Java):** Basado en el patrón MVC, es similar a NestJS en cuanto a las abstracciones que ofrece y la inyección de dependencias. Es de alto rendimiento y es muy popular en entornos empresariales.

Tiene una alta curva de aprendizaje al estar basado en Java, pero cuenta con una amplia comunidad y extensa documentación.

- **Laravel (PHP)**: Utiliza el patrón MVC para crear aplicaciones robustas. Existe bastante documentación y una comunidad amplia dado que PHP es uno de los lenguajes más utilizados en el ámbito web. Integra una gran cantidad de herramientas, como EloquentORM para interactuar con la base de datos y Blade para las vistas.

Se decide utilizar **NestJs** por el uso de tipado estático en un entorno fácil de aprender como lo es JavaScript, ofrecer características que permitan escalar en caso de que la aplicación lo requiera e incluye módulos para la validación de peticiones, caché y autenticación.

Sistema de contenedores

Un contenedor es un proceso que ejecuta una aplicación o servicio y se encuentra encapsulado y aislado del resto del sistema, compartiendo una cantidad limitada de recursos con el anfitrión. Es utilizado en entornos cloud porque simplifica el despliegue de aplicaciones, es más ligero que la virtualización tradicional y da cierto grado de garantía respecto a la reproducibilidad del sistema.

- **Docker** es la herramienta más popular y estable existente para trabajar con contenedores. La documentación es extensa gracias a ser el líder del mercado. Ciertas características no son de código abierto y requieren suscripción en determinados casos.
- **Podman** es una alternativa a Docker desarrollada por Red Hat. Su principal característica es que no utiliza la arquitectura cliente y servidor, sino que es un proceso que se ejecuta en el espacio del usuario. Se centra en la seguridad y cumplir el estándar OCI (Open Container Initiative).

Dada la amplia adopción de **Docker** en la comunidad de desarrollo web, se decide utilizar esta alternativa. Las imágenes generadas al compilar los componentes se almacenan en un registro de artefactos privado. Algunas alternativas existentes son **Harbor** en caso de ser conveniente desplegarlo dentro del VPS o **Amazon Elastic Container Registry** (ECR) si no.

La orquestación de los contenedores puede realizarse por medio de Kubernetes o Docker Swarm y se decidirá en una etapa posterior. En la etapa de MPV se utilizará Docker Compose para levantar la plataforma y los servicios de bases de datos.

Bases de Datos

Los módulos del sistema E-Commerce propuesto tienen distintos requisitos a la hora de gestionar los datos, por ejemplo, el tiempo de lectura. Para ello se utiliza una base de datos relacional y una no relacional, con el fin de cumplir con requerimientos de confiabilidad y rendimiento específicos:

Base de datos relacional

La información de la aplicación, como los registros de las tiendas o información de los usuarios, se almacena usualmente en tablas estructuradas dentro de bases de datos relacionales.

- **MySQL**: Sistema de gestión de base de datos relacional de fácil uso, alto rendimiento y una de las más populares para aplicaciones web y móviles. No posee características avanzadas como PostgreSQL y tiene dificultades para el escalamiento horizontal. Al estar respaldado por Oracle, tiene popularidad dentro de la industria corporativa.
- **MariaDB**: Surge como una bifurcación del código fuente de MySQL luego de ser adquirido por Oracle Corporation. Busca ser completamente compatible con MySQL garantizando una migración sin problemas. Además, implementa distintos motores de almacenamiento que agilizan la escalabilidad horizontal o vertical y pueden mejorar la disponibilidad.
- **PostgreSQL**: La alternativa más utilizada a nivel empresarial porque posee funcionalidades avanzadas en comparación a MySQL o MariaDB. Ha sido utilizado en aplicaciones de alto rendimiento y seguridad en reiteradas ocasiones.

Para un sitio marketplace donde la información de las tiendas y las transacciones deben tener un alto estándar de calidad, integridad y consistencia, el uso de PostgreSQL responde adecuadamente a estos requerimientos.

Base de datos no relacional

Se utilizan en casos donde no se requiere una estructura en los datos, o el esquema puede ser flexible. También si la velocidad de consultas debe ser rápida.

- **MongoDB**: Base de datos orientada a documentos en que cada dato se almacena en formato JSON binario. Es la alternativa más popular dentro de las NoSQL. El escalamiento horizontal es sencillo de configurar y es muy eficiente en la lectura de datos.
- **Apache Cassandra**: Orientada a columnas, se desarrolla pensando en la confiabilidad, rendimiento y escalabilidad al trabajar con grandes volúmenes de datos. Cuenta con mecanismos de redundancia para evitar el único punto de falla.

Se escoge **MongoDB** como base de datos no relacional al ser una alternativa popular, con una comunidad amplia y soporte en varios módulos de mapeo objeto-relacional.

Documentación de API

Las API REST se documentan mayoritariamente con el estándar **OpenAPI**. El framework **NestJS** permite seguir el flujo de diseño “Code-First”. Cuenta con un módulo que integra la herramienta **Swagger** para generar

documentación por medio de anotaciones en el código fuente y **Swagger UI** que produce una página interactiva que describe la API y permite evaluar las rutas.

Capítulo 4

Implementación de la Solución

4.1 Diseño de interfaz web del backend

La API de *Madeby* es la capa de acceso que sirve para disponibilizar los recursos de la plataforma a los usuarios que ocupen el frontend de Marketplace o el de Backoffice. Dado que no se toma en consideración el uso público de la API, es un contrato al que solo los desarrolladores del backend y frontend deben atenerse y si bien no es necesario publicar abiertamente la documentación, es importante especificarla de manera clara y útil.

Esta sección se discuten las alternativas y decisiones tomadas en cuanto a puntos clave de acuerdo, como estrategias de versionado de la API y esquemas de autenticación y autorización.

4.1.1 Esquema de seguridad

Madeby es una plataforma que interactúa con una gran cantidad de usuarios, cada uno categorizable como vendedor, comprador o administrador. Debe poseer un mecanismo de acceso que permita asegurar la identidad de cada uno y limitar su experiencia de usuario según corresponda.

Hay 4 criterios en la seguridad web que deben ser cubiertos:

- **Confidencialidad:** Mantener información privada al almacenar y al transmitir
- **Integridad:** Identificar y prevenir alteraciones en el contenido de los mensajes
- **Identidad:** Identificar que las partes sean quienes indican ser.
- **Confianza:** Autorizar a una parte para interactuar de forma definida

El uso de HTTPS provee de integridad y confidencialidad las interacciones a través de internet al reemplazar el protocolo TCP por TLS, o Transport Layer Security.

El esquema de autenticación usado para asegurar la identidad del usuario se basa en OAuth 2.0 y el uso de tokens de acceso generados mediante usuario y contraseña, o utilizando servicios de identidad de terceros como Google o Facebook. En cada consulta deben ser anotados en la cabecera `Authorization: bearer <access_token>`. El formato de los tokens es JWT, lo que puede tener beneficios en caso de migrar a arquitecturas web distribuidas como microservicios.

En el caso de la autorización, se ocupa control de acceso basado en roles, o RBAC, para limitar el acceso a la funcionalidad de algunos recursos. El rol específico del usuario es almacenado en el token.

Por medio de HTTPS, OAuth 2.0 y RBAC se satisfacen los 4 criterios de seguridad web considerados necesarios en esta etapa.

Implementación de Integridad y Confidencialidad

Se puede integrar el uso de HTTPS por medio de la configuración del web server nginx con credenciales obtenidas en sitios como Let's Encrypt. Se delega a las siguientes iteraciones del proyecto para enfocarse en la funcionalidad.

Implementación de Autenticación

Para implementar la autenticación de los usuarios se utiliza el módulo Passport. Es utilizado para definir y ocupar “estrategias” de autenticación dentro de aplicaciones de Node.js, muchas de las cuales ya están implementadas y requieren una configuración mínima. Este proyecto utiliza las siguientes estrategias:

- `passport-jwt` para permitir la autenticación descentralizada ocupando tokens y una llave privada
- `passport-local` para que el usuario pueda ocupar una contraseña para generar los tokens
- `passport-google-oauth20` para permitir la autenticación por medio de google

Las estrategias de `passport.js` se implementan como middleware y se configuran usualmente registrando una función `validate` usada como callback. De esta manera el desarrollador puede inyectar la lógica necesaria para la validación de la solicitud, tal como el acceso a los datos, o el uso de claves secretas específicas a la aplicación. El diagrama 4.1 muestra la comunicación entre los módulos de la aplicación al momento de autenticar a un usuario con usuario y contraseña. Esto solamente se utiliza para la generación de tokens JWT.

Existen dos tipos de tokens. Los **tokens de acceso** son utilizados para acceder a recursos del e-commerce accesibles solo por usuarios registrados como vendedores o compradores. El diagrama 4.2 indica como se valida el token sin la necesidad de acceder a datos u otros módulos del sistema, únicamente necesita verificar que la clave privada JWT SECRET sea la que firma el token.

Por seguridad, al token de acceso se le asigna una vida corta en el rango de los minutos. Esto hace necesario el uso de un **token de refresco** para generar otro token de acceso. El diagrama 4.3 muestra el proceso, que requiere

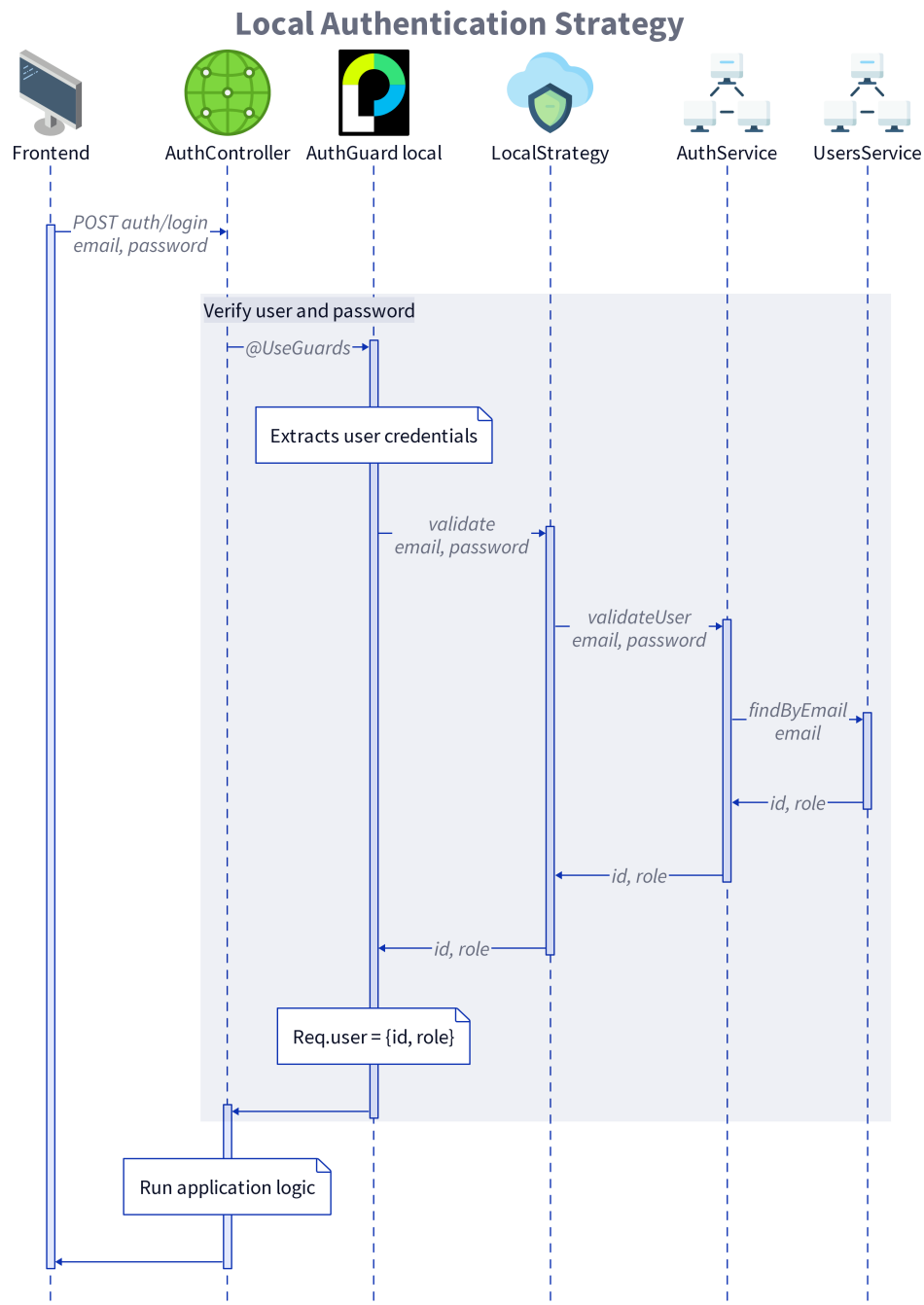


Figura 4.1: Diagrama de secuencia de autorización con estrategia “local”

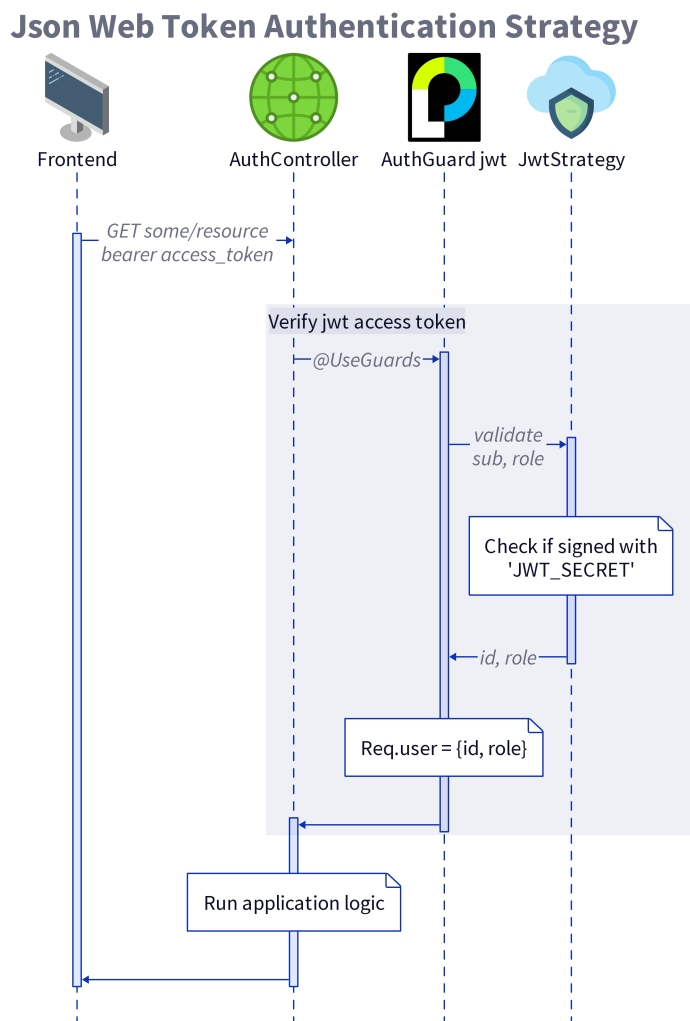


Figura 4.2: Diagrama de secuencia de autorización con estrategia “jwt”

acceder a datos del sistema para comprobar que el token sigue siendo válido. Este token solo debe ser utilizado una vez, y también puede ser revocable por los administradores en caso de detectar comportamiento anómalo. Los tokens de acceso no pueden ser revocados, sin embargo, operaciones críticas pueden consultar el estado del token de refresco antes de ejecutarse a modo de protección.

OAuth2 es un estándar que puede tener implementaciones levemente diferentes por los distintos proveedores de identidad, como Google, Facebook o Twitter. Passport delega estas complejidades a paquetes predefinidos como `passport-linkedin-oauth2`, `passport-bitbucket-oauth20` y `passport-google-oauth20`.

El diagrama de secuencia 4.4 muestra el proceso de autenticación con los servicios de Google, lo que requiere de registrar la aplicación a través de la consola de Google Cloud Platform y anotar credenciales como el identificador de cliente de google, una llave secreta y una dirección URI utilizada por google para entregar las

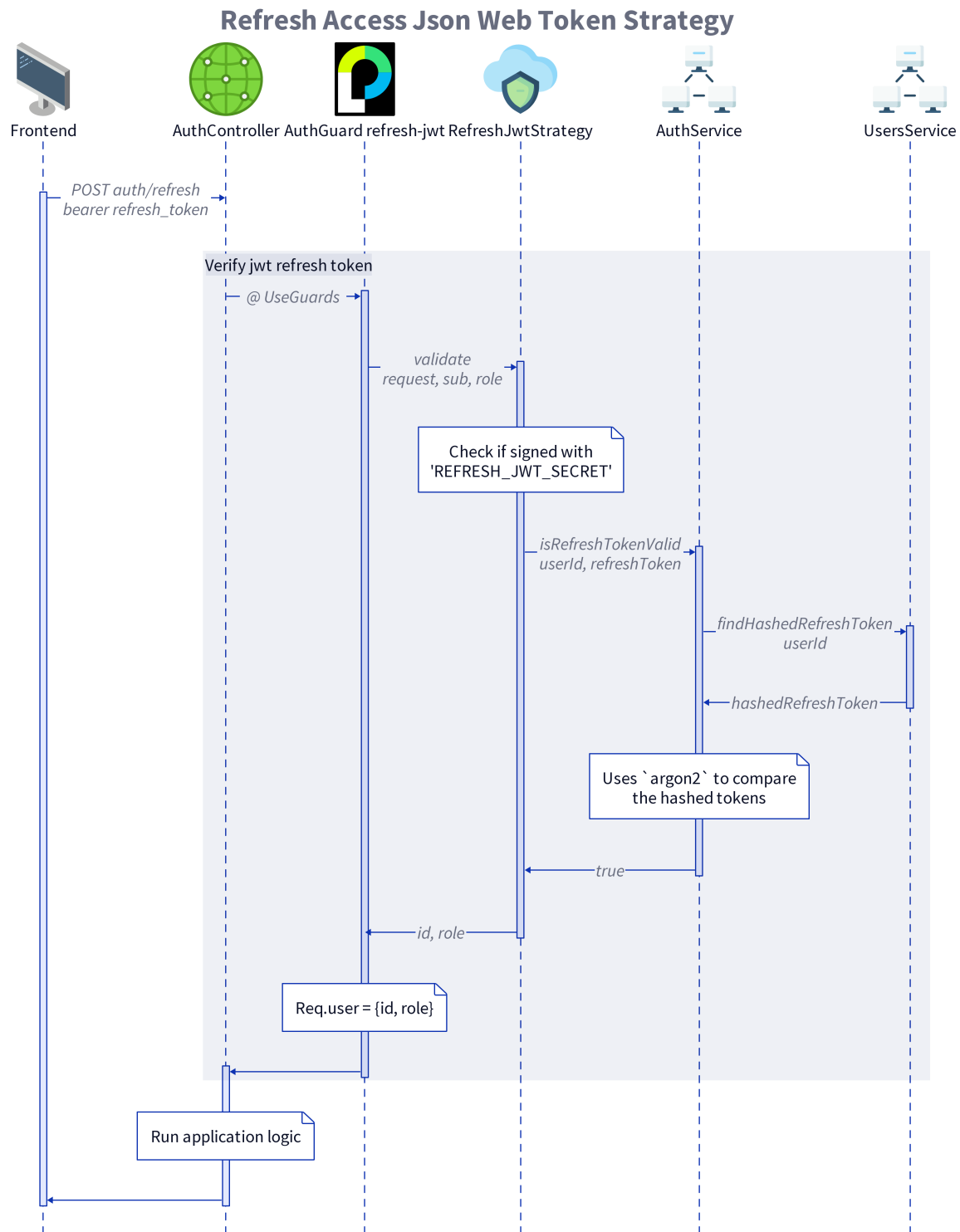


Figura 4.3: Diagrama de secuencia de validación de token de refresco con estrategia “refresh-jwt”

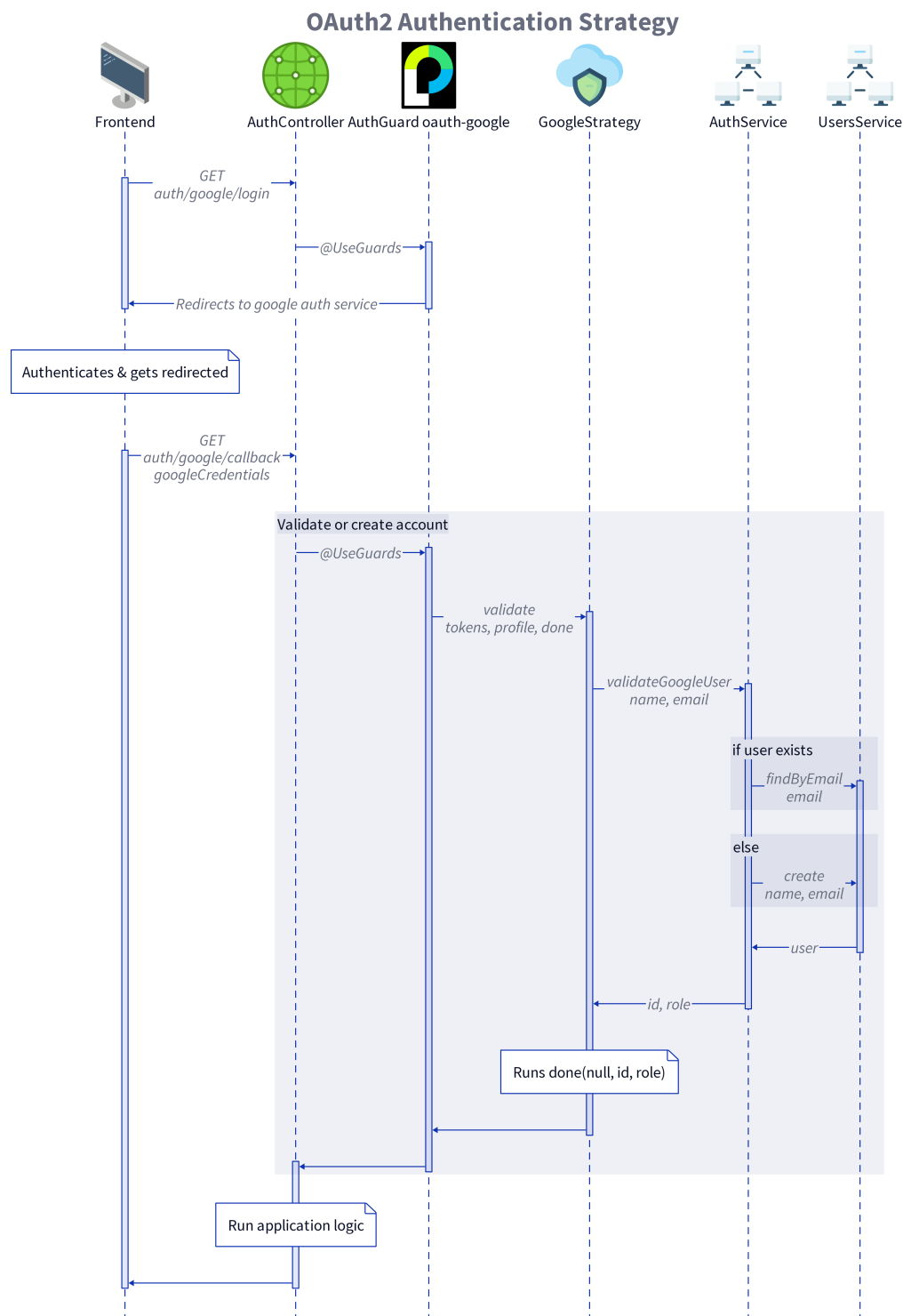


Figura 4.4: Diagrama de secuencia de autorización con estrategia “google-jwt”

credenciales del usuario en caso de autenticarse correctamente. Esta estrategia, al igual que “local”, se utiliza para la generación de tokens JWT.

Implementación de Autorización

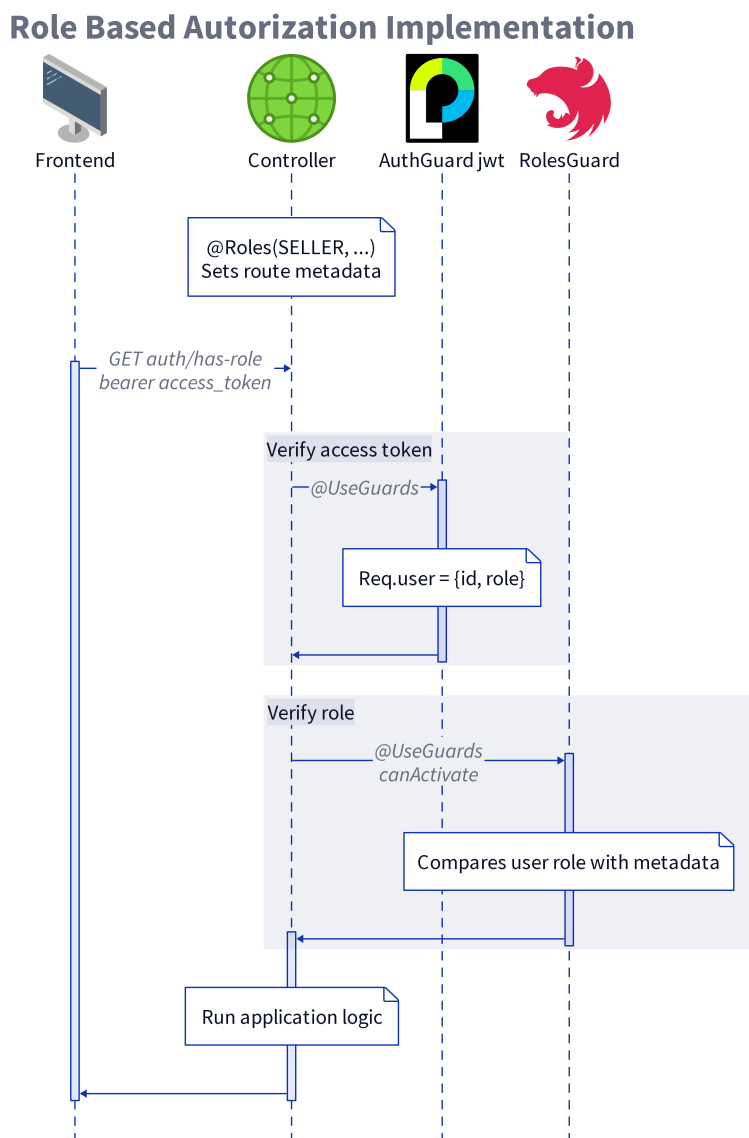


Figura 4.5: Diagrama de secuencia de verificación del rol del usuario

El framework **NestJS** no cuenta con un módulo específico para definir la autenticación basada en roles, sin embargo, ofrece abstracciones que permiten implementar el middleware necesario. El decorador `@Roles` () registra los roles admitidos por una ruta dentro de los metadatos del entorno de ejecución. Mientras que la clase `RolesGuard` implementa la interfaz `CanActivate` usada para filtrar usuarios cuyo rol no coincide con los permitidos.

El flujo de autenticación y autorización se muestra en la figura 4.5.

4.1.2 Manejo de versiones

Todas las aplicaciones cambian a lo largo de su tiempo de operación, ya sea por la necesidad de modificar la representación de un recurso, la integración de nuevas funcionalidades por cambios en el mercado o la incursión en nuevos nichos de negocio.

Es importante establecer de manera clara el mecanismo con el que se gestionan las versiones de la API porque los clientes, tanto el equipo de frontend como sus usuarios, pueden verse afectados negativamente si se rompe la compatibilidad de manera imprevista. Las principales maneras de indicar la versión de una API son comparadas en la tabla comparativa 4.1.

Ubicación	Ejemplo	Ventajas	Desventajas
Por URI	/v1/customer	• Ampliamente utilizado • Visible y fácil de entender • Cacheable	• No sigue principios REST • URIs acoplados a versión específica
Subdominio	v1.api.example.com/customer	• Visible y fácil de entender • Cacheable	• Requiere gestión de infraestructura
Parámetro de Consulta	/customer?version=v1	• Visible y fácil de entender	• Agrega un query a todas las consultas • Posible conflicto con filtrado, paginación
Cabecera Personalizada	/customer X-API-Version: 1	• Mantiene URI limpia • No requiere análisis de cabecera	• Difícil de evaluar en navegador
Tipo de medio	/customer Accept: application/vnd.madeby.v1+json	• Mantiene URI limpia • Conforme a REST y negociación de contenido	• Difícil de evaluar en navegador • Requiere análisis de cabecera • Poco intuitivo

Tabla 4.1: Comparativa de formas de especificar la versión

Implementación de selección de versiones

El framework **NestJS** admite versionado por URI, cabeceras personalizadas, por tipo de medios y, en caso de ser necesario, permite definir un esquema arbitrario. Sin embargo, integrar en el código fuente la selección de versión presenta la desventaja de aumentar su complejidad, puesto que se da mantenimiento a métodos compatibles con representaciones y operaciones que no necesariamente siguen siendo de valor al negocio, pero deben ser tomadas en consideración por los desarrolladores.

En esta aplicación, se decide versionar utilizando una **cabecera personalizada** y direccionar las peticiones por medio de un proxy reverso a instancias etiquetadas del servidor, como se muestra en el diagrama 4.6. Entre las ventajas de esta solución se destacan las siguientes:

- El uso de contenedores permite integrar el versionado del backend con el orquestador
- Se evita mantener rutas y métodos en el backend cuyo único propósito sea mantener la compatibilidad, pero requiere de un correcto etiquetado de las imágenes del backend en el registro privado

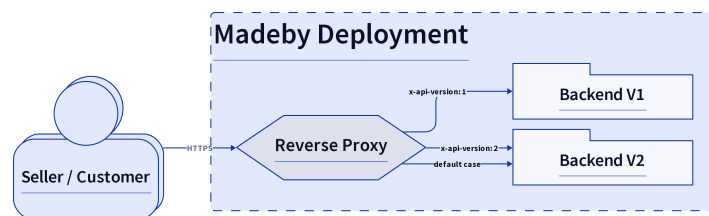


Figura 4.6: Implementación de versionado por cabecera personalizada

- Dado que no se considera disponibilizar la API a terceros, no es necesario notificar públicamente la estrategia de versionado
- Para cambios menores se pueden establecer periodos de evaluación redirigiendo una fracción del tráfico de las peticiones. Mientras que para cambios mayores el periodo de soporte coordina con los equipos encargados del frontend

4.1.3 Descripción de API HTTP

La especificación OpenAPI es un formato de descripción de API REST utilizado para declarar de que manera interactuar con una aplicación web de forma agnóstica a los detalles de implementación. Se utiliza para indicar endpoints y los verbos y parámetros que aceptan, métodos de autenticación y otra información relevante para los desarrolladores que deban ocuparla o implementarla.

Es una parte relevante de la documentación, puesto que su uso facilita el uso de la API por parte de terceras personas y puede ser analizada por otras herramientas para generar entornos de evaluación de API como Swagger UI. Existen 3 métodos de generación de estos archivos.

Generación Manual

Un documento OAS versión 3 (OpenAPI Specification) es un archivo JSON o YAML que describe aspectos de uso de la API como información sobre los servidores, licencias de uso, estructuras de los modelos de datos y los recursos disponibles con sus operaciones, parámetros y respuestas. Escribir manualmente este archivo es un proceso extenso y propenso a errores, por tal motivo se suele recurrir a generarlo a partir de otras fuentes.

Enfoque “Code-First”

El foco está en describir aquello que se está implementando o se ha implementado, dejando en un plano secundario la generación de documentación. Frameworks como **NestJS** cuentan con módulos permiten ocupar decoradores para denotar elementos de OpenAPI, funciones para generar el archivo OAS y rutas para disponer de un entorno de prueba de la API con Swagger-UI.

Durante el desarrollo del MPV se utiliza este enfoque para dar énfasis a la implementación.

Enfoque “API Design-First”

Depender enteramente de la implementación de un servicio web para documentar su interfaz reduce la flexibilidad de la API y dificulta el análisis desde una perspectiva amplia. Enfocarse primero en el diseño de la API antes de implementarla tiene una serie de ventajas:

- Establecer los modelos y operaciones disponibles agiliza las iteraciones y modificaciones al contrato
- Permite mejorar la consistencia de la API al reutilizar modelos de datos, respuestas y parámetros de consulta
- Evaluar la calidad y seguridad de la API por medio de herramientas que ocupan el archivo OAS

Cumplir con los puntos anteriores tiene un efecto positivo en la adopción de la API por parte de los clientes y orienta el trabajo de los desarrolladores. El motivo por el que no se aplicó en este proyecto es que durante el periodo de implementación de la memoria estas prácticas y herramientas no estaban difundidas o no existían.

4.2 Desarrollo por requisitos

El desarrollo de los servidores para el backend y el frontend se dirige por medio de la implementación de los requisitos funcionales. Cada requisito se describe y luego se modela con un diagrama de solución antes de explicar como se realiza la implementación, indicando las limitaciones en caso de presentarse alguna.

Con tal de simplificar la información, los diagramas omiten elementos relacionados con la autenticación o autorización de las peticiones a menos que sea parte explícita del alcance del requisito.

4.2.1 Requisito Funcional 1: Registro de usuarios

Un usuario, vendedor o comprador, puede realizar un registro básico con un correo electrónico y contraseña

Diagrama de Solución

Como se aprecia en el diagrama 4.8, un usuario no registrado que ingresa al sitio e inicia el proceso de registro puede hacerlo mediante un formulario o un sitio de terceros. Cuando se decide utilizar el formulario, el sistema (frontend y backend) verifica el formulario y registra correo electrónico y hash de la contraseña, entre otra información relevante. Si hay información inválida o errónea, se indicarán los campos correspondientes y se permitirá reintentar el proceso. Antes de finalizar y autorizar la cuenta, se solicita una confirmación por correo electrónico.

Cuando el usuario ya está registrado, podrá iniciar sesión con un formulario de ingreso o utilizado un servicio de terceros, como se muestra en la figura 4.7.

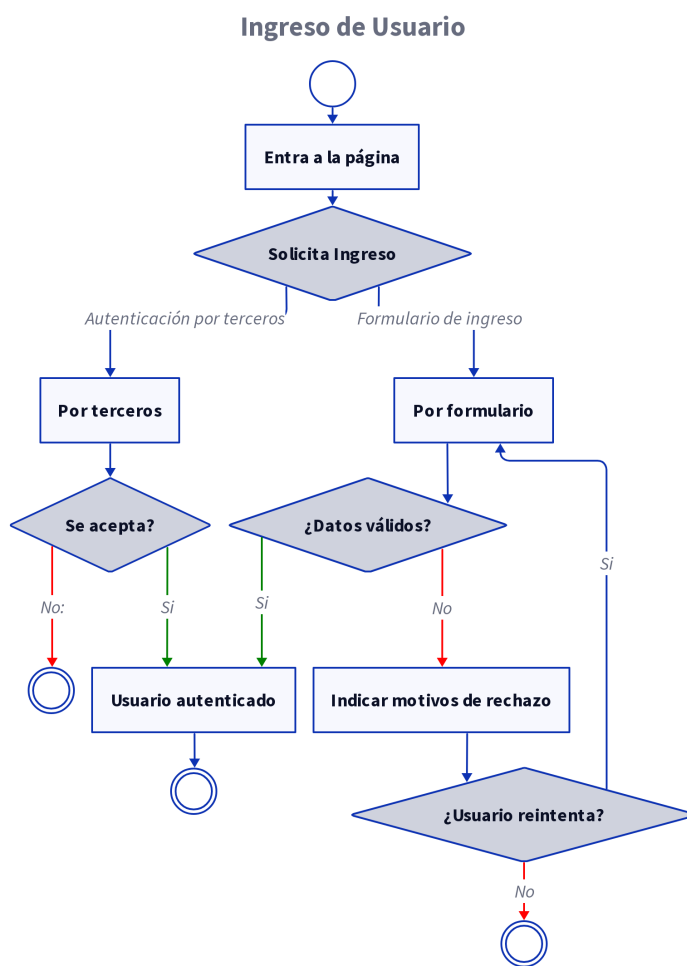


Figura 4.7: Diagrama de flujo de autenticación e ingreso

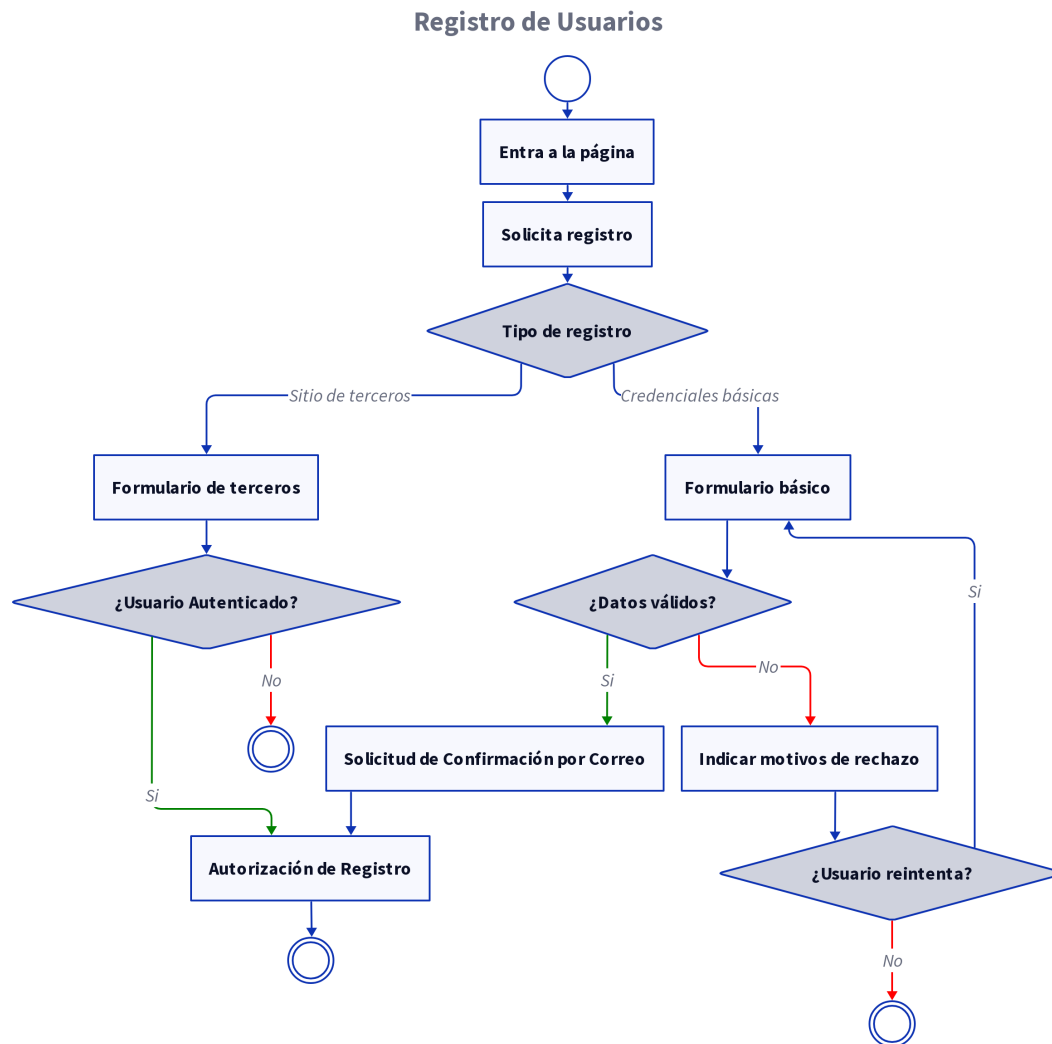
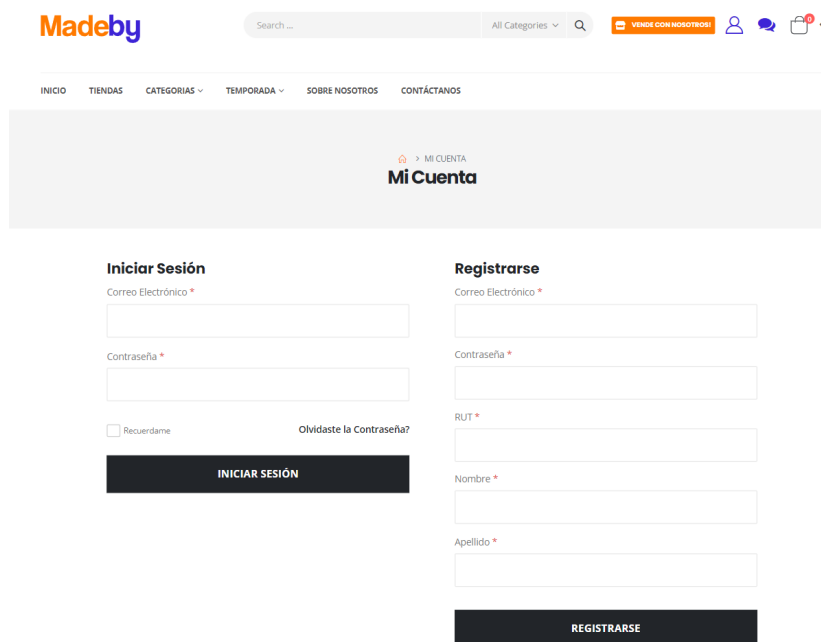


Figura 4.8: Diagrama de flujo de registro de usuarios

Formularios de registro

Si bien se considera que tanto comprador como vendedor son usuarios, el registro se realiza con formularios distintos porque en el caso del vendedor también se debe registrar una tienda. El formulario de registro y acceso a cuenta del comprador se muestra en 4.9 y solo requiere credenciales básicas e información del usuario como su nombre, apellido y rut. En el caso del vendedor, además de lo anterior se solicita información de registro de tienda, como se muestra en las capturas 4.10, 4.11 y 4.12. Los formularios incluyen mecanismos de validación sobre ciertos atributos para evitar problemas antes de comunicarse con el backend. Por ejemplo, verificar que el RUT sea válido o que el correo tenga el formato correcto.



The screenshot shows the Madeby website's user interface. At the top, there is a navigation bar with the Madeby logo, a search bar, and links for 'All Categories', 'VENIR CON NOSOTROS', and user icons. Below the navigation bar is a menu with links: INICIO, TIENDAS, CATEGORÍAS, TEMPORADA, SOBRE NOSOTROS, and CONTÁCTANOS. The main content area features a 'Mi Cuenta' section with a user icon and the text 'MI CUENTA' and 'Mi Cuenta'. Below this, there are two columns of forms. The left column is for 'Iniciar Sesión' (Login) and the right column is for 'Registrarse' (Register).

Iniciar Sesión

Correo Electrónico *

Contraseña *

☐ Recuérdame ☐ Olvidaste la Contraseña?

INICIAR SESIÓN

Registrarse

Correo Electrónico *

Contraseña *

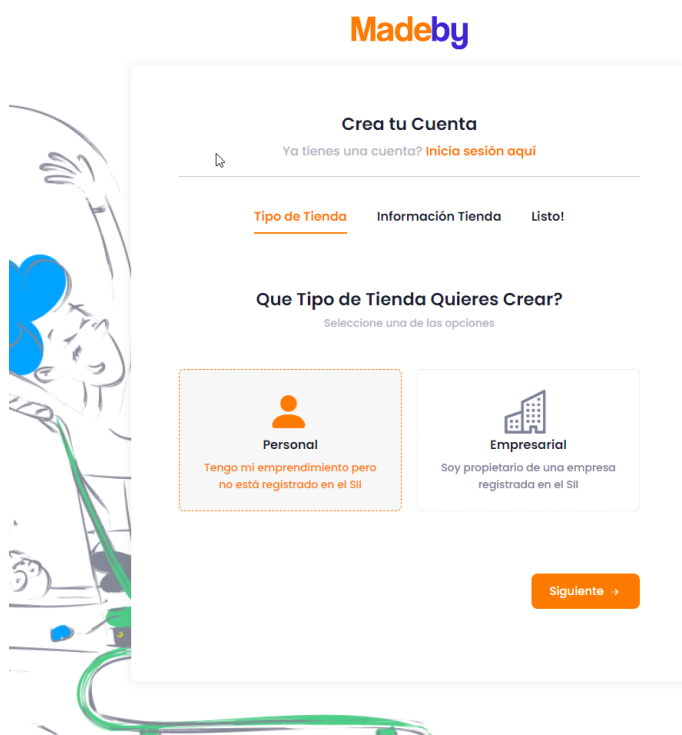
RUT *

Nombre *

Apellido *

REGISTRARSE

Figura 4.9: Registro de Comprador



The screenshot shows the Madeby website's 'Crea tu Cuenta' (Create your account) form. The form is titled 'Crea tu Cuenta' and has a link 'Ya tienes una cuenta? Inicia sesión aquí'. Below the title, there are three tabs: 'Tipo de Tienda', 'Información Tienda', and 'Listo!'. The 'Tipo de Tienda' tab is selected. The main heading is 'Que Tipo de Tienda Quieres Crear?' with the instruction 'Seleccione una de las opciones'. There are two options: 'Personal' and 'Empresarial'. The 'Personal' option is highlighted with a dashed orange border and includes the text 'Tengo mi emprendimiento pero no está registrado en el SII'. The 'Empresarial' option includes the text 'Soy propietario de una empresa registrada en el SII'. At the bottom right, there is a 'Sigüente >' button.

Crea tu Cuenta

Ya tienes una cuenta? [Inicia sesión aquí](#)

Tipo de Tienda Información Tienda Listo!

Que Tipo de Tienda Quieres Crear?

Seleccione una de las opciones

Personal

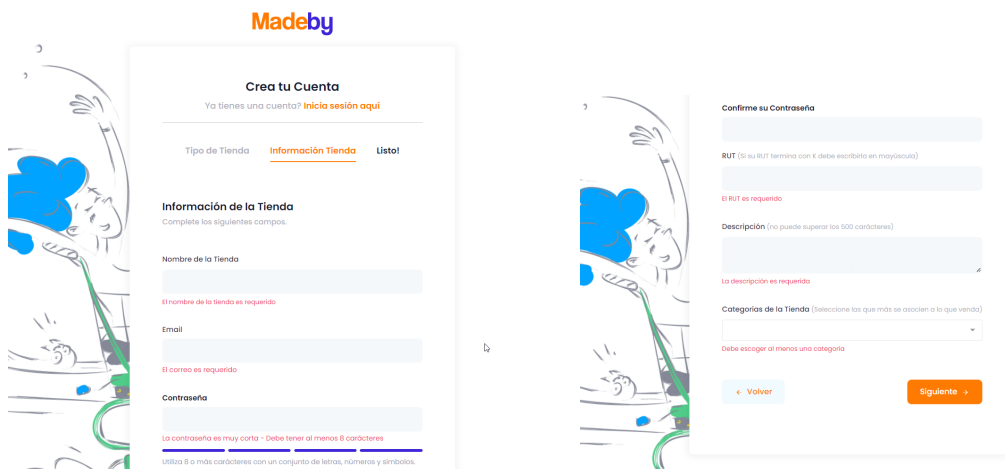
Tengo mi emprendimiento pero no está registrado en el SII

Empresarial

Soy propietario de una empresa registrada en el SII

Sigüente >

Figura 4.10: Registro de vendedor: Tipo de tienda



Madeby

Crea tu Cuenta
Ya tienes una cuenta? [Inicia sesión aquí](#)

Tipo de Tienda **Información Tienda** **Listo!**

Información de la Tienda
Complete los siguientes campos.

Nombre de la Tienda

El nombre de la tienda es requerido

Email

El correo es requerido

Contraseña

La contraseña es muy corta - Debe tener al menos 8 caracteres
Utiliza 8 o más caracteres con un conjunto de letras, números y símbolos.

RUT (Si su RUT termina con K debe escribirlo en mayúscula)

El RUT es requerido

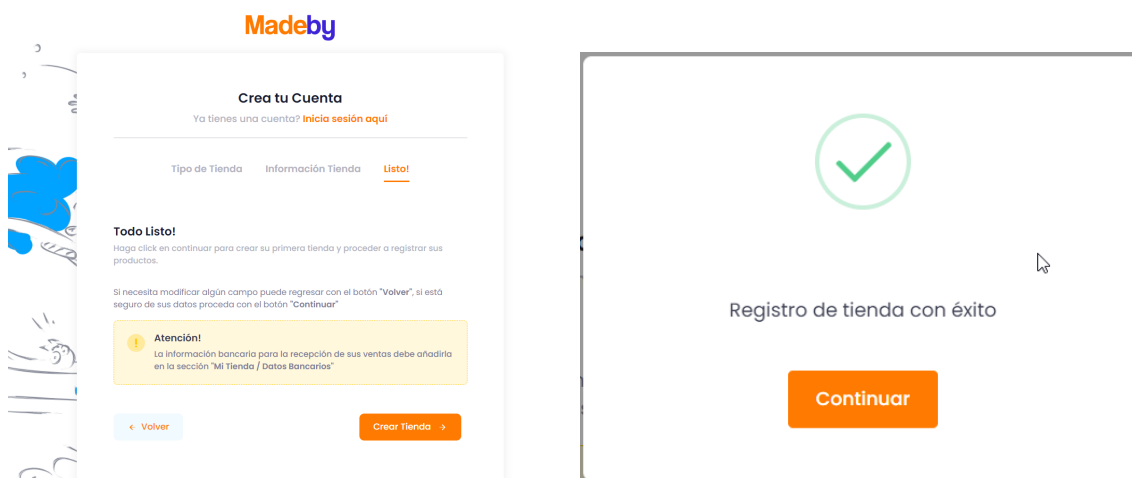
Descripción (no puede superar los 500 caracteres)

La descripción es requerida

Categorías de la Tienda (seleccione las que más se asocien a lo que vende)

Debe escoger al menos una categoría

[← Volver](#) [Siguinte →](#)

Figura 4.11: Registro de vendedor: Información y credenciales

Madeby

Crea tu Cuenta
Ya tienes una cuenta? [Inicia sesión aquí](#)

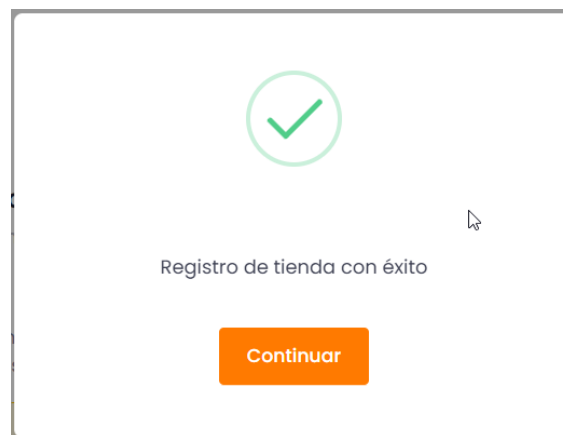
Tipo de Tienda Información Tienda **Listo!**

Todo Listo!
Haga click en continuar para crear su primera tienda y proceder a registrar sus productos.

Si necesita modificar algún campo puede regresar con el botón "Volver", si está seguro de sus datos proceda con el botón "Continuar"

Atención!
La información bancaria para la recepción de sus ventas debe añadirla en la sección "Mi Tienda / Datos Bancarios"

[← Volver](#) [Crear Tienda →](#)



Registro de tienda con éxito

[Continuar](#)

Figura 4.12: Registro de vendedor: Confirmación



Figura 4.13: Acceso a cuenta del vendedor

Ruta de acceso de usuario

Para obtener los tokens de acceso a los recursos del sitio, los usuarios cuentan con los formularios mostrados en las figuras 4.9 y 4.13. Luego de llenar el formulario, el frontend realiza la petición `POST auth/login`, lo que inicia el proceso descrito en la figura 4.14, en donde se utiliza la estrategia de autenticación local con usuario y contraseña para generar los tokens de acceso y refresco. Por seguridad, se registra el de refresco para restringirlo a una única instancia válida que puede ser anulada en caso de detectar comportamiento malicioso.

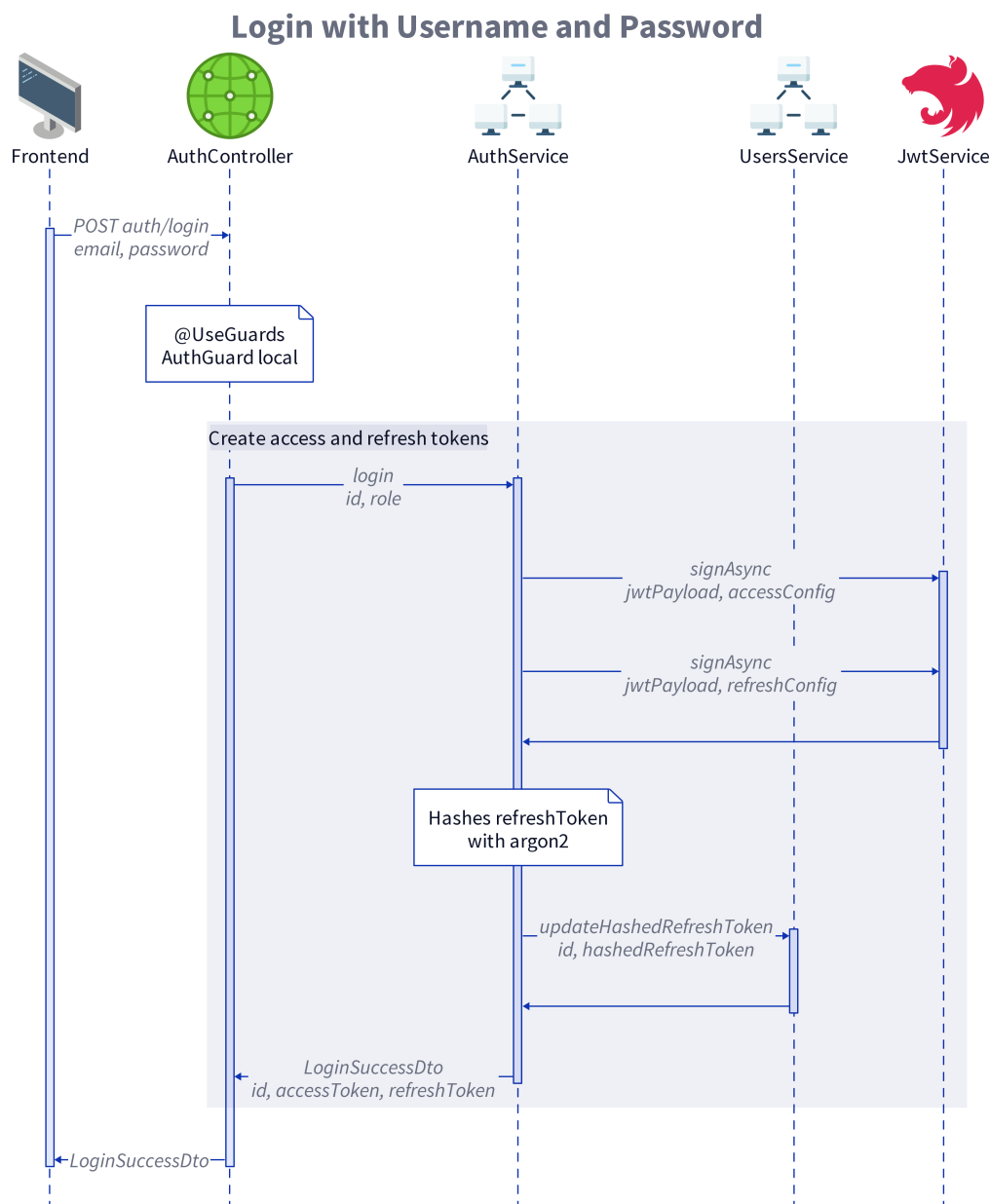
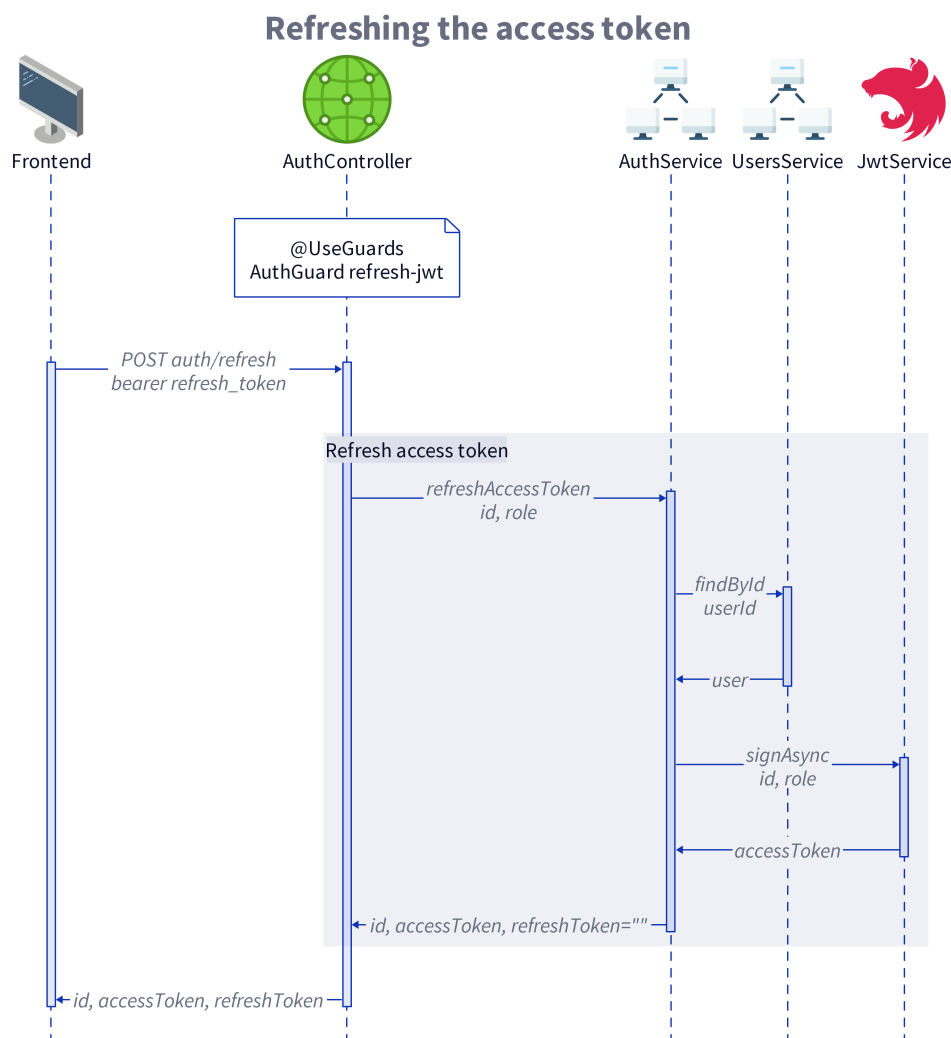


Figura 4.14: Secuencia de autenticación de usuario

**Figura 4.15:** Secuencia de refresco de tokens

Cuando el token de acceso expira, se debe iniciar el proceso para obtener un nuevo token con la ruta `POST auth/refresh` utilizando el token de refresco en la cabecera `Authorization`. El proceso se detalla en el diagrama 4.15. En caso de expirarse el token de refresco, el usuario se verá forzado a autenticarse nuevamente.

Limitaciones

Al implementar este requisito funcional se decide limitar su alcance en vista de cumplir con el mínimo producto viable. En particular, se omite la validación de cuenta por medio de correo electrónico y tampoco se considera información de sistema de pagos al momento de registrarse.

Durante la implementación se da cuenta de que mantener un sistema de autenticación y gestión de usuarios es un proceso crítico que requiere un esfuerzo considerable para ser realizado correctamente. Es posible que en

próximas iteraciones se decida utilizar únicamente servicios de terceros para verificar la identidad de los usuarios, por ejemplo **Auth0**, **Firebase Auth** o **AWS Cognito**, entre otros.

4.2.2 Requisito Funcional 2: Recuperación de contraseña

Un usuario puede recuperar su contraseña

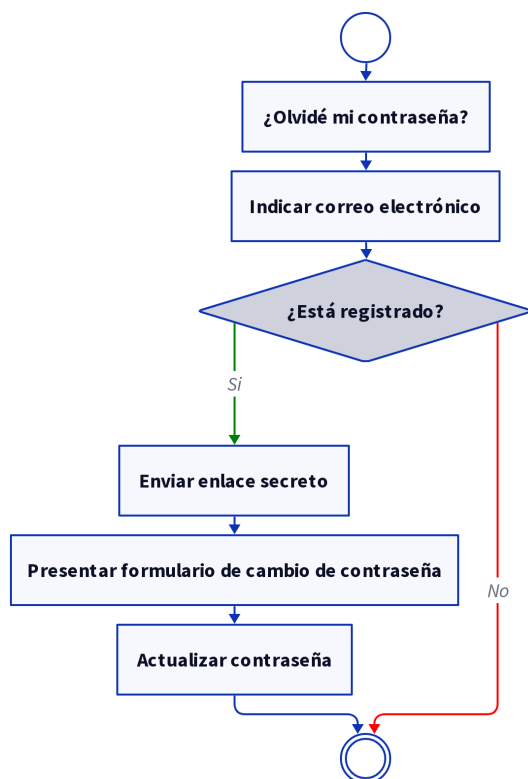
Diagrama de Solución

En caso de no recordar la contraseña el proceso para cambiarla a una conocida se muestra en el diagrama 4.16a. Por medio del correo electrónico, se envía un enlace de un solo uso con una clave secreta que permite acceder al formulario para cambiar la contraseña en caso de olvido. Usuarios registrados por medio de sitios de terceros no pueden modificar su contraseña.

Implementación

Dado que no se integra el uso de correos electrónicos en esta iteración, y para dar énfasis a funcionalidad de la lógica de negocio, se opta por omitir la implementación de este requisito. Sin embargo, se dispone de una vista en el frontend como se muestra en 4.16b.

Recuperar contraseña



(a) Diagrama de flujo de cambio de contraseña



(b) Formulario para cambiar la contraseña

4.2.3 Requisito Funcional 3: Autenticación por terceros

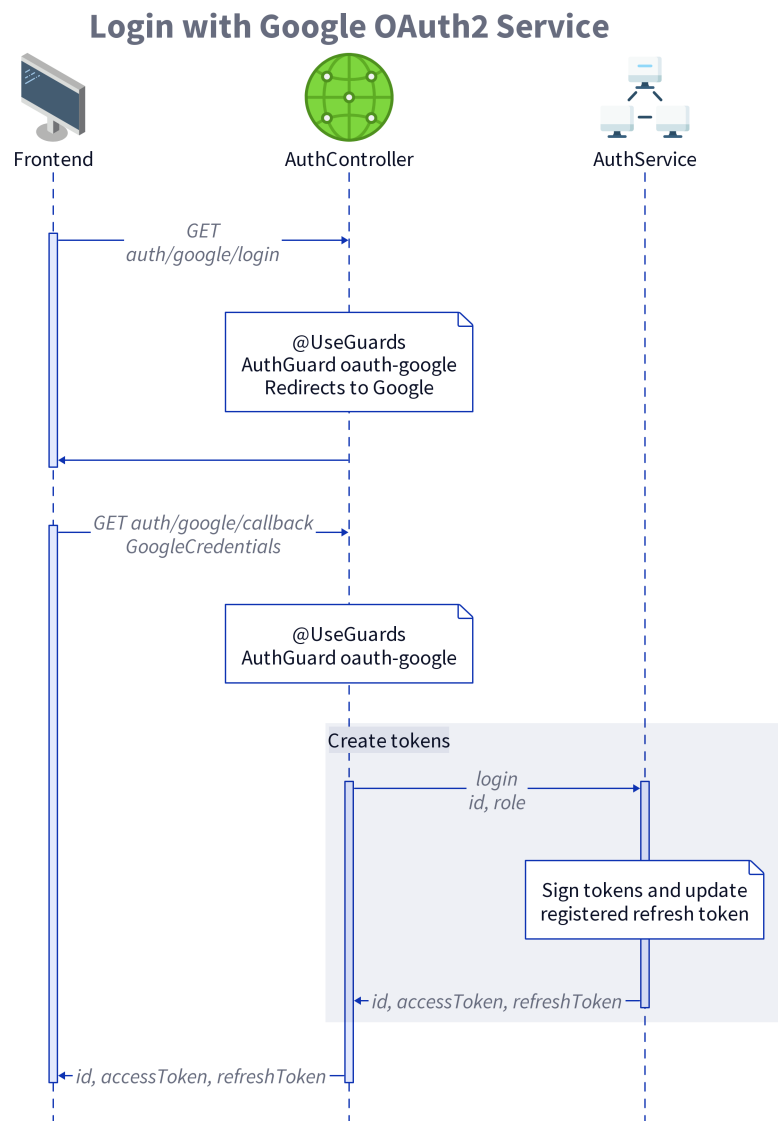
Un usuario puede autenticarse utilizando sitios de terceros

Diagrama de solución

El flujo de registro con autenticación de terceros se describe en el diagrama 4.8 del requisito funcional “Registro de Usuarios”. Al escoger el registro por formulario de terceros, se redirige a una vista perteneciente al sitio del proveedor de identidad y en caso de autenticarse correctamente los datos del usuario serán registrados en el sistema.

Implementación

Se integra la autenticación de terceros utilizando la estrategia OAuth 2.0 con Google por medio del módulo Passport y la estrategia `passport-google-oauth20` tal como se explica en la subsección 4.1.1. Aquel diagrama muestra como se realiza el registro del usuario en caso de no poseer una cuenta. Luego de que el proveedor de identidad confirme la identidad del usuario, el proceso de generación de tokens es el mismo que al autenticarse con usuario y contraseña, como se muestra en el diagrama 4.17.

**Figura 4.17:** Secuencia de autenticación de usuario OAuth2 con Google

4.2.4 Requisito Funcional 9: Gestión del producto

El vendedor puede publicar productos y editarlos

Diagrama de Solución

El administrador de la tienda puede agregar, modificar o eliminar sus productos, véase diagrama 4.18. Según sea el caso, tiene que seleccionar alguno de los productos, rellenar uno de los formularios y luego confirmar su operación. Si se está registrando o modificando un producto, el formulario contendrá la información de este, y mostrará una vista previa con el fin de confirmar que la información sea válida. Mientras que si se está eliminando un producto, el formulario es una confirmación para eliminarlo.

Dentro de la base de datos, no se recomienda eliminar el registro como tal, solo ocultarlo de la interfaz del vendedor y los compradores.

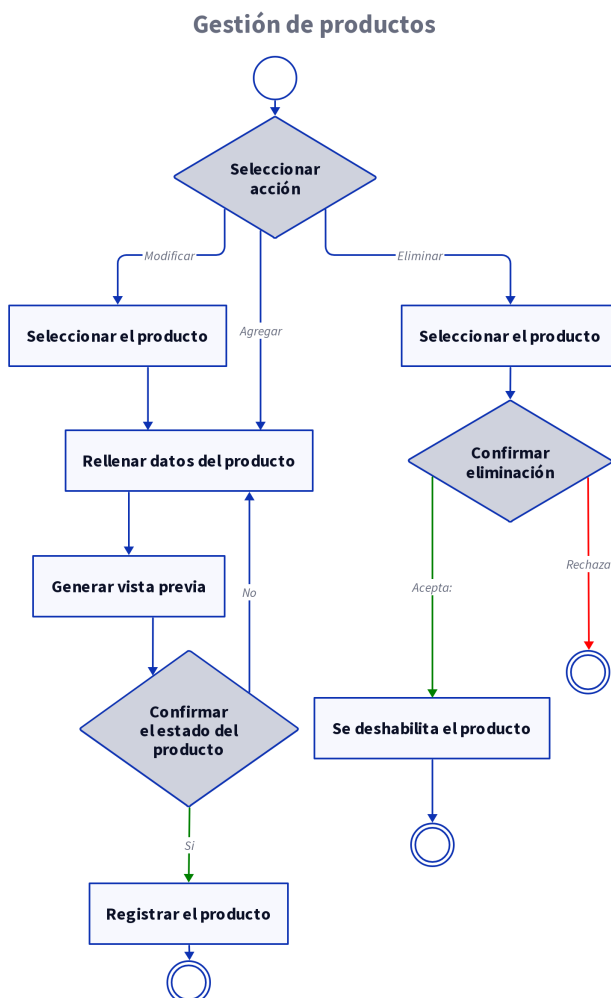
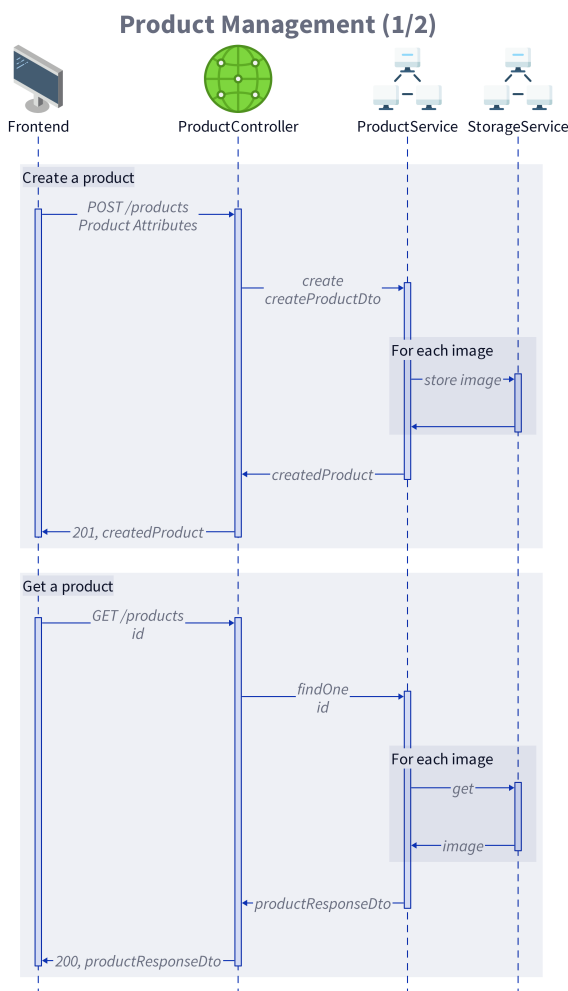
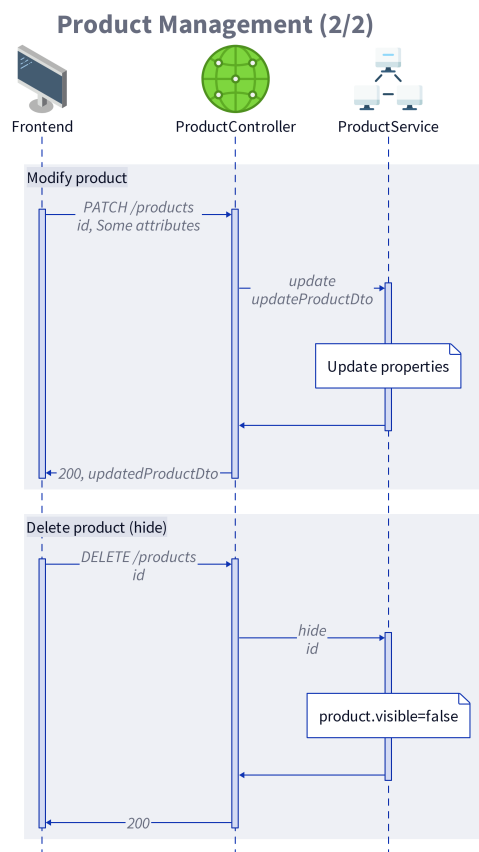


Figura 4.18: Diagrama de gestión de productos

(a) Diagrama de secuencia de métodos de creación y consulta de un producto



(b) Diagrama de secuencia de métodos de modificación y remoción de un producto



Implementación

Se implementa el módulo `products` que dispone de una interfaz CRUD en la ruta `/products`, tal como lo indican los diagramas 4.19a y 4.19b. Depende del módulo `storage` para dar persistencia a las imágenes de manera local dado que aún no se realiza la integración con un servicio de almacenamiento en la nube.

Formularios de gestión de productos

El proceso de registro de producto se puede ver en la sucesión de figuras 4.20, 4.21, 4.22 y 4.23. En esta última captura se muestran los botones para acceder a modificar y eliminar un producto.

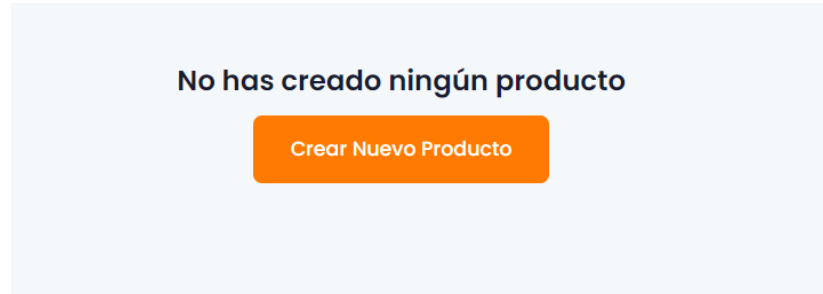


Figura 4.20: Tienda sin productos

Crear nuevo producto

Por favor rellene el siguiente formulario

Nombre del Producto *

Test

Categoría *

Jugantes x Deportes x

Tipo de producto *

Físico

Descripción *

Normal B I U        

Tipo de Producto

test
test 2

- Test
- Test
- Test

Imagen del Producto *



Añadir imagen


10.png

11.png

12.png

Precio *

\$ 50.000

Stock *

30

Registrar Producto

Figura 4.21: Interfaz de registro de producto

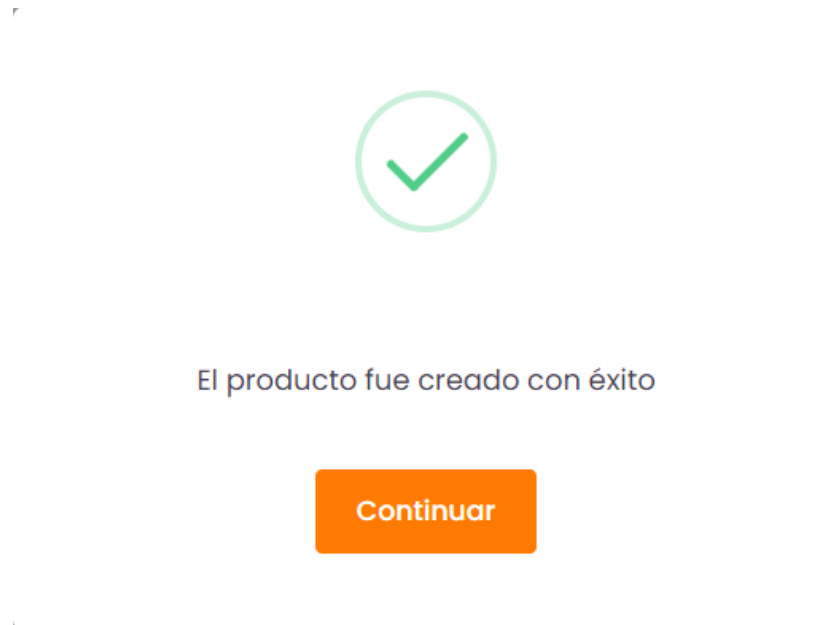


Figura 4.22: Registro exitoso de producto

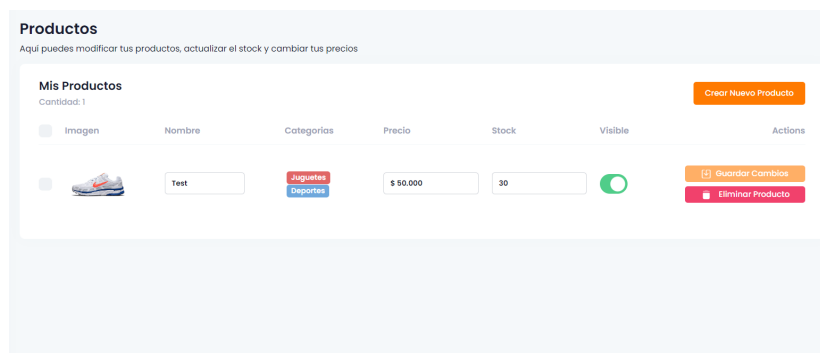


Figura 4.23: Lista de productos registrados

4.2.5 Requisito Funcional 10: Estilo de la tienda

El vendedor puede definir el estilo de su tienda con elementos como el logo, esquema de colores y categorías asociadas

Diagrama de Solución

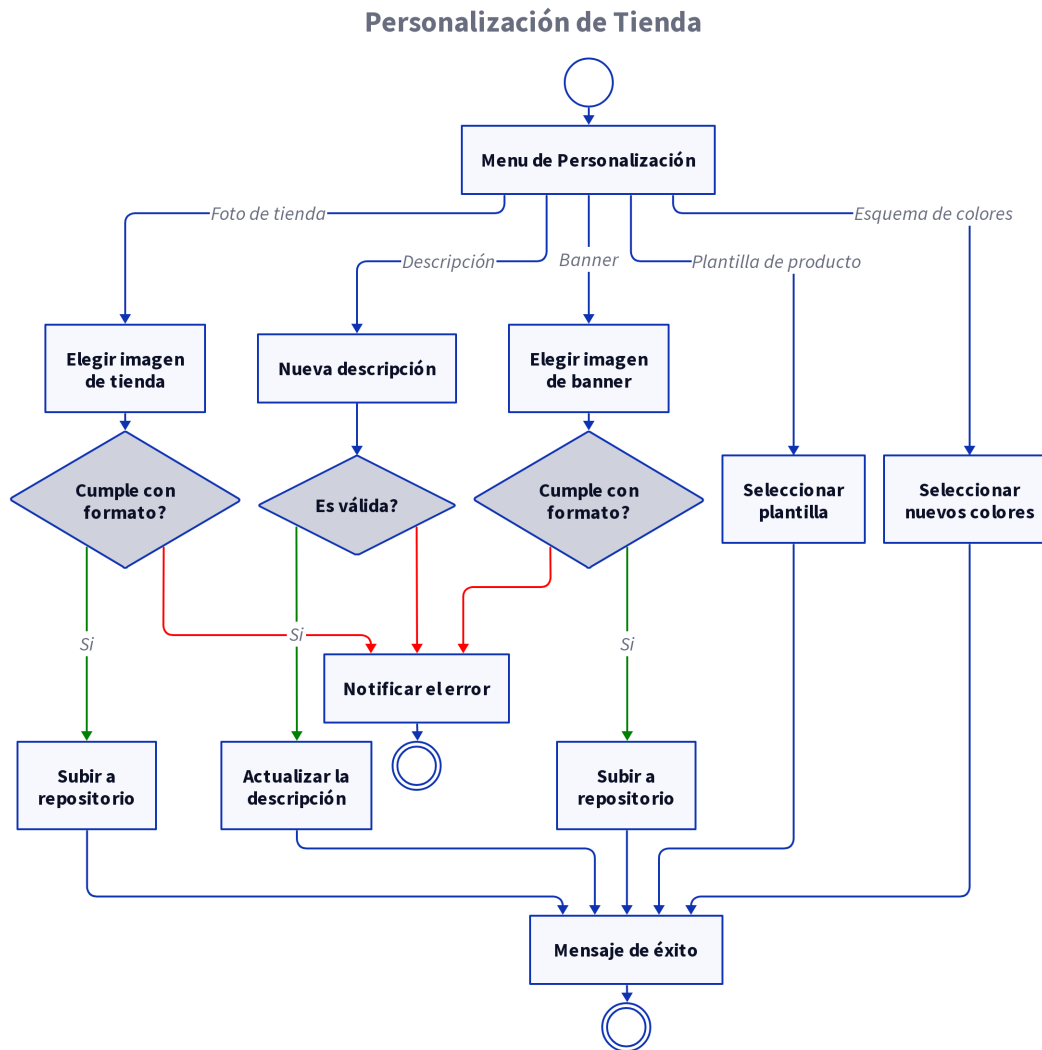


Figura 4.24: Diagrama de flujo sobre la configuración de la tienda y la vista de los productos

El proceso de personalización mostrado en el diagrama 4.24 se divide en 5 componentes que definen la estética de la tienda: una descripción; un logo; un banner; el esquema de colores y la plantilla de la tienda para mostrar los productos. La descripción se verifica para evitar ataques de texto y limitar su extensión, mientras que las imágenes de logo y banner deben tener un tamaño y dimensiones adecuadas. El esquema de colores cuenta con opciones para el color primario y secundario, el color de fondo y de texto. Finalmente, el sistema de plantillas

consiste en opciones que afectan el orden en que se disponen los elementos de la tienda como imágenes, la lista de productos, elementos destacados, entre otros.

Implementación

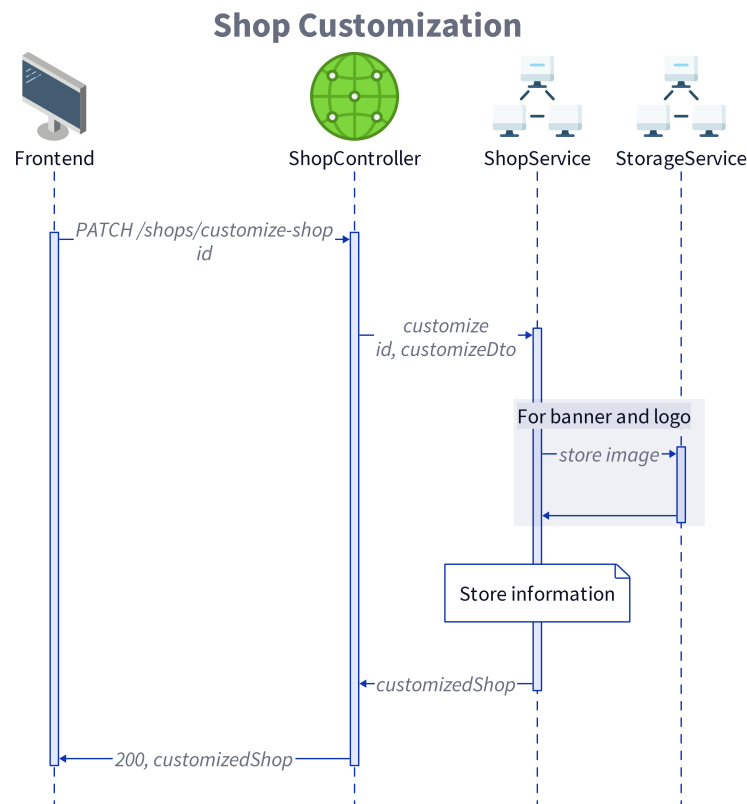


Figura 4.25: Diagrama de secuencia sobre la personalización de una tienda

Se dispone de la ruta `PATCH /shop/customize` para que un vendedor autorizado personalice su tienda, el diagrama de secuencia 4.25 muestra que ocurre internamente. El cuerpo de la consulta acepta los siguientes parámetros opcionales:

- El índice de la plantilla para la tienda
- La descripción
- Logo y banner en formato base64
- El tipo de letra
- Una lista de colores

- Un booleano que puede ocultar la tienda del internet

Es importante mencionar que en esta iteración no todos los campos son integrados con los formularios del frontend, ni tienen lógica asociada en el backend.

Formularios de personalización de tienda

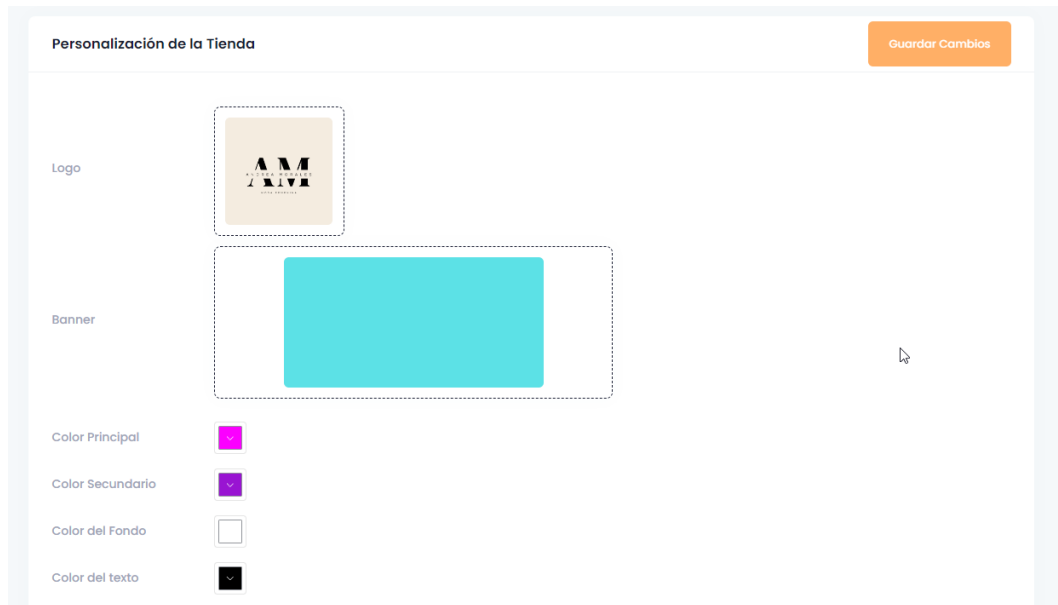
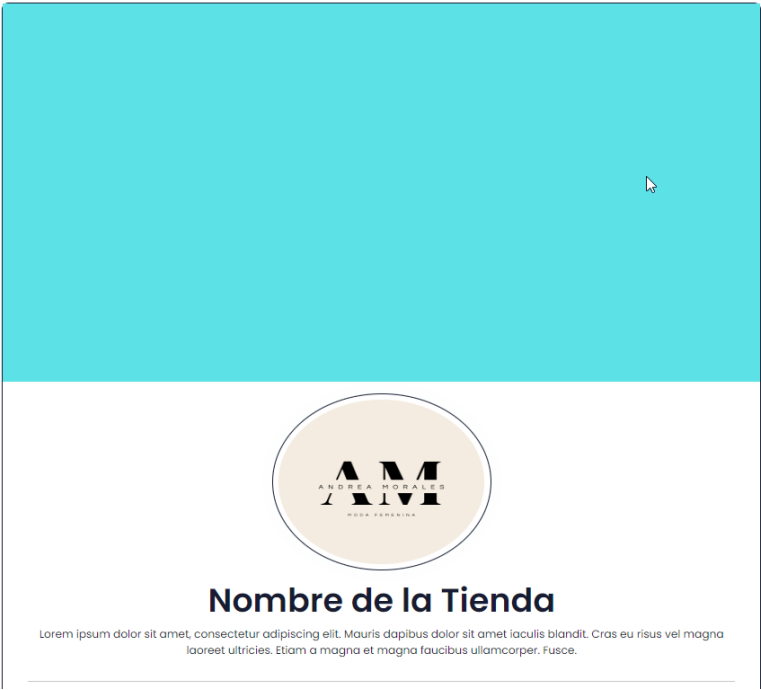


Figura 4.26: Formulario de personalización de imágenes y colores

En las figuras 4.26 y 4.27 se muestra el formulario que permite al vendedor estilizar su tienda, presentando una vista previa antes de realizar la petición. Se aprecia en 4.28 como dos tiendas despliegan estilos propios pese a ser parte de la misma plataforma web,

Formato de Tienda

Tipo 2



Nombre de la Tienda

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris dapibus dolor sit amet iaculis blandit. Cras eu risus vel magna laoreet ultricies. Etiam a magna et magna faucibus ullamcorper. Fusce.

Figura 4.27: Formulario de personalización de descripción y plantilla

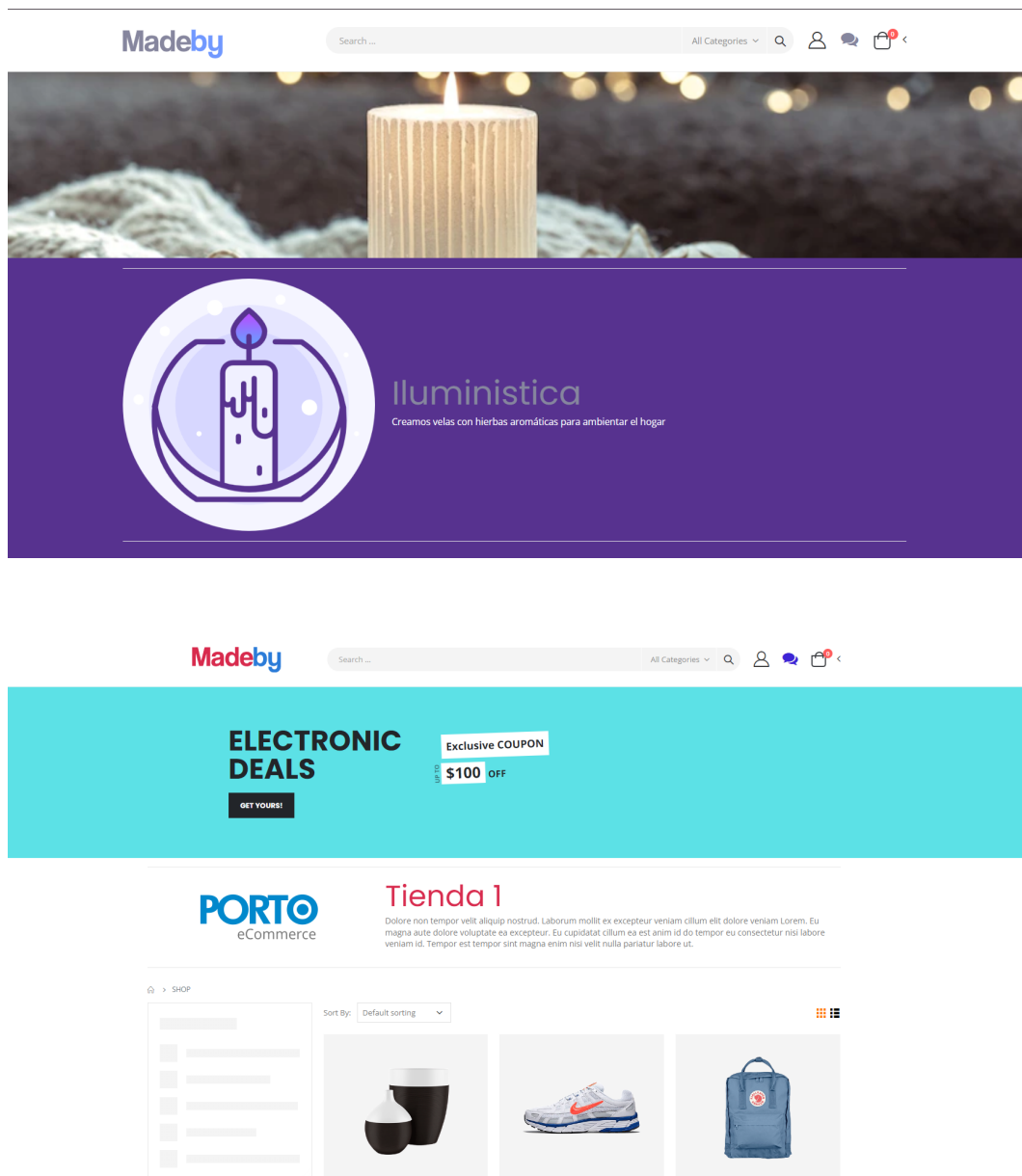


Figura 4.28: Ejemplo de tiendas personalizadas

4.2.6 Requisito Funcional 13: Carrito de compras

Un comprador puede seleccionar productos de distintas tiendas en su carrito

Diagrama de Solución

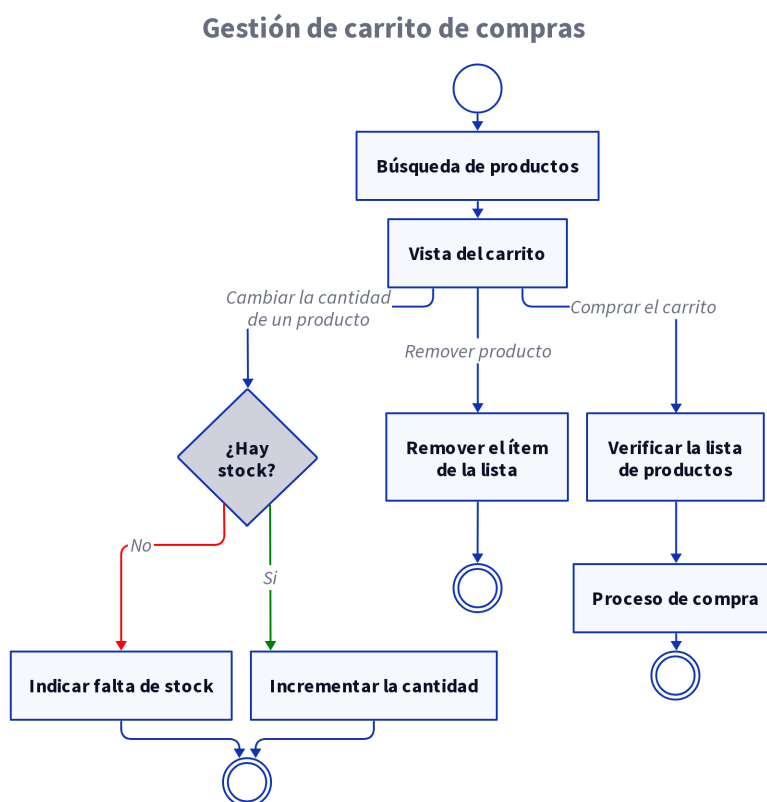


Figura 4.29: Diagrama de flujo de filtrado de productos

Este requisito se satisface con un carrito de compras que sea capaz de seguir productos de múltiples tiendas y permita modificar parámetros de compra como la cantidad de ítems considerados, remover producto del carro, personalizaciones en caso de que el producto lo permita, entre otras acciones comunes.

El diagrama 4.29 comienza con un comprador selecciona productos y los agrega al carrito. Eventualmente ingresa a la vista que dispone de operaciones sobre el carrito y los productos, solo se describen algunas acciones. Cuando el usuario esté conforme puede iniciar el proceso de compra.

Implementación

La lógica de carrito se implementa principalmente en el frontend, ocupando la información entregada por la API cuando se consulta por los productos de las tiendas. Por tanto, la mayoría de operaciones como cambiar la cantidad de productos o eliminarlos del carrito no tienen un análogo en el backend.

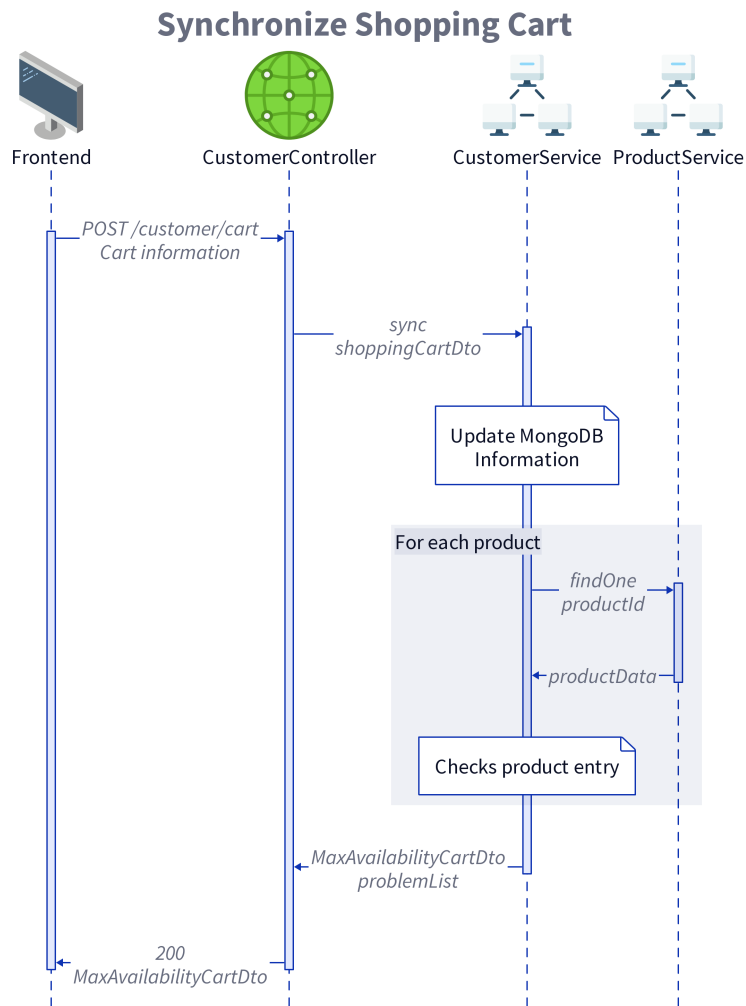


Figura 4.30: Diagrama de secuencia de sincronización y validación del carrito

Pese a lo anterior, la API internamente posee una representación del carrito que debe ser sincronizada con la del frontend y validada antes de iniciar el proceso de compra. Para esto se dispone de la ruta `POST /customer/cart` donde se envía la lista de productos con sus características. El proceso se representa en el diagrama 4.30 y termina retornando la lista de productos del carro con la cantidad máxima de stock, un código de error acorde al resultado e información extra en caso de haber algún problema como falta de existencias.

La imagen 4.31 muestra la vista resumida de la lista de productos del carrito e información de la compra cuando no hay problemas al sincronizar el carrito con el backend.

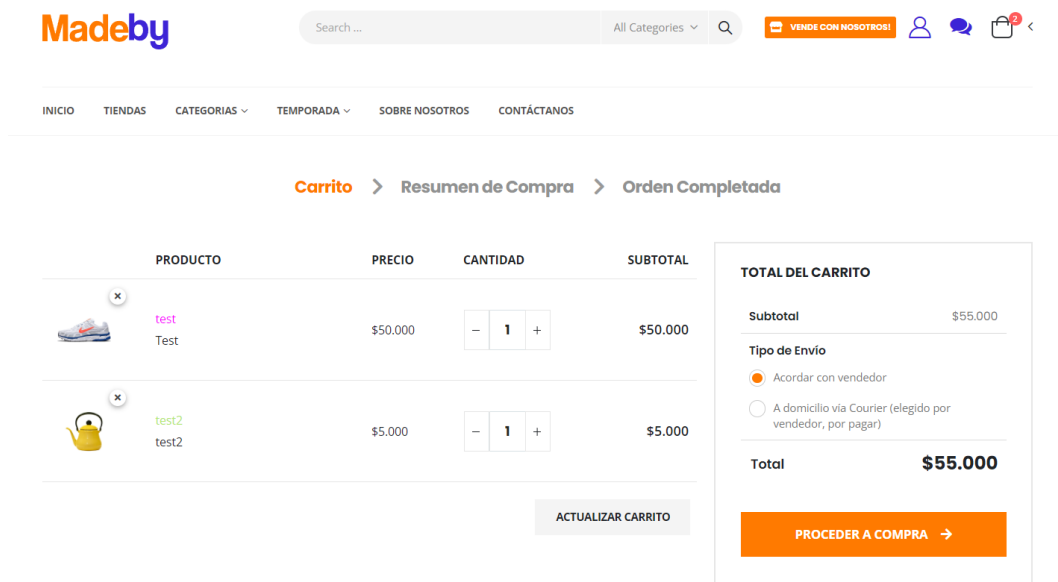


Figura 4.31: Carrito de compras

4.2.7 Requisito Funcional 14: Filtro de productos

Un comprador puede filtrar los productos según el nombre, categorías, precios y otros criterios de interés

Diagrama de Solución

Buscar, filtrar y comparar productos

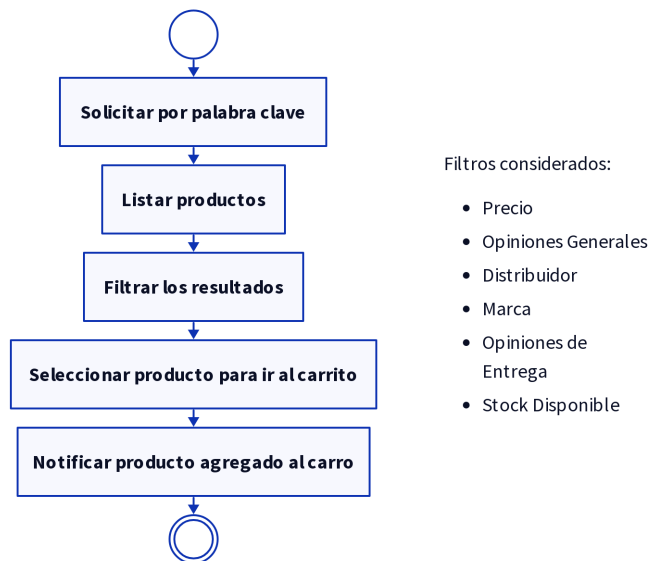


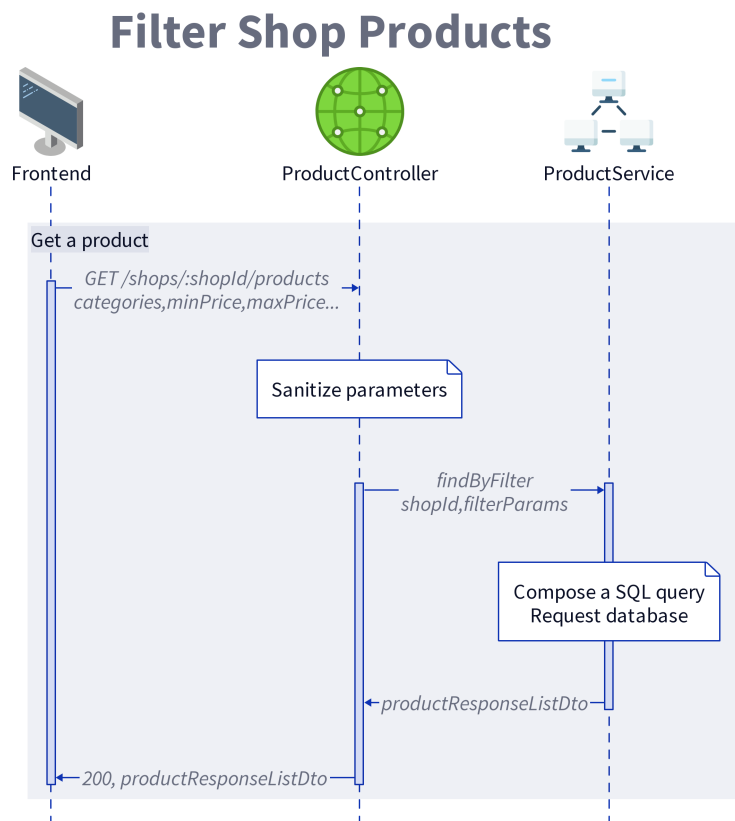
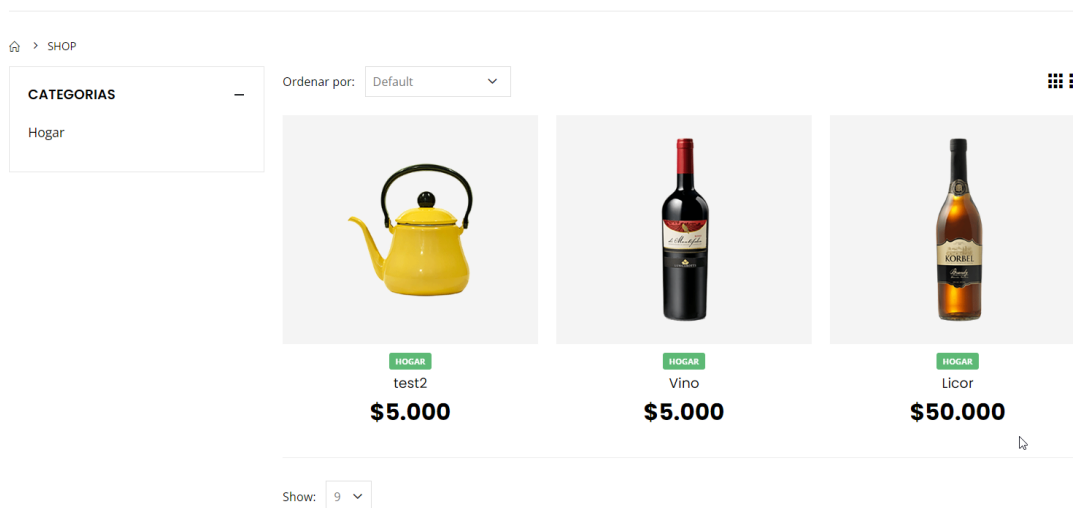
Figura 4.32: Diagrama de flujo de filtrado de productos

El diagrama 4.32 ilustra los procesos que un comprador realiza al momento de buscar un producto y comprarlo. Los filtros que se indican son comunes en los E-Commerce existentes, lo que simplifica la experiencia del usuario.

Implementación

El filtrado se implementa dentro de la ruta `GET /shops/:shopId/products` por medio de parámetros de consulta, y el procedimiento se explica en el diagrama 4.33. Se utiliza middleware para verificar el contenido de los parámetros de la consulta antes de la ejecución del controlador.

La figura 4.34 muestra en el panel izquierdo los criterios de filtrado y una lista de productos que se adecuan a esos filtros.

**Figura 4.33:** Diagrama de secuencia de consulta sobre productos filtrados**Figura 4.34:** Filtrado en tienda por categoría de producto

4.2.8 Requisito Funcional 15: Visualización de productos por usuario no autenticado

Un usuario no autenticado puede ver los productos del marketplace y agregarlos a su carrito

Implementación

La ruta para obtener la lista de productos de una tienda no utiliza autenticación JWT, sino que ocupa el decorador `@Public()`. Este decorador únicamente agrega el metadato `IS_PUBLIC_KEY = true` a la ruta `GET /shops/:shopId/products` para deshabilitar la guarda de autenticación JWT.

La lógica de gestión de productos del carrito se implementa en el frontend.

4.2.9 Requisito Funcional 16: Compra de carrito por usuario autenticado

Un usuario no autenticado debe registrarse para acceder a la compra de productos

Implementación

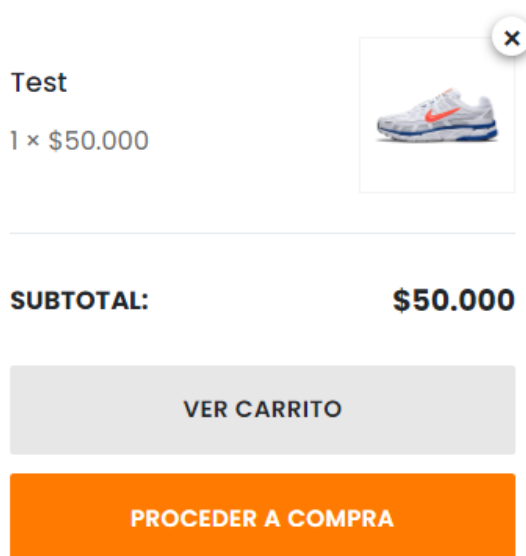
La verificación de autenticidad del comprador antes de iniciar la compra se realiza tanto en el frontend como en el backend. Por parte del frontend, la vista detallada del carrito se bloquea redirigiendo a la ruta de autenticación del servidor si no se cuenta con el token de acceso, como se muestra en la captura 4.35.

En la API, la ruta de sincronización y verificación del carrito `POST /customer/cart` se encuentra asegurada por medio de AuthGuard `jwt`. Sin este paso no es posible registrar un carrito ni iniciar el proceso de compra.

(a) Carrito de compras cuando la sesión no está iniciada

(b) Carrito de compras con sesión iniciada

Carrito de Compras



Carrito de Compras

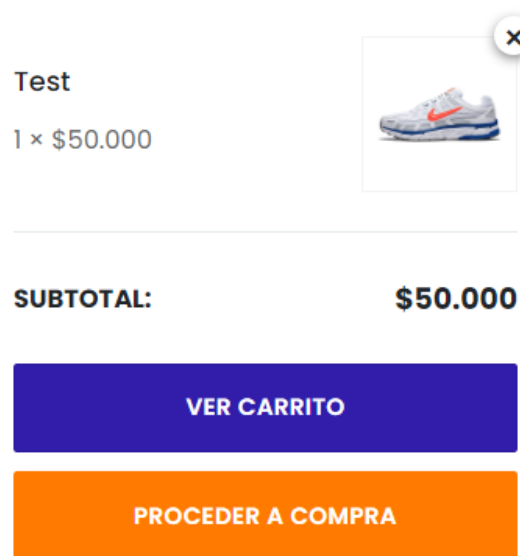


Figura 4.35: Carrito de compras cuando la sesión no está iniciada

4.2.10 Requisito Funcional 23: Gestión de direcciones de comprador

El comprador puede establecer múltiples direcciones de entrega preferidas

Diagrama de Solución

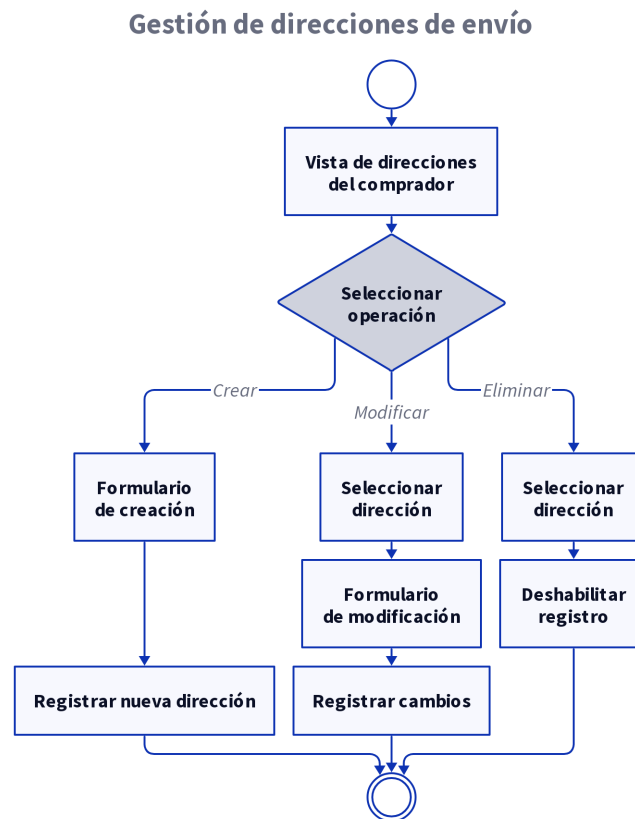


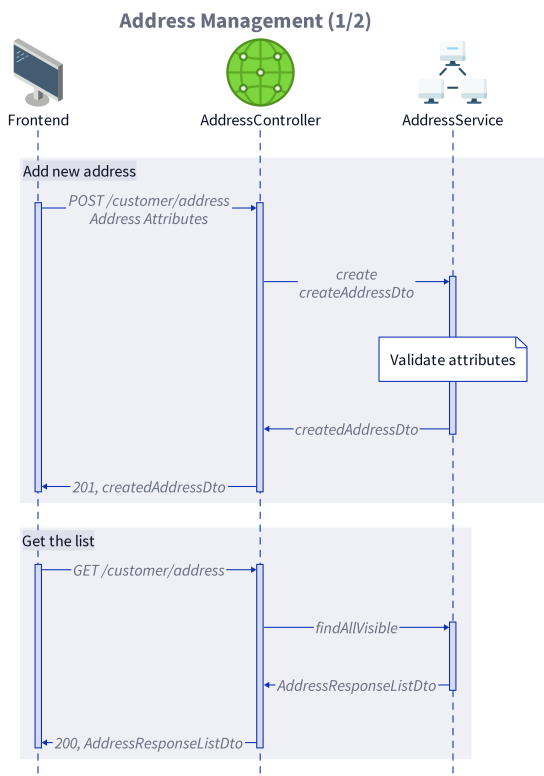
Figura 4.36: Diagrama de flujo de gestión de direcciones de comprador

El diagrama de flujo 4.36 muestra las operaciones que un comprador puede realizar para gestionar sus direcciones. Cuando agrega una nueva dirección de envío, se pide llenar un formulario de registro. Al modificar una dirección existente se solicita modificar otro formulario que contiene los valores de la dirección original. Finalmente, el comprador puede eliminar una dirección de la lista.

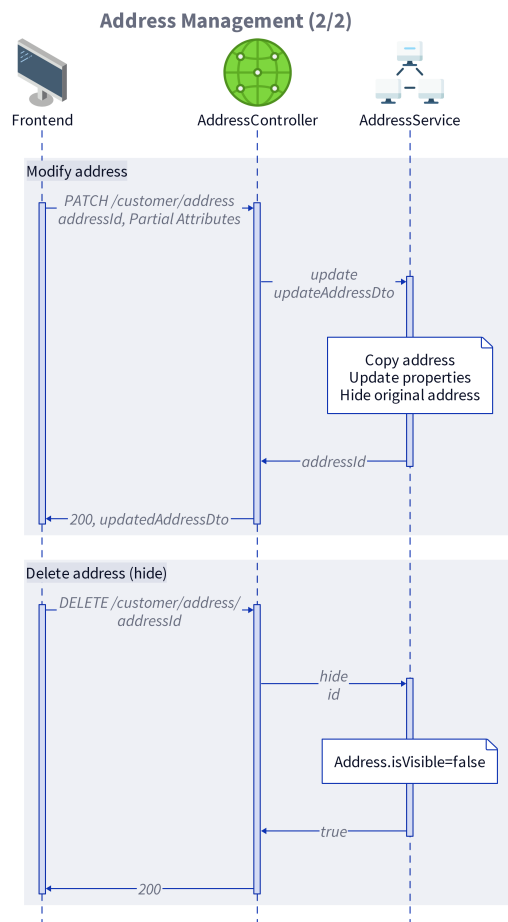
Implementación

Se implementa una interfaz CRUD en la ruta `/customer/address` como se muestra en las figuras 4.37a y 4.37b. Una observación importante es que para proteger la consistencia de los datos, modificar una dirección crea un nuevo registro en la base de datos con los datos entregados mientras que eliminar una solamente la oculta de la interfaz del comprador.

(a) Diagrama de secuencia de métodos de creación y consulta de direcciones de comprador



(b) Diagrama de secuencia de métodos de modificación y eliminación direcciones de comprador



El frontend accede a la gestión de direcciones en la vista detallada del carrito, como paso previo a la compra. Se da a escoger al usuario entre las entradas registradas o una nueva, como se muestra en la captura 4.38. Posterior a eso, el formulario de resumen de la compra mostrado en 4.39 informa al usuario sobre los productos del carrito, el total a pagar y la información necesaria para la facturación.

The screenshot shows the 'Resumen de Compra' (Purchase Summary) page. At the top, there's a navigation bar with the Madeby logo, a search bar, and links to 'VENDER CON NOSOTROS', a user profile, a chat bubble, and a shopping cart icon. Below this is a menu with 'INICIO', 'TIENDAS', 'CATEGORIAS', 'TEMPORADA', 'SOBRE NOSOTROS', and 'CONTÁCTANOS'. The main content area has a breadcrumb trail: 'Carrito > Resumen de Compra > Orden Completada'. The central table lists items in the cart:

PRODUCTO	PRECIO	CANTIDAD	SUBTOTAL
test Test	\$50.000	- 1 +	\$50.000
test Producto Test 3	\$10.000	- 1 +	\$10.000

Below the table is an 'ACTUALIZAR CARRITO' button. To the right, the 'TOTAL DEL CARRITO' section shows a 'Subtotal' of \$60.000. Under 'Tipo de Envío', 'A domicilio vía Courier (elegido por vendedor, por pagar)' is selected. The 'Datos de Envío' section includes fields for 'Región de Los Lagos', 'Puerto Montt', 'Pasaje 1, 123', and '5480000'. A 'Total' of \$60.000 is displayed, followed by a 'PROCEDER A COMPRA' button.

Figura 4.38: Panel de datos de envío antes de proceder a la compra

The screenshot shows the 'Resumen de Compra' page with the 'Orden Completada' step. The 'Datos de Facturación' section contains form fields for 'Nombre' (Cliente), 'Apellido' (Test), 'Región' (Región de Los Lagos), 'Dirección' (Pasaje 1, 123), 'Departamento, Suite, etc.', 'Ciudad' (Puerto Montt), 'Código Postal' (5480000), 'Teléfono' (12345612), 'Correo Electrónico' (cliente@gmail.com), and 'Notas Adicionales (Opcional)'. The 'Tu Pedido' section shows a list of products: 'test Test x 1' for \$50.000 and 'test Producto Test 3 x 1' for \$10.000, with a 'Subtotal' of \$60.000 and a 'Total' of \$60.000. The 'Método de Pago' section shows 'Pasarela de Pago' selected. A 'PROCEDER A PAGO' button is at the bottom.

Figura 4.39: Resumen de la compra y datos de facturación

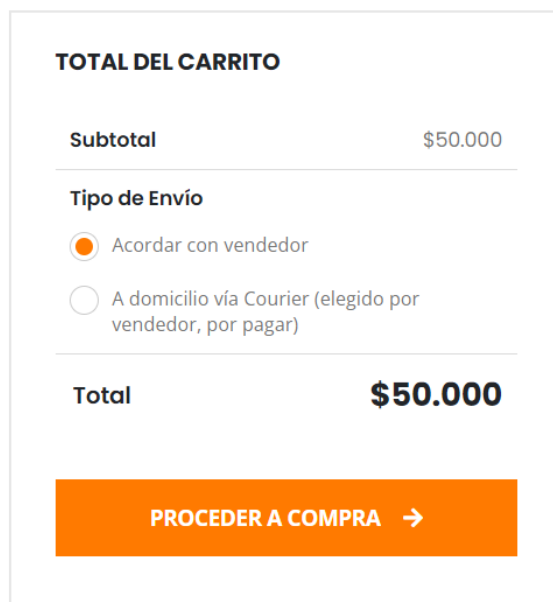
4.2.11 Requisito Funcional 24: Registro de envío de producto

El vendedor debe registrar los datos de envío necesarios después de acordarlo con el cliente

Implementación

En el frontend se agrega un panel que consulta el tipo de envío al comprador, de esta manera la compra indica si la entrega se realiza acordándolo con el vendedor o vía courier. Esto se muestra en la imagen 4.40.

La lógica de negocio requiere de integración con servicios de distribución y de pasarela de pagos. Estos ítems no fueron cubiertos durante esta iteración y en consecuencia este requisito no cuenta con una implementación en el backend.



The image shows a checkout summary box with the title "TOTAL DEL CARRITO". It contains a "Subtotal" of "\$50.000". Below this, under the heading "Tipo de Envío", there are two radio button options: "Acordar con vendedor" (which is selected) and "A domicilio vía Courier (elegido por vendedor, por pagar)". At the bottom of the box, the "Total" is displayed as "\$50.000". Below the total, there is a large orange button with the text "PROCEDER A COMPRA" and a right-pointing arrow.

Figura 4.40: Opciones de envío

4.2.12 Requisito Funcional 26: Mensajería posventa

Luego de concretar una compra, un comprador puede acceder a un canal de mensajería instantánea con el vendedor para coordinar la entrega del producto

Implementación

En el backend se agregan las rutas `/customer/chat` y `/shops/:shopId/chat` que el frontend puede establecer un canal WebSockets y transmitir datos en tiempo real. Estas rutas también son autenticadas y requieren la información sobre el canal específico con el que se interactúa. Cada uno se vincula a una orden de compra y un emprendimiento para que la vista muestre información específica del estado de la compra y el envío.

La figura 4.41 muestra el flujo de mensajería del comprador, que primero tiene que seleccionar una empresa con la que tenga alguna orden de compra vigente para luego mostrar la conversación.

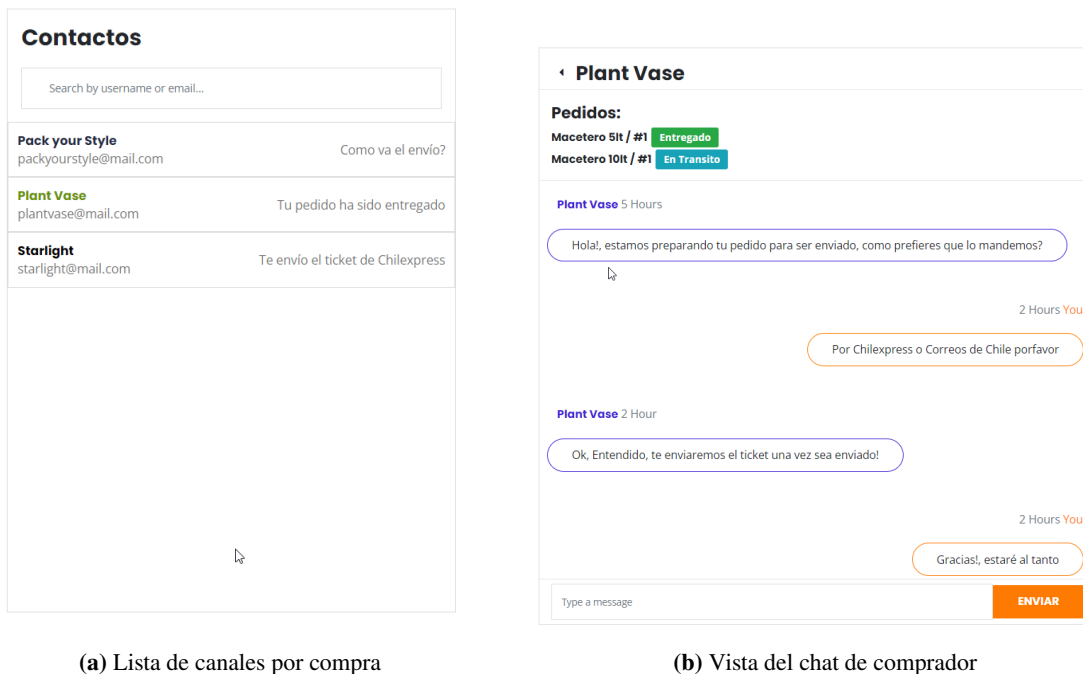


Figura 4.41: Vista del chat de comprador

La vista 4.42 muestra el chat del vendedor, que además de los mensajes enviados y recibidos muestra información sobre las preguntas y el estado de cada pedido.

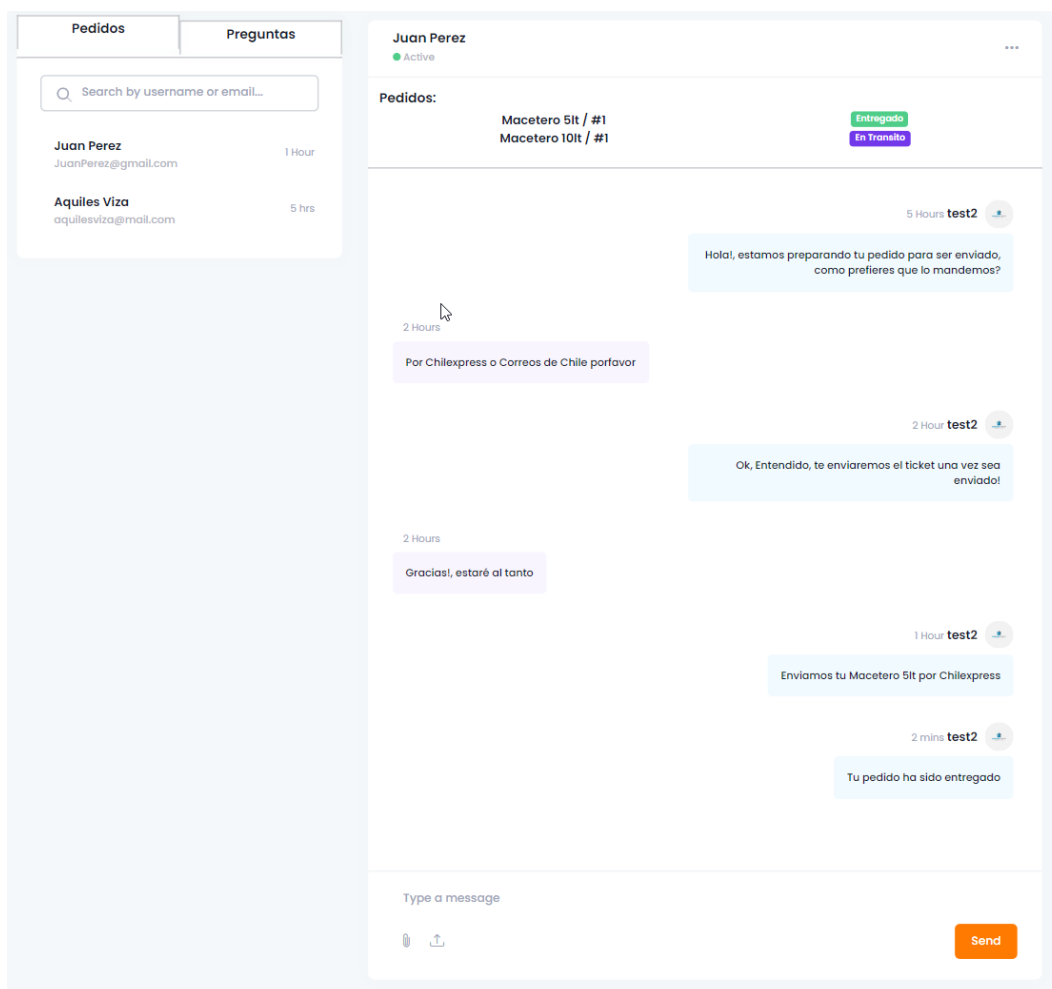


Figura 4.42: Vista del chat de vendedor

4.3 Requisitos funcionales no implementados

Debido al limitado tiempo de desarrollo y contratiempos ocurridos durante el proceso de diseño e implementación, no se consigue implementar la totalidad de requisitos funcionales descritos en la sección 3.2.1. En esta sección se enumeran los requisitos funcionales que no fueron implementados.

4.3.1 Requisitos de la gestión de usuarios

RF 4 La autenticación con doble factor o 2FA está disponible como estrategia de Passport. Sin embargo, el cómo se integre depende de la decisión sobre el uso de plataformas de autenticación de terceros, delegada para después del MPV

RF 5 La eliminación de una cuenta requiere consideraciones sobre la consistencia e integridad de los datos almacenados, además de que puede requerir criterios legales y no agrega un valor directo a los usuarios. Se deja fuera del MPV

RF 6 El módulo de analíticas y recomendaciones se deja fuera del MPV

RF 7 La integración con una pasarela de pagos se deja fuera del MPV

4.3.2 Requisitos de gestión de tienda

RF 8 Se unifica el concepto de vendedor con el de la tienda, con tal de agilizar el proceso de registro. Los detalles se encuentran en 4.2.1

RF 11 Su implementación se integra al RF 8, véase 4.2.5.

RF 12 Unificar la implementación del vendedor con la tienda dificulta la lógica de mantener múltiples usuarios gestionando una misma tienda, además de requerir un esquema de autorización más elaborado.

4.3.3 Requisitos del marketplace

RF 17 La sección de preguntas y respuestas sobre productos se deja fuera del MPV

4.3.4 Requisitos de ventas

RF 18 Si bien se integra validación de formularios en el lado del frontend, esta funcionalidad es parte del módulo de analíticas y recomendaciones que se deja fuera del MPV

RF 19 La integración con una pasarela de pagos se deja fuera del MPV

RF 20 La publicación de ofertas se deja fuera del MPV

RF 21 La lógica del estado de la compra se deja fuera del MPV

RF 22 La sección de reseñas de un producto se deja fuera del MPV

4.3.5 Requisitos de envío

RF 24 El módulo de integración con servicios de courier se deja fuera del MPV

RF 25 El módulo de analíticas y recomendaciones se deja fuera del MPV

4.3.6 Requisitos de analíticas

RF 26 El módulo de analíticas y recomendaciones se deja fuera del MPV

4.3.7 Requisitos de soporte

RF 28 a 30 La implementación de una API de soporte deja fuera del MPV

4.3.8 Requisitos de festividades y campañas

RF 31 y 32 La gestión de campañas o eventos se deja fuera del MPV

4.4 Demostración del sistema

En esta sección se describe por medio de imágenes el proceso de uso del sistema completo siguiendo el “Happy Path”, o flujo sin errores. La demostración se divide en 3 etapas, representadas en el diagrama 4.43, y cada una enumera hitos importantes durante su ejecución. Los procesos son los siguientes:

- El registro de un vendedor y su tienda e ingreso de productos
- Un comprador que visita la plataforma, busca productos y los agrega al carrito, crea su cuenta y finaliza su compra
- Interacción entre comprador y vendedor para concretar venta y entrega del producto

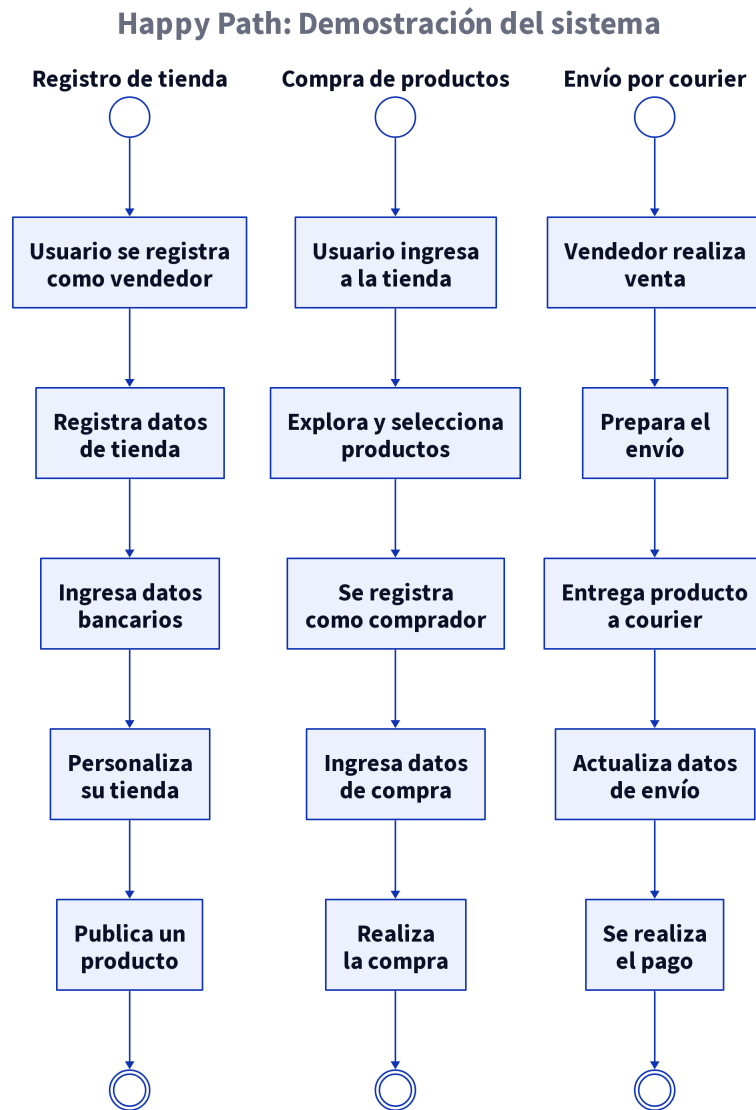


Figura 4.43: Diagramas de flujo que demuestran la operación del sistema *Madeby*

4.4.1 Happy Path 1: Registro de vendedor y gestión de tienda

1. Un usuario puede registrarse como vendedor por medio del “landing page” de gestión de tiendas, y en caso de no tener una cuenta puede registrarse con sus datos. El proceso se muestra en la figura 4.44

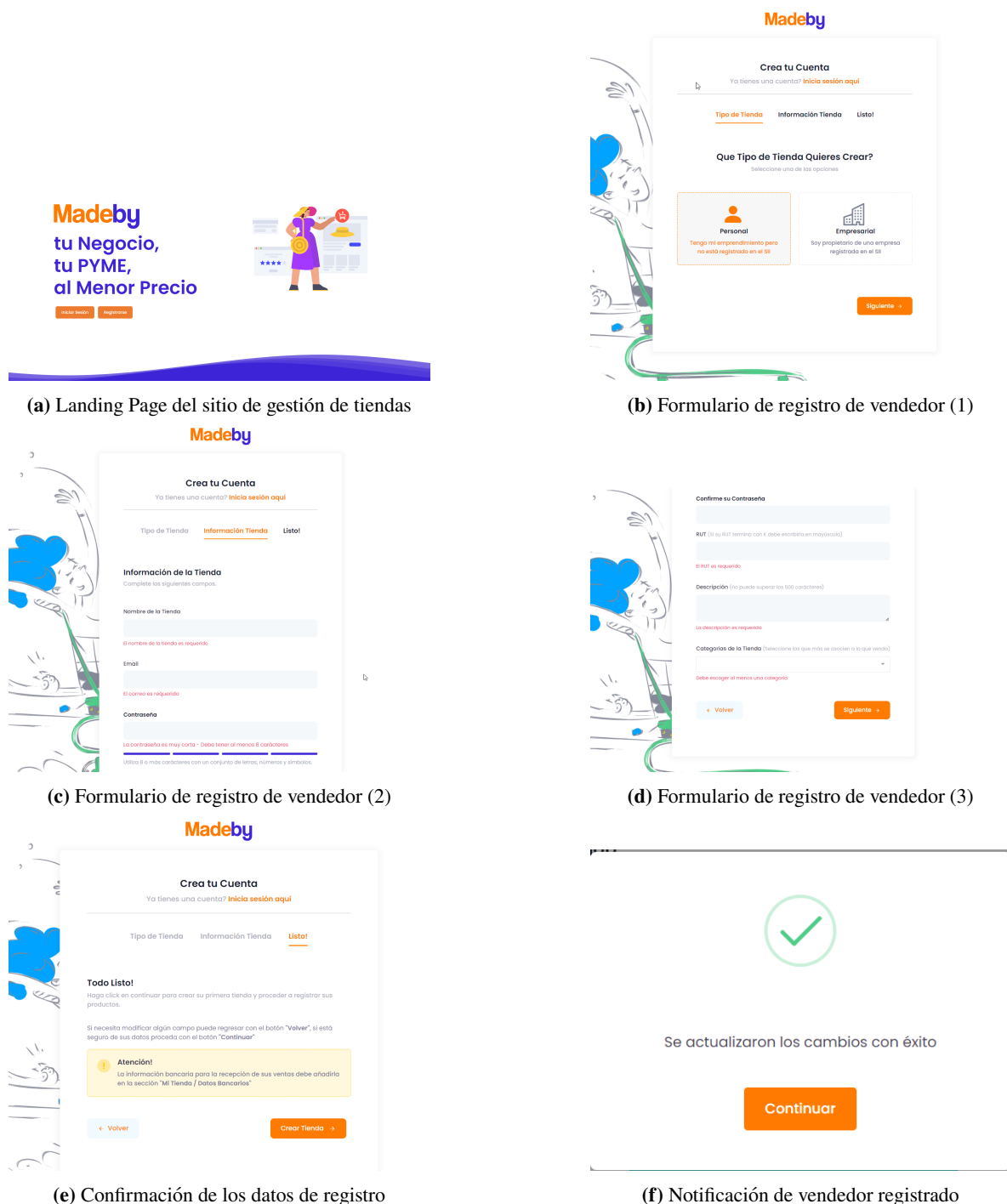
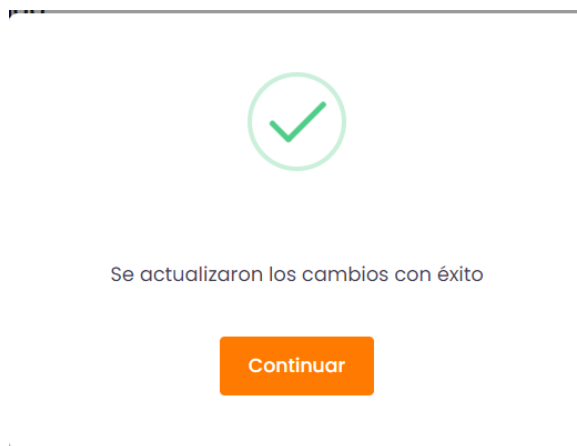
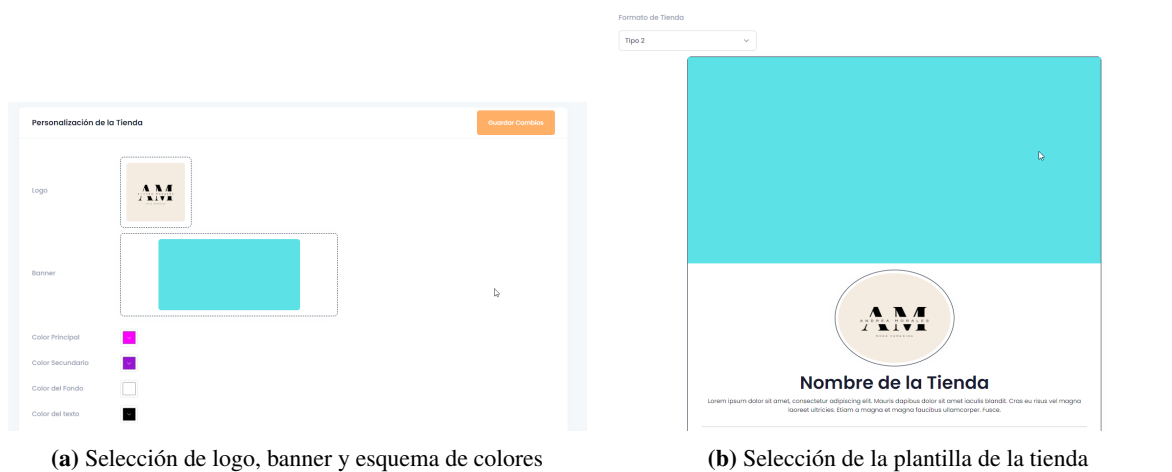


Figura 4.44: Registro de vendedor

2. El vendedor personaliza su tienda definiendo los colores y las figuras a ocupar, lo que se observa en las imágenes de la figura 4.45



(c) Notificación de cambios a la tienda

Figura 4.45: Personalización de tienda

3. La figura 4.46 muestra como se registra un producto en el sistema por medio del formulario de registro de producto, y como se muestran en la vista de la tienda del vendedor

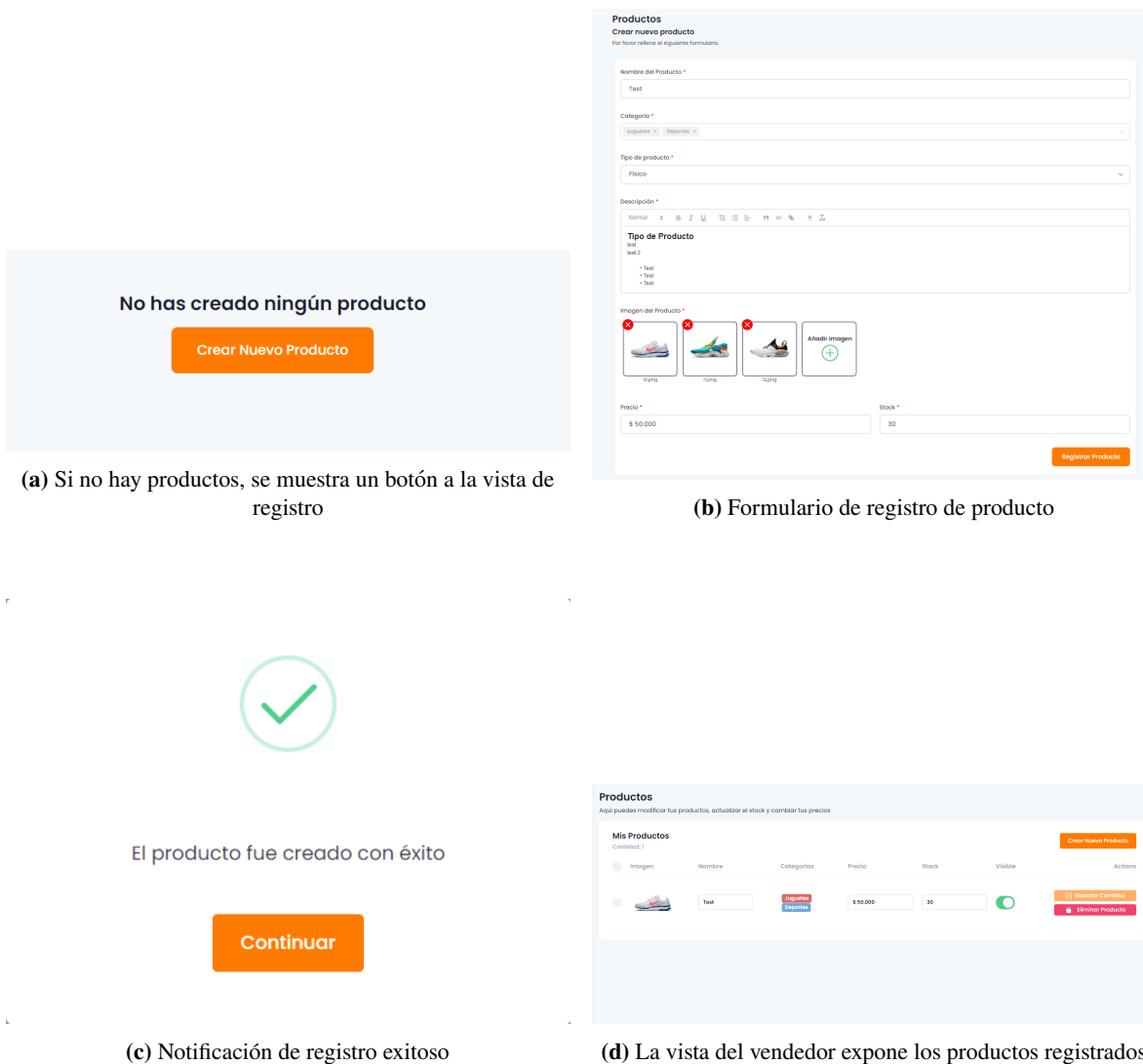
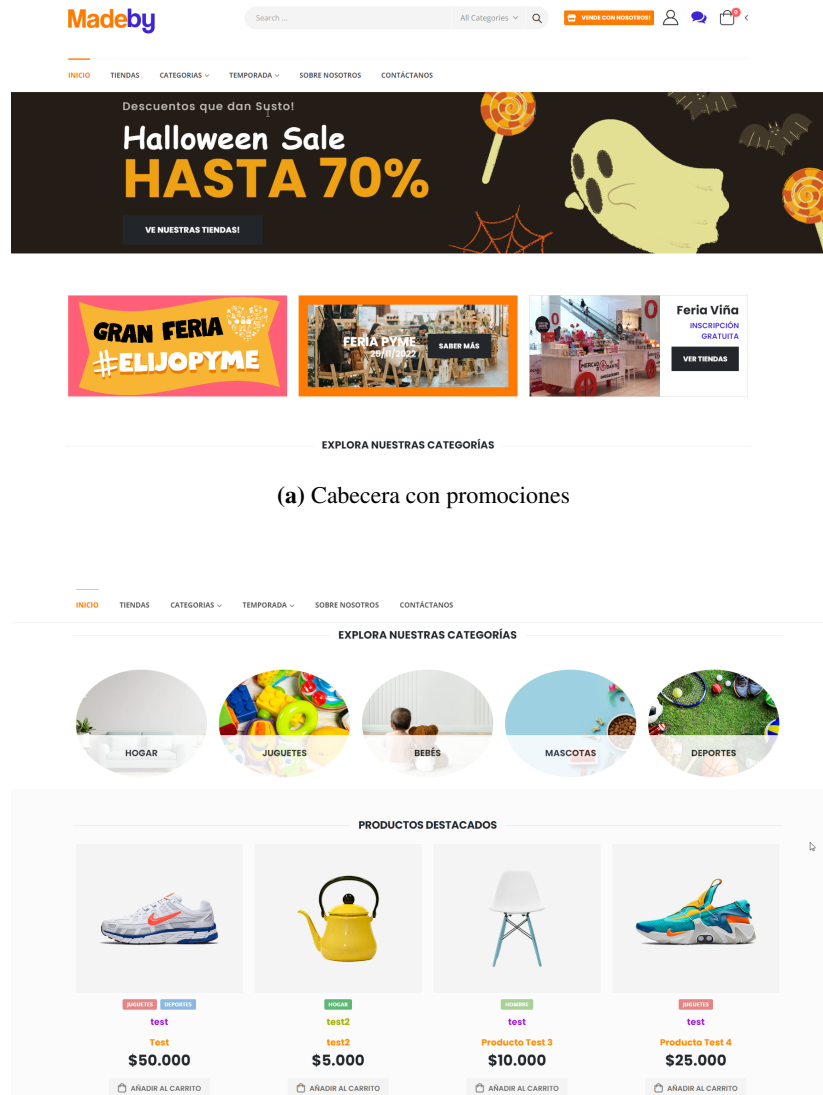


Figura 4.46: Publicación de un producto

4.4.2 Happy Path 2: Registro de vendedor y su tienda

1. Al ingresar al marketplace, un usuario accede al “landing page” mostrado en la captura 4.47 y recibe información sobre promociones, productos destacados y categorías, entre otros.



(a) Cabecera con promociones

(b) Categorías y productos

Figura 4.47: Landing page de Madeby

2. El usuario accede a una tienda y puede seleccionar productos como se aprecia en la figura 4.48. No requiere de credenciales de comprador aún.

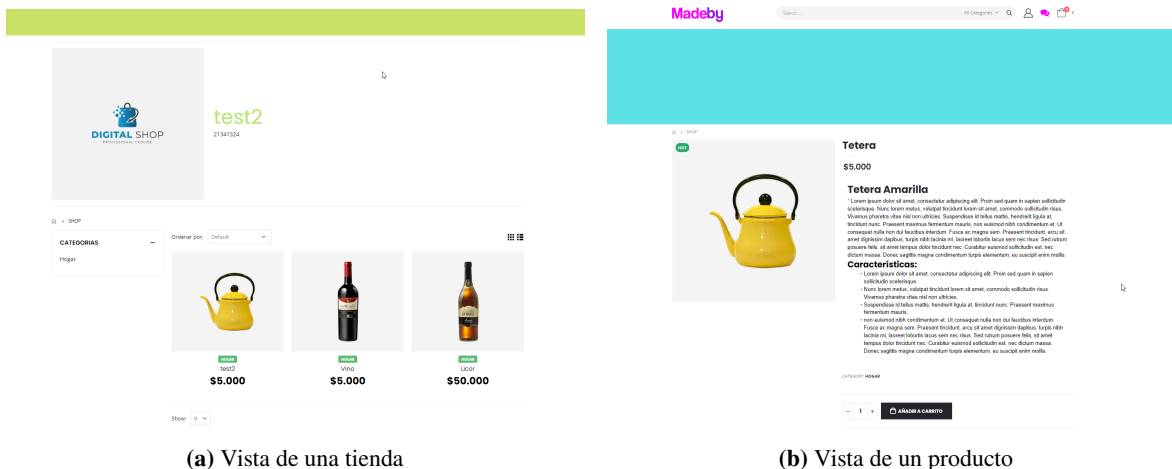


Figura 4.48: Vista de tienda y de producto

3. Cuando un usuario agrega un producto a su carrito, se muestra la notificación de la figura 4.49

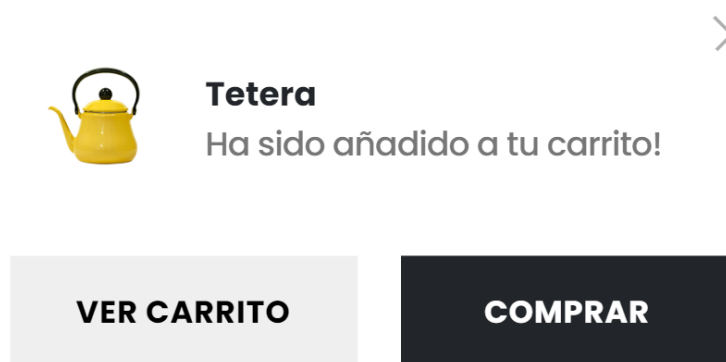


Figura 4.49: Notificación de producto agregado al carrito

4. Cuando el usuario no se encuentra autenticado como comprador, se redirige a la vista del formulario de registro donde puede acceder con una cuenta existente o crear una nueva, como se muestra en 4.50.

The screenshot shows the 'Mi Cuenta' (My Account) page of the Madeby website. The page layout includes a header with the Madeby logo, a search bar, and navigation links. Below the header is a navigation bar with links: INICIO, TIENDAS, CATEGORÍAS, TEMPORADA, SOBRE NOSOTROS, and CONTÁCTANOS. The main content area is titled 'Mi Cuenta' and contains two forms: 'Iniciar Sesión' (Login) and 'Registrarse' (Register). The 'Iniciar Sesión' form has fields for 'Correo Electrónico' and 'Contraseña', a 'Recordarme' checkbox, and a 'Recuperar contraseña' link. The 'Registrarse' form has fields for 'Correo Electrónico', 'Contraseña', 'RUT', 'Nombre', and 'Apellido', and a 'REGISTRARSE' button.

Figura 4.50: Formulario de acceso o registro de comprador

5. Luego de obtener las credenciales, la plataforma verifica que el carrito sea válido y redirige al comprador a la vista detallada del carrito de compras. Aquí se especifica la información del envío y se va al formulario de facturación que contiene los datos de la compra. Finalmente se confirma la compra y se indica con una notificación al usuario. El proceso se muestra en la secuencia de imágenes de la figura 4.51

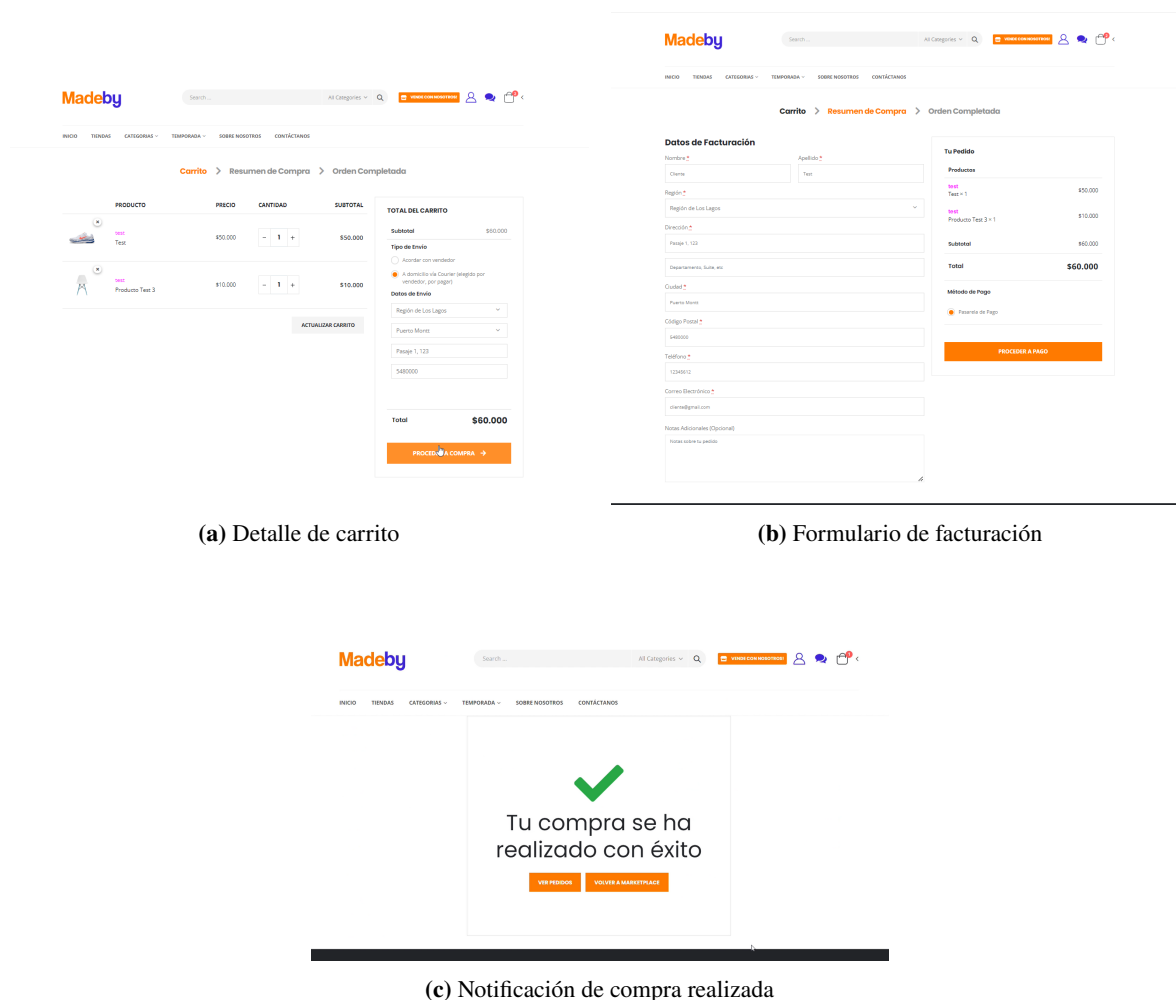


Figura 4.51: Proceso de compra de carrito

4.4.3 Happy Path 3: Registro de vendedor y su tienda

1. El vendedor revisa la sección de órdenes de compra en su perfil de usuario. La vista 4.52 da información como la cantidad de órdenes pendientes y permite ir al chat.

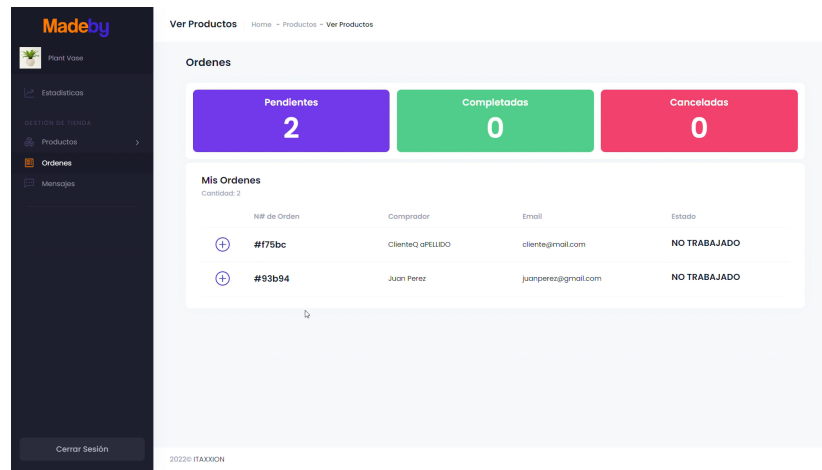


Figura 4.52: Órdenes de compra del vendedor

2. Por medio del chat, un vendedor se puede comunicar con sus compradores para coordinar la entrega del producto, como se ve en la figura 4.53

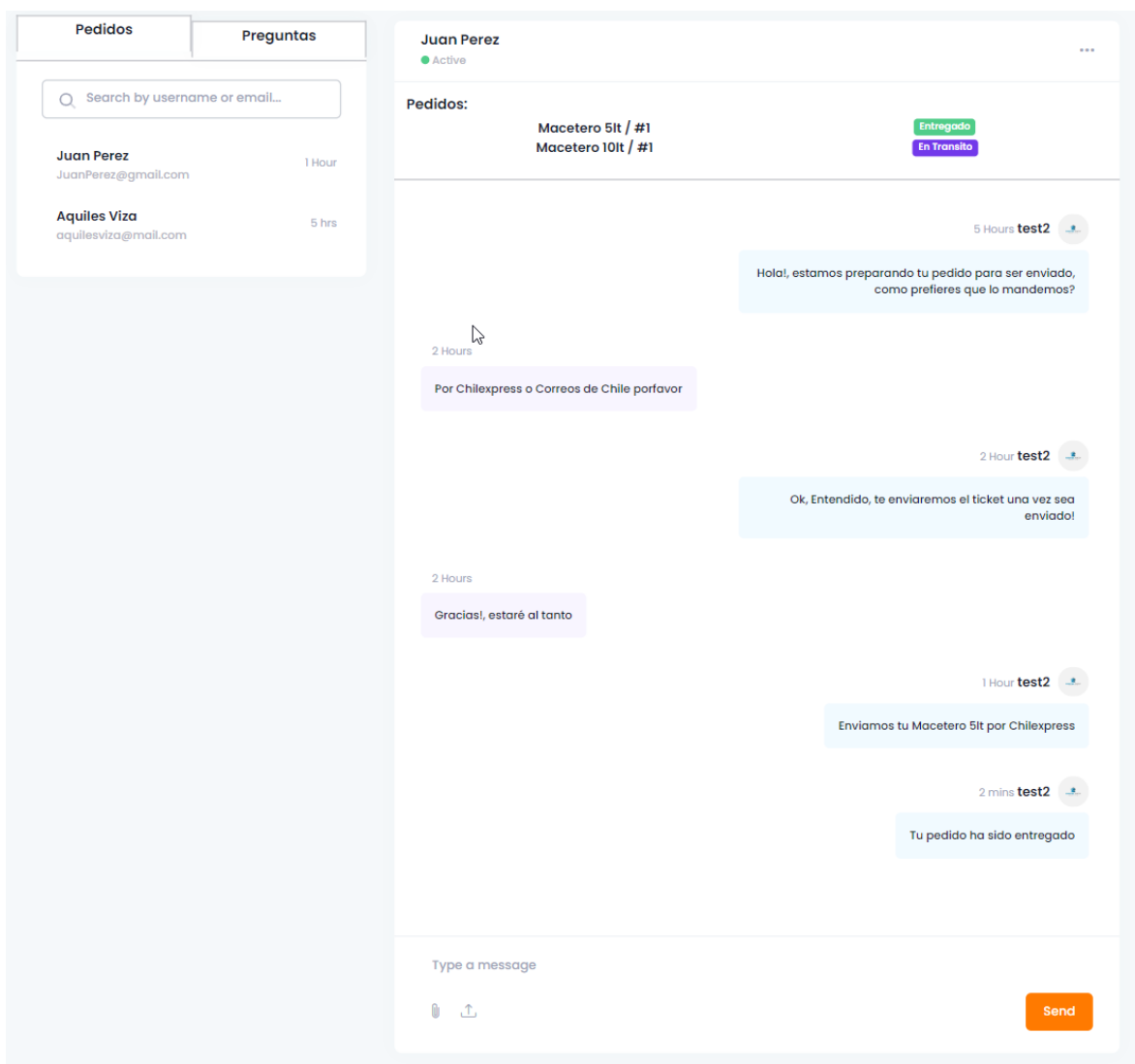


Figura 4.53: Chat del Vendedor

3. Por medio del chat, el comprador se coordina con el vendedor. La imagen 4.54 muestra que la vista entrega información sobre el estado del pedido

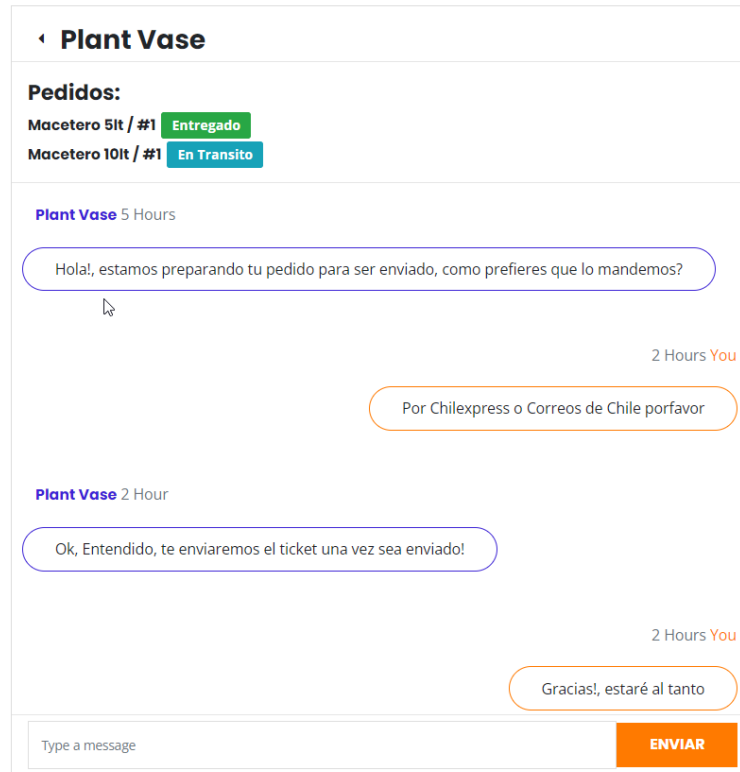


Figura 4.54: Chat del comprador

Capítulo 5

Validación

5.1 Validación de usabilidad

Para validar la usabilidad del sitio, se ha realizado una entrevista tanto a compradores como vendedores. En cada una se les solicita realizar tareas dentro del sitio y se consulta por su opinión sobre la plataforma, con el objetivo de obtener retroalimentación que oriente el desarrollo.

5.1.1 Vendedores

La entrevista realizada a los vendedores consiste en realizar las siguientes tareas:

- Registrar una tienda
- Registrar un producto
- Personalizar la tienda

Entrevistados

Se entrevista a 5 emprendedores dedicados a la producción de bienes en distintos rubros. Los rangos etarios se anotan en la tabla 5.1 y entre estos dos llevan menos de un año emprendiendo y solo uno más de 3 años. Solamente uno de los entrevistados no ha ocupado nunca una plataforma marketplace.

Resultados

No hubo problemas de uso al momento de realizar las tareas solicitadas. El registro de tienda fue descrito como sencillo por los usuarios, y se comparó al proceso de otros sitios similares, mientras que el registro de productos se consideró como intuitivo y simple. Se menciona que el proceso de personalización de tienda es

Edad	Cantidad
Entre 18 y 29 años	1
Entre 30 y 39 años	2
Entre 40 y 49 años	1
De 50 años o más	1

Tabla 5.1: Rango etario de emprendedores entrevistados

rápido y fácil. Los emprendedores que no ocupaban aplicaciones E-Commerce afirmaron que la plataforma les permitiría crear un sitio de ventas de forma rápida.

Al comentar que los productos se publicarían en un marketplace, los usuarios indicaron que el proyecto puede ayudarlos a alcanzar nuevas personas y es una buena idea juntar a los emprendedores en un lugar donde los chilenos puedan comprar toda clase de productos.

5.1.2 Compradores

Procedimiento

Se solicita relatar el proceso mental y opiniones al realizar las siguientes tareas:

- Buscar un producto
- Ver los detalles del producto
- Comprar el producto

Entrevistados

Se entrevistó a 10 personas, tomando en consideración los siguientes criterios:

- Al menos una que nunca haya realizado compras por internet
- Al menos tres que compren habitualmente por internet
- Al menos una mayor a 45 años

Resultados

No se reportaron problemas de uso y se comenta que el sitio funciona de la manera esperada, similar al proceso de compra de otras aplicaciones.

Se detecta que el prototipo no resuelve la selección del método de entrega. Los usuarios reafirman su preferencia por el despacho domicilio, pero en general las compras a emprendedores se entregan en puntos de la

ciudad según la conveniencia de ambas partes. En consecuencia, se opta por el uso del acuerdo con el vendedor como método de entrega en el MPV, en donde el comprador decide si recibir el producto en un lugar acordado con el emprendedor, o si desea ocupar una empresa de transporte.

Otra observación relevante es que además del uso del buscador, los usuarios ocupan directamente las categorías para filtrar y encontrar productos. Un elemento que hay que integrar en el trabajo futuro.

5.2 Verificación funcional del backend

El framework **NestJS** contiene utilidades de verificación de código por medio del módulo `@nestjs/testing`. Con esta biblioteca se realiza una prueba de extremo a extremo que requiere levantar toda la infraestructura de forma local antes de ejecutar el servidor y efectuar las consultas.

La prueba consiste en realizar el “Happy Path” descrito a continuación por medio del uso de peticiones HTTP a una instancia temporal del servidor.

1. Registrar una categoría de productos
2. Registrar una tienda y verificar que no tenga productos
3. Registrar un producto y verificar que la tienda lo muestre y tenga las propiedades esperadas, como sus categorías e información de la tienda

```
PASS test/app.e2e-spec.ts
App e2e
  Products
    Create and Delete Products
      ✓ should register a category (66 ms)
      ✓ should get the created category (12 ms)
      ✓ should register a shop (45 ms)
      ✓ should get the created shop with no products (26 ms)
      ✓ should register a product (35 ms)
      ✓ should get the shop with the created product (22 ms)
      ✓ should get a list of products w/ category and shop info (20 ms)
      ✓ should get a product w/ category and shop info (15 ms)
      ✓ should get a list of products related to a shop (19 ms)
      ✓ should be able to customize the shop (20 ms)

Test Suites: 1 passed, 1 total
Tests:       10 passed, 10 total
Snapshots:  0 total
Time:        3.884 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more._
```

Figura 5.1: Salida del test de extremo a extremo

4. Personalizar la tienda
5. Crear un comprador y obtener su información
6. Actualizar la información del comprador
7. Autenticar al comprador

Al ejecutar la prueba se generan los resultados mostrados en la figura 5.1, validando que la lógica de negocio de estas operaciones está implementada.

Capítulo 6

Conclusiones

El presente trabajo desarrolló, implementó y validó un mínimo producto viable de plataforma E-Commerce orientada a los emprendedores locales y artesanos, priorizando una fase sólida de caracterización del mercado nacional y el estado del arte y la identificación y descripción de los requisitos de usuario. Esto permitió crear una base con potencial de facilitar la entrada de micro, pequeñas y medianas empresas al comercio digital, proporcionando un espacio diferenciado que preserva su identidad de marca y evita la competencia directa con grandes empresas de retail.

La implementación demuestra que la plataforma cumple con los flujos principales de los usuarios (compradores y vendedores) mediante ítems como la personalización de tiendas y el chat posventa. Sin embargo, el potencial competitivo y valor diferencial dependen enteramente del desarrollo de funcionalidades que no han sido implementadas durante esta iteración.

Para que *Madeby* se integre al mercado como una solución viable y sostenible, se requiere dirigir el desarrollo a:

1. La conexión con redes sociales como Instagram Shops y Facebook Marketplace para ampliar el alcance
2. Integrar servicios de courier tradicionales y desarrollar un módulo de gestión de entregas utilizando servicios de MIPYMES
3. Disponer de dashboards analíticos a los vendedores, con miras al uso de Machine Learning para generar conocimiento útil para el mercado MIPYME
4. Establecer convenios con actores relevantes como empresas de microcrédito, organizaciones de MIPYMES y municipalidades con tal de captar la atención de más usuarios y participar en ferias o eventos.
5. Disponer de un módulo orientado a la formación en aspectos legales, financieros y de marketing ya sea mediante una plataforma de aprendizaje integrada o convenios de capacitación y talleres de terceros

La validación técnica realizada por medio de entrevistas a usuarios y la verificación del backend con pruebas de extremo a extremo confirman la funcionalidad y usabilidad del producto mínimo viable en ítems como la gestión de tiendas personalizables, elementos básicos del proceso de compra y la autenticación y autorización de los usuarios. No obstante, el éxito a largo plazo, la capacidad de establecerse en el mercado y la rentabilidad de la plataforma dependen exclusivamente de la ejecución e integración oportuna de las funcionalidades recién descritas. Las que en conjunto a la implementación presentada en esta memoria, constituyen una propuesta de valor innovadora.

Anexos

En este capítulo, se adjuntan los contenidos utilizados para desarrollar el proyecto, incluido un video de demostración del sistema.

1. Videos Demo del sistema
2. Diagramas, Proceso de Diseño del Sistema

Glosario

API (Application Programming Interface)

Conjunto de reglas y protocolos que establecen mecanismos para que componentes de software se comuniquen e intercambien información. En aplicaciones web representa la interfaz entre backend, frontend, apps móviles y servicios externos.

Backend

Parte de una aplicación que contiene la lógica de negocio y sus algoritmos, el manejo de las bases de datos, la integración de servicios externos y se encarga de disponer de una API para los usuarios.

BackOffice

Conjunto de herramientas y subsistemas privados utilizados por el personal administrativo de una aplicación para la gestión de aspectos internos. Suele incluir funcionalidades para la generación de informes, modificación de parámetros de aplicación y gestión de recursos como usuarios, ordenes de compra, entre otros.

Base de Datos

Sistema organizado para almacenar, gestionar y recuperar datos de manera eficiente, proporcionando una estructura lógica y un conjunto de reglas para el acceso y manipulación de la información.

Componente

Parte modular, reutilizable e independiente de una aplicación que representa servicios o módulos internos con una responsabilidad definida. Pueden ser reutilizados en diferentes contextos.

Consultoría

Servicio profesional que brinda asesoría experta en tecnología, estrategia y procesos para resolver problemas específicos o mejorar sistemas.

Courier

Empresa de transporte y envío de paquetes, documentos y mercancías que usualmente garantiza seguridad y un tiempo de entrega definido. Pueden disponer de una API pública para compartir información de seguimiento del producto, costos de envío y notificaciones de estado.

CRUD (Create, Read, Update, Delete)

Conjunto de operaciones de básicas aplicadas sobre un recurso del sistema. Suelen ocuparse sobre entidades persistentes y se utiliza comunmente en interfaces de APIs REST, bases de datos y ORMs

Digitalización

Conversión de procesos, servicios e información analógicos en formatos y flujos digitales, permitiendo su automatización y facilitando su acceso, manipulación y distribución a través de medios electrónicos.

DTO (Data Transfer Object)

Patrón de diseño que encapsula los atributos de un recurso al transferir entre distintos componentes de la aplicación. Permite mantener múltiples representaciones con distintos campos de un mismo recursos, reduciendo el acople con el modelo interno y evitando la transferencia de campos innecesarios o que pueden representar un riesgo.

E-Commerce

Modelo de negocio basado en la compra y venta de productos o servicios a través de internet.

Emprendedor

Individuo que identifica una oportunidad de negocio, y asume los riesgos de crear, desarrollar y gestionar una empresa.

Framework

Conjunto estructurado de herramientas, bibliotecas y convenciones que provee una base para desarrollar aplicaciones de manera rápida, estandarizada y mantenible.

Frontend

Parte de una aplicación con la que interactúa un usuario de manera gráfica por medio de su navegador o aplicación móvil. Su responsabilidad es establecer una estética y presentación alineados con el sistema y el usuario objetivo, implementar funcionalidades de borde como mantener el estado del cliente y, en caso de existir un backend, realizar peticiones a las rutas disponibles de la API.

Happy Path

Flujo de ejecución ideal y esperado de un caso de uso o funcionalidad, donde no ocurren errores ni condiciones excepcionales. Se utiliza como escenario principal de prueba.

HTML (Hypertext Transfer Markup Language)

Es utilizado para estructurar las páginas de un sitio web por medio de marcas que etiquetan el contenido y enlaces para conectar a otras páginas y sitios.

HTTP (Hypertext Transfer Protocol)

Define un conjunto de directivas para la transmisión de documentos entre clientes y servidores. Opera mediante un modelo de petición y respuesta “stateless”.

JWT (JSON Web Token)

Estándar abierto RFC 7519 que define un formato de información basado en un objeto JSON firmado criptográficamente. Es utilizado en autenticación y autorización en APIs web.

Marketplace

Plataforma digital que facilita transacciones entre múltiples vendedores y compradores, encargándose de tareas como el proceso de distribución, comisiones, descuentos, entre otros ítems.

MIPYME (Micro, Pequeña y Mediana Empresa)

Conjunto de empresas que agrupa a las PYME y a las microempresas.

MPV (Mínimo producto viable)

Versión de un producto que cuenta con capacidades suficientes para validar su viabilidad en el mercado. Permite evaluar hipótesis clave con usuarios reales por medio de una implementación con características esenciales, mecanismos de feedback temprano e iteraciones basadas en datos.

NoSQL (Not only SQL)

Tipo de bases de datos diseñadas para operar sobre datos no estructurados o semiestructurados. En general se caracterizan por su escalabilidad horizontal y soportar esquemas dinámicos.

ORM (Object-Relational Mapping)

Técnica de programación que genera una capa de abstracción entre un lenguaje de programación y las consultas realizadas a una base de datos. Evita mantener consultas específicas de un motor de base de datos en el código y usualmente facilita la portabilidad entre estos.

Pasarela de Pagos

Servicio externo que procesa transacciones de pago online de forma segura (Ej: Stripe, PayPal, MercadoPago).

PYME (Pequeña y Mediana Empresa)

Empresas de tamaño reducido que cumplen con ciertos criterios de clasificación según el número de empleados, el volumen de ventas o el balance general. Suelen tener una estructura organizativa más flexible y adaptativa que las grandes corporaciones y desempeñan un papel importante en las economías locales.

Red social

Plataforma online que conecta usuarios para compartir contenido, interactuar y formar comunidades por medio de interfaces interactivas.

Repositorio

Sistema de almacenamiento de artefactos como archivos y código fuente que permite su manipulación, la gestión de permisos y el historial de cambios.

Requisito Funcional

Descripción de un comportamiento concreto, observable y esperado del sistema. Es necesario para cumplir con los objetivos del negocio o las necesidades del usuario.

Requisito no Funcional

descripciones sobre cómo debe operar un sistema mediante criterios asociados a atributos de calidad o la manera en que ejecuta una acción. Si bien no establece una funcionalidad específica, se asocian a comportamientos medibles y verificables empíricamente.

REST (Representational State Transfer)

Estilo de arquitectura distribuida que establece restricciones de diseño para APIs web. Se caracteriza por establecer al recurso como unidad fundamental identificable mediante una o varias URIs, la transferencia de representaciones “stateless” mediante operaciones HTTP (GET, POST, entre otras) y el uso de hipermedia como mecanismo de navegación entre estados de la aplicación.

Sitio Web

Conjunto de páginas web interconectadas accesibles bajo un dominio, y cuyo propósito suele ser disponibilizar información, recursos y/o servicios relacionados con un tema o propósito específico con sus usuarios.

SQL (Structured Query Language)

Lenguaje estandarizado para la gestión y consulta de bases de datos relacionales. Permite la implementación de operaciones CRUD básicas, modificación de la estructura y los permisos de tablas relacionales, realizar peticiones complejas mediante “joins” y subconsultas, así como garantizar la integridad transaccional.

SSL (Socket Secure Layer)

Estándar de seguridad para establecer enlaces encriptados entre cliente y servidor en Internet, asegurando la transmisión privada y segura de información.

Stateless

Principio de diseño de sistemas cliente servidor donde cada petición debe contener la información necesaria para su procesamiento, sin que el servidor mantenga un contexto entre solicitudes.

URL (Uniform Resource Locator)

Es un texto que especifica la ubicación de un recurso multimedia en Internet. Suelen denominarse dirección web o enlaces.

Bibliografía

- [1] *The History of the Internet in a Nutshell*. <https://www.webfx.com/blog/web-design/the-history-of-the-internet-in-a-nutshell/>. Accessed: 06.12.2022.
- [2] *E-commerce turns 10*. <https://www.cnet.com/tech/services-and-software/e-commerce-turns-10/>. Accessed: 06.12.2022.
- [3] *What Was the First Thing Sold on the Internet?* <https://www.smithsonianmag.com/smart-news/what-was-first-thing-sold-internet-180957414/>. Accessed: 06.12.2022.
- [4] *Pasado, presente y futuro del comercio electrónico*. <https://www.empresaactual.com/historia-del-comercio-electronico/>. Accessed: 06.12.2022.
- [5] *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. <https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>. Accessed: 06.12.2022.
- [6] *Estadísticas de Cuota de Mercado y Uso de Shopify*. <https://kinsta.com/es/cuota-de-mercado-de-shopify>. Accessed: 06.12.2022.
- [7] *Son casi el 99 % de las empresas en Chile: Lo cifras detrás de las pymes, protagonistas en el debate previo a las primarias*. <https://www.emol.com/noticias/Economia/2021/07/13/1026565/cifras-pymes-chile-primarias.html>. Accessed: 13.12.2024.
- [8] *Las mipymes chilenas en el 2022*. https://obtienearchivo.bcn.cl/obtienearchivo?id=repositorio/10221/33318/1/N_25_22_Las_mipymes_chilenas_en_el_2022.pdf. Accessed: 13.12.2024.
- [9] *Agenda Digital: Una hoja de ruta para avanzar en el desarrollo digital del país*. <https://www.gob.cl/noticias/agenda-digital-una-hoja-de-ruta-para-avanzar-en-el-desarrollo-digital-del-pais/>. Accessed: 18.12.2024.
- [10] *Índice de Transformación Digital de Empresas 2018*. <https://www.ecommerceccs.cl/wp-content/uploads/2019/03/Indice-Transformaci%C3%B3n-Digital-en-Empresas-2018.pdf>. Accessed: 18.12.2024.

- [11] *Análisis descriptivo del impacto de la pandemia sobre las empresas en Chile.* <https://www.economia.gob.cl/wp-content/uploads/2021/07/Boletin-Analisis-descriptivo-del-impacto-de-la-pandemia-sobre-las-empresas-en-Chile-1.pdf>. Accessed: 18.12.2024.
- [12] *Estudio de Asech revela que el 91 % de los emprendedores se vio afectado producto de la pandemia.* <https://www.df.cl/df-lab/innovacion-y-startups/estudio-de-asech-revela-que-el-91-de-los-emprendedores-se-vio-afectado>. Accessed: 18.12.2024.
- [13] *Encuesta Impacto Pandemia.* <https://www.diariosustentable.com/wp-content/uploads/2021/03/graficos-encuesta-v3.pdf>. Accessed: 18.12.2024.
- [14] *Comercio electrónico evitó una caída histórica en las ventas del retail en 2020.* <https://www.ccs.cl/2021/01/29/comercio-electronico-evito-una-caida-historica-en-las-ventas-del-retail-en-2020>. Accessed: 18.12.2024.
- [15] *DIGITAL REPORT 2022: EL INFORME SOBRE LAS TENDENCIAS DIGITALES, REDES SOCIALES Y MOBILE.* <https://wearesocial.com/es/blog/2022/01/digital-report-2022-el-informe-sobre-las-tendencias-digitales-redes-sociales-y-mobile/>. Accessed: 06.12.2022.
- [16] *Mercado Libre anuncia inversión histórica en Chile.* <https://www.latercera.com/pulso/noticia/mercado-libre-anuncia-inversion-historica-en-chile/FREDUDYTABBIHFLFMUGI7U4T54/>. Accessed: 10.02.2024.