Pontificia Universidad Católica del Ecuador

Facultad De Ingeniería

Escuela de Sistemas



TEMA:

SISTEMA WEB PARA LA AUTOMATIZACIÓN DE LOS PRINCIPALES PROCESOS DE LOGÍSTICA DE UNA EMPRESA DE COURIER.

AUTOR:

DANIEL SEBASTIÁN BONILLA CARVAJAL

DIRECTOR:

INGENIERO. NICOLALDE RODRIGUEZ DAMIAN ANIBAL

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS DE INFORMACIÓN

QUITO DM, JUNIO DE 2024

AGRADECIMIENTOS

Agradezco a mis padres, Gustavo Bonilla y Bernardita Carvajal, por apoyarme, cuidarme y confiar en mi en cada momento de mi carrera. Han sido los pilares que han sostenido y me han dado el aliento en todo mi desarrollo académico.

A mi hermana, por brindarme su incondicional guía. Su sacrificio y dedicación han sido mi inspiración constante, y por eso, quiero compartir con ella este logro.

Finalmente, a mis amigos, mis compañeros en todo este viaje, gracias por los buenos momentos, las risas, los consejos, y esencialmente, por hacer de mis días en la universidad una aventura inolvidable. Que nuestra amistad perdure, irradie, y sea infinita, a la voz de un se vale de todo.

ÍNDICE

ÍNDICE DE ILUSTRACIONES Y TABLAS 6
INDICE DE TABLAS 6
INDICE DE ILUSTRACIONES 7
CAPÍTULO I: INTRODUCCIÓN 8
1. Marco de Referencia 8
1.1 Justificación 8
1.2 Planteamiento del problema 8
1.3 Objetivo general 10
1.4 Objetivos específicos 10
1.5 Antecedentes 10
1.6 Alcance 11
CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA 13
2 Marco teórico y conceptual 13
2.1 Marco Conceptual 13
2.1.1 Empresa Courier 13
2.1.2 Categorías de paquetes Courier 13
2.1.3. Desarrollo web 14
2.1.3.1 Frontend 14
2.1.3.2 Backend 15
2.1.3.3 Frameworks para construcción de UI 15
2.2 Marco Teórico 15
2.2.1 Metodologías de desarrollo de software 15
2.2.1.1 Concepto 15
2.2.1.2 XP 15
2.2.1.3 SCRUM 16
2.2.1.4 Kanban 16
2.2.1.5 Selección de la metodología de desarrollo 17
2.2.2 Arquitectura basada en microservicios 18
2.2.2.1 Concepto 18
2.2.2.2 API 18
2.2.2.3 API REST 18
2.2.2.4 Bus de eventos 19
2.2.3 Tecnologías y herramientas del desarrollo 19
2.2.3.1 Backend 19
2.2.3.1.1 Django REST framework 19

2.2.3.1.2 ASP.NET Web API 19
2.2.3.1.3 Java Spring Boot 20
2.2.3.1.4 Selección de herramienta de desarrollo de API REST 20
2.2.3.2 Frontend 21
2.2.3.2.1 Vue.js 21
2.2.3.2.2 React 21
2.2.3.2.3 Laravel 22
2.2.3.2.4 Selección de herramienta de desarrollo de frontend 22
2.2.3.3 Bases de datos 23
2.2.3.3.1 Concepto 23
2.2.3.3.2 Diagrama conceptual 23
2.2.3.3.3 Diagrama lógico 23
2.2.3.3.4 Diagrama físico 24
2.2.3.3.5 Selección de base de datos 24
CAPÍTULO III: METODOLOGÍA 26
3 Metodología de desarrollo aplicada 26
3.1 Planificación 26
3.1.1 Historias de usuarios 27
3.1.2 Definición de iteraciones 31
3.1.3 Plan de entregas 32
CAPÍTULO IV: DISEÑO 34
4.1 Diseño aplicado en la metodología 34
4.1.1 Diseño arquitectónico del sistema 34
4.1.2 Arquitectura basada en microservicios 35
4.2 Tarjetas CRC 36
4.3 Diagrama conceptual 40
4.4 Diagrama Físico de base de datos 42
4.5 Diseño de las interfaces de usuario 43
4.5.1 Interfaz de registro al sistema 43
4.5.1.1 Clientes 43
4.5.2 Interfaz de acceso al sistema 44
4.5.2.1 Clientes 44
4.5.2.2 Empleados 45
4.5.3 Interfaz ingreso prealertas 45
4.5.4 Interfaz de clasificación de prealertas y paquetes 46
4.5.5 Interfaz de historial de paquetes 46
4.5.6 Interfaz de visualización de paquetes 47

4.5.7 Interfaz de administración de empleados 47
4.5.8 Interfaz de administración de socios 48
4.5.9 Interfaz de reportes 48
CAPÍTULO V: DESARROLLO 50
5.1 Fase de iteraciones 50
5.1.1 Primera iteración 50
5.1.1.1 Historias 50
5.1.1.2 Tareas 50
5.1.2 Segunda iteración 51
5.1.2.1 Historias 51
5.1.2.2 Tareas 52
5.1.3 Tercera iteración 52
5.1.3.1 Historias 52
5.1.3.2 Tareas 52
5.2 Fase de pruebas 52
5.2.1 Primera iteración 53
5.2.1.1 Pruebas de aceptación 53
5.2.1.2 Prueba de carga servicio de autenticación 54
5.2.1.2.1 Escenario de prueba: 54
5.2.1.2.1 Resultados de la Prueba: 54
5.2.2 Segunda iteración 55
5.2.2.1 Pruebas de aceptación 55
5.2.2.2 Prueba de carga servicio de gestión de paquetes 64
5.2.2.2.1 Escenario de prueba: 64
5.2.2.2.1 Resultados de la Prueba: 64
5.2.3 Tercera iteración 64
5.2.3.1 Pruebas de aceptación 64
5.2.3.2 Prueba de carga servicio de consulta de reportes 66
5.2.3.2.1 Escenario de prueba: 66
5.2.3.2.1 Resultados de la Prueba: 66
CONCLUSIONES Y RECOMENDACIONES 67
Conclusiones: 67
Recomendaciones 68
BIBLIOGRAFÍA 70

ÍNDICE DE ILUSTRACIONES Y TABLAS

INDICE DE TABLAS

Tabla 1: Categorías de paquetes Courier 13
Tabla 2: Comparativa de metodologías 17
Tabla 3: Comparativa de herramientas para backend 20
Tabla 4: Comparativa de herramientas para frontend 22
Tabla 5: Comparativa de bases de datos 24
Tabla 6: Listado de historias de usuario 27
Tabla 7: Historia de usuario Nº1 – Autenticación en el sistema: 28
Tabla 8: Historia de usuario Nº2 – Administración de empleados 28
Tabla 9: Historia de usuario Nº3 – Administración de prealertas 29
Tabla 10: Historia de usuario Nº4 – Administración de paquetes 29
Tabla 11: Historia de usuario Nº5 – Generación reportes localidad 30
Tabla 12: Historia de usuario Nº6 – Generar reportes de clientes con mayor volumen de compras 30
Tabla 13: Definición de iteraciones 31
Tabla 14: Plan de entregas 32
Tabla 15: Tarjeta CRC - Cliente 36
Tabla 16: Tarjeta CRC - Pedido 37
Tabla 17: Tarjeta CRC - Tienda 38
Tabla 18: Tarjeta CRC - Carrier 38
Tabla 19: Tarjeta CRC - Empleado 38
Tabla 20: Tarjeta CRC - Paquete 39
Tabla 21: Tarjeta CRC - Usuario 40
Tabla 22: Roles de Usuario 51
Tabla 23 Caso de prueba RF01_P01 53
Tabla 24 Caso de prueba RF01_P02 53
Tabla 25 Caso de prueba RF01_P03 54
Tabla 26 Caso de prueba RF03_P01 55
Tabla 27 Caso de prueba RF03_P02 55
Tabla 28 Caso de prueba RF03_P03 56
Tabla 29 Caso de prueba RF03_P04 56
Tabla 30 Caso de prueba RF03_P05 57

Tabla 31 Caso de prueba RF03_P06 58
Tabla 32 Caso de prueba RF04_P01 58
Tabla 33 Caso de prueba RF04_P02 59
Tabla 34 Caso de prueba RF04_P03 59
Tabla 35 Caso de prueba RF04_P04 60
Tabla 36 Caso de prueba RF02_P01 61
Tabla 37 Caso de prueba RF02_P02 61
Tabla 38 Caso de prueba RF02_P03 62
Tabla 39 Caso de prueba RF02_P04 62
Tabla 40 Caso de prueba RF02_P05 63
Tabla 41 Caso de prueba RF05_P01 64
Tabla 42 Caso de prueba RF06_P01 65

INDICE DE ILUSTRACIONES

Ilustración 1: Arquitectura de microservicios utilizada en el desarrollo del proyecto 36
Ilustración 2: Diagramas conceptuales del sistema 41
Ilustración 3: Diagramas físicos de las bases de datos 42
Ilustración 4: Interfaz de registro de clientes 43
Ilustración 5: Interfaz de acceso al sistema para clientes 44
Ilustración 6: Interfaz de acceso al sistema para empleados 45
Ilustración 7: Interfaz de ingreso de prealertas 45
Ilustración 8: Interfaz de clasificación de prealertas y paquetes 46
Ilustración 9: Interfaz de historial de paquetes 46
Ilustración 10: Interfaz de visualización de paquetes 47
Ilustración 11: Interfaz de administración de empleados 47
Ilustración 12: Interfaz de administración de socios 48
Ilustración 13: Interfaz de reportes 48

CAPÍTULO I: INTRODUCCIÓN

1. Marco de Referencia

1.1 Justificación

Desde el inicio de la pandemia de Covid-19 en marzo 2019, el sector de servicios Courier ha enfrentado a una creciente demanda de eficiencia y transparencia. Este aumento responde a un entorno cada vez más digital y competitivo, agravado además por la amplificación en las ventas de productos por redes sociales. Ante esta situación, ofrecer un servicio Courier que únicamente cumpla con su propósito ya no es suficiente.

Así lo valida Alejandro Montalvo, representante de marketing del Courier DHL, en una entrevista realizada por El universo, donde expresa abiertamente cómo, en búsqueda de cumplir con tiempos de entrega más bajos y ágiles para sus clientes, la empresa cuenta con dos aviones propios que son indispensables para cumplir con las exigencias del mercado actual (Montalvo, 2021).

Este contexto ha obligado a las empresas a ofrecer soluciones innovadoras para seguir siendo competitivas en el mercado actual, ocasionando que la gestión de envíos de manera rápida y efectiva se transforme en una prioridad tanto para las empresas de Courier como para sus clientes.

En este escenario, surge la necesidad de optimizar los procesos en el sector Courier, lo cual constituye la base y el objetivo principal de este trabajo, el desarrollo de una solución tecnológica que permita un registro y seguimiento de prealertas, además de un control detallado del estado de los envíos. Esto mejorará significativamente la calidad y rapidez del servicio, brindando una experiencia más satisfactoria para los clientes, y a las empresas, una ventaja competitiva en el mercado.

1.2 Planteamiento del problema

Desde el 15 de junio de 2021, el comité de comercio exterior del Ecuador tomó la decisión de eliminar por completo la tarifa de USD 42 para la importación de paquetes a través del sistema 4x4, operado a través de empresas Courier. Esta medida fue recibida con entusiasmo por parte de la población en general, ya que abrió un gran mercado donde era posible traer cosas del exterior para

uso personal sin la imposición de impuestos ni cargos adicionales.

Según datos del Servicio Nacional de Aduanas, a pesar de la pandemia Covid-19, los envíos y movimientos de las empresas Courier en la categoría B aumentaron significativamente, pasando de 635.000 envíos contabilizados y con un registro de $68,64 millones a 880.000 envíos por $102,75 millones en el tercer trimestre del 2021 (Servicio Nacional de Aduanas, 2021).

Este suceso supuso un gran impulso para las empresas internacionales especializadas en servicios de Courier, que rápidamente se adentraron al mercado con costos sumamente elevados en la entrega de paquetes. Al mismo tiempo, surgieron varios emprendedores y empresas locales que intentaban competir en este sector, con costos significativamente más bajos, pero sin la eficiencia, calidad y seguridad que ofrecía la empresa internacional ya establecida.

En base a esta necesidad se realiza el presente trabajo de titulación, ya que se constató como las empresas locales no manejan un registro y control adecuado de los pedidos de los clientes, desempeñando sus actividades en archivos de texto plano u hojas de cálculo, lo que a menudo resulta en conflictos en la gestión de entregas y recepción de los paquetes. Al mismo tiempo, estas empresas no desarrollan un seguimiento detallado de la ubicación de donde se encuentra su paquete, lo que genera inseguridad e insatisfacción al cliente final.

En función de esta problemática se plantea la siguiente pregunta principal de investigación:

- ¿Cómo se puede integrar la tecnología en una empresa de servicios Courier?
- Y las siguientes preguntas secundarias:
- ¿Cuáles son los beneficios potenciales de la implementación de tecnología en términos de eficiencia operativa y satisfacción del cliente?
- ¿Cuáles son las tecnologías más relevantes y efectivas que pueden ser integradas en una empresa de servicios Courier?
- ¿Cuáles son las mejores prácticas y ejemplos de empresas de Courier que han tenido éxito en la implementación de tecnología en sus operaciones?

1.3 Objetivo general

Diseñar una aplicación web que permita la gestión de los principales procesos logísticos de una empresa de Courier, incluyendo el registro y seguimiento detallado de prealertas.

1.4 Objetivos específicos

- Identificar los requerimientos del usuario relacionados al levantamiento de los principales procesos logísticos, características, funciones o servicios necesarios de la empresa.
- Determinar las características funcionales y de interfaz de usuario para el aplicativo web, que permita al operador de bodega mantener un control y gestión efectivo de los diferentes paquetes manejados por la empresa de Courier.
- Construir un prototipo funcional de una aplicación web que facilite la gestión de paquetes mediante una plataforma intuitiva y accesible desde cualquier dispositivo.

1.5 Antecedentes

Esta sección del presente trabajo hace referencia a los diferentes estudios y tesis que tienen enfoques similares a la presente investigación, su análisis y comparativa apoya a obtener un estudio enfocado y respaldado sobre la aplicación de la metodología de “extreme programming” en un entorno real, así como la definición de los requerimientos y procesos principales de una empresa de Courier.

A. Nicolalde Freire David. (2013). Desarrollo de una aplicación web para la gestión de movimiento de paquetes para empresas de Courier aplicación a un caso de estudio. Tesis para optar el Título Profesional de Ingeniero en sistemas informáticos y de computación. ESCUELA POLITÉCNICA NACIONAL

Esta tesis se enfoca en desarrollar un proyecto que permita a las empresas de Courier automatizar sus procesos y desarrollar un manejo de información útil mediante reportes e información confiable.

El aporte de este antecedente en el presente trabajo de titulación es la guía y respaldo del levantamiento de requerimientos dentro de una empresa enfocada en Courier, lo cual permite a este

estudio analizar y comparar los requerimientos y necesidades definidas.

B. Delzo Gutierrez, Ennis Martin. (2018). Desarrollo de un sistema de información web basado en la metodología extreme programming para mejorar la gestión editorial del fondo editorial de la Universidad Continental. UNIVERSIDAD NACIONAL DEL CENTRO DE PERU.

Este estudio se basa en desarrollar un sistema web para mejorar el proceso de Gestión Editorial del Fondo Editorial de la Universidad Continental, de tal manera que se reduzcan los tiempos de procesamiento de la información.

Este antecedente marca los diferentes valores y recursos de la metodología XP, además de presentar varios procedimientos a utilizar para la metodología de desarrollo flexible.

C. Diego Hallo, Jaime Pinto. (2014). Desarrollo de una aplicación para la gestión del mercado inmobiliario en la ciudad de Quito usando la plataforma iphone. ESCUELA POLITÉCNICA NACIONAL.

Este trabajo académico se centra en desarrollar una aplicación para la ciudad de Quito para la compra o alquiler de inmuebles, esto con la finalizar de realizar compras y ventas de manera ágil por la web, interviniendo como intermediario en la comunicación entre vendedor y el comprador.

El aporte de este antecedente en este trabajo radica en la guía proporcionada por el apartado de informes y estadísticas, la cual destaca la necesidad de desarrollar un sistema de información que presente informes y estadísticas útiles en la toma de decisiones para el usuario, o en este caso, la empresa Courier.

1.6 Alcance

El presente trabajo abarca el análisis, diseño y desarrollo de un prototipo para una aplicación web que permita la gestión adecuada de los principales procesos de logística de una empresa de Courier para mejorar su eficiencia y efectividad al contratar este servicio.

Bajo este enfoque se desarrolla un análisis de las principales funciones clave del negocio, el diseño de la arquitectura del sistema, y el desarrollo de una solución que apoye al negocio utilizando tecnologías web modernas.

El proyecto no incluye la instalación e implementación del sistema en un entorno de producción. Por lo tanto, el alcance de la aplicación será el siguiente:

- El ingreso a la aplicación con una sesión de usuario y contraseña.
- La creación y visualización de prealertas.
- La gestión del envió de paquetes en conjunto con su estado.
- La visualización de reportes y analítica de los paquetes.

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2 Marco teórico y conceptual

2.1 Marco Conceptual

2.1.1 Empresa Courier

Según la ley ecuatoriana, en el Reglamento para la aplicación del impuesto a la salida de divisas (2008), las empresas Courier se clasifican en dos tipos: mensajería expresa y Courier propiamente dichos (Servicio Nacional de Aduanas del Ecuador, 2022):

> "1. Mensajería expresa o correos rápidos: Son sociedades reguladas por la Superintendencia de Compañías, que prestan el servicio de envío o traslado de encomiendas, paquetes o sobres al exterior.
>
> 2. Courier propiamente dichos: Son sociedades reguladas por la Superintendencia de Compañías, que prestan el servicio de transferencias, traslado o envío, y recepción de divisas, paquetes, encomiendas y sobres, desde y hacia el exterior"

En definitiva, se define a este servicio como: "el envío de paquetes y/o bultos a través del Operador Público del Servicio Postal Oficial del Ecuador o de una Empresa Courier" (Servicio Nacional de Aduana del Ecuador, 2022). El envío, costo y valor de estos paquetes puede variar dependiendo del servicio, trabajando tanto a nivel nacional como internacional, pero siguiendo siempre las regulaciones en las normas aduaneras y bajo las categorías según el tipo de paquete.

2.1.2 Categorías de paquetes Courier

Los paquetes a enviar por las empresas están clasificados por el servicio nacional de aduana del Ecuador bajo 6 categorías, enlistadas de A – G. A continuación, se presenta un cuadro realizado por la Aduana del Ecuador con respecto a las especificaciones y limitaciones de cada una de estas categorías (SENAE, 2020).

Tabla 1:

Categorías de paquetes Courier

CATEGORÍAS DE PAQUETES COURIER		
CATEGORÍAS	**DESCRIPCIÓN**	**IMPUESTO**
A	Correspondencia como documentos, cartas, periódicos, fotografías, etc.	Libre de tributos
B	Paquetes hasta 4kg y USD $400	Libre de tributos
C	Paquetes hasta 100kg y USD $5000 (**Requiere Documento de control previo de acuerdo al producto importado)**	Arancel: % depende del producto IVA: 12% FODINFA: 0.5%
D	Prendas de vestir, textiles confeccionados y calzado hasta 20kg y $2.000 (Requiere Documento de control previo, excepto la primera vez al año con un monto de hasta $500)	Textiles: 10% ADV + USD $5,5 x KG Calzado: 10% ADV + USD $ 6 x Par IVA: 12% FODINFA: 0.5%
E	Medicinas (con receta médica)	FODINFA: 0.5%
	Equipos ortopédicos, órganos, tejidos, ect. Sin fines comerciales	Libre de tributos Art. 125 COPCI
F	Libros de lectura	FODINFA: 0.5%
	Equipos de Computación y sus partes	IVA: 12% FODINFA: 0.5%
G	Paquetes hasta 4kg y Valor FOB de un (1) Salario Básico Unificado Registro del familiar del migrante https://registronucleofamiliar.cancilleria.gob.ec/	Libre de tributos

Nota: La tabla presenta un resumen de las diferentes categorías de paquetes Courier en conjunto con el impuesto asociado al mismo. (Aduana del Ecuador, 2021). Recuperado de: https://www.aduana.gob.ec/servicio-al-ciudadano/envios-courier-postal/

2.1.3. Desarrollo web

El desarrollo web, según Barba (2015) es “la programación necesaria para la construcción del sitio web. Se divide en dos partes que pueden estar o no conectadas, la parte del cliente y la parte del servidor” estas dos partes se conocen comúnmente como backend y frontend, las cuales se encuentran en constante comunicación para presentar al usuario la información necesaria para el cumplimiento de sus objetivos con la página.

2.1.3.1 Frontend

Este término hace referencia a los diferentes elementos de la página web con los que el usuario puede interactuar, estos elementos son los denominados componentes de la interfaz y pueden incluir

desde botones y listas, hasta menús y formularios. Los desarrolladores utilizan esta parte del desarrollo web para gestionar las acciones de los usuarios (Amazon AWS, 2020).

2.1.3.2 Backend

El backend es una parte del desarrollo web que se encarga de procesar los datos y la lógica del negocio a fin de hacer funcionar a un sitio u aplicación web. Esta ocurre de forma no perceptible para los usuarios y sus principales acciones se basan en recopilar y procesar solicitudes http para realizar funcionalidades adicionales o acceder a información relevante en las bases de datos (Amazon AWS, 2020).

2.1.3.3 Frameworks para construcción de UI

Un framework o también conocido como marco de trabajo, es una plantilla que recopila herramientas y componentes para crear interfaces de usuario de forma rápida y eficiente. Existen varias de estas soluciones que aceleran el desarrollo y reducen los costos de programación al manejar un conjunto de reglas y estrategias que mantienen el código del programa robusto y más limpio (Sandoval, 2023).

2.2 Marco Teórico

2.2.1 Metodologías de desarrollo de software

2.2.1.1 Concepto

Una metodología de desarrollo de software es una agrupación de técnicas y métodos utilizados para estructurar, planear y controlar el desarrollo de una solución de software. Son de suma utilidad para los desarrolladores, ya que establecen parámetros para la comunicación y trabajo en equipo, optimizando el tiempo y el coste del desarrollo (GooApps, 2022).

2.2.1.2 XP

Esta es una de las clasificadas como metodologías de desarrollo ágil, que permiten a los equipos ejecutar proyectos de creación de software de manera rápida, flexible y controlada. Su principal enfoque es generar productos de calidad de manera constante para adaptarse a los cambios

en los requerimientos de los clientes (Bello, 2021).

Esta metodología utiliza principalmente la técnica denominada “historias de usuario” para recopilar y determinar los requisitos del software, contando así con 5 fases en total, las cuales son; planificación, diseño, codificación, pruebas y lanzamiento.

En cada una de estas fases interactúan seis roles clave que define XP para el éxito de un proyecto: el cliente, encargado de escribir y priorizar las historias de usuario; el coach, que guía al equipo en las buenas prácticas de XP; el programador, encargado de escribir el código del programa; el rastreador, responsable de monitorear el desarrollo y detectar posibles errores; el probador o tester, cuya tarea es realizar probas continuas para asegurar la calidad del software; y el pronosticador o Doomsayer, encargado de definir los posibles riesgos del proyecto para crear un software más robusto (Marvin, 2020).

2.2.1.3 Scrum

Scrum es un marco de trabajo utilizado por equipos de trabajo para entregar proyectos de manera eficiente, se centra en entregas periódicas de avances del proyecto denominadas Sprints. En cada Sprint se busca resolver problemas y requerimientos determinados por el cliente modificando el software para adaptarlo a sus necesidades.

Esta metodología define roles específicos que son vitales para el desarrollo de proyectos, cada uno con sus respectivas responsabilidades y tareas: el Product Owner, responsable de expresar la voz del cliente durante todo el desarrollo y de asegurarse que el equipo trabaje en las tareas que proporcionen el mayor valor al negocio; el Scrum Master, encargado de guiar al equipo en la metodología Scrum promoviendo el aprendizaje continuo; y el Scrum Team, el equipo encargado de desarrollar y entregar cada uno de los productos e incrementos de cada sprint. (Hurtado, 2021)

2.2.1.4 Kanban

Esta metodología de desarrollo ágil se centra en la reducción de tiempo de iteración entre actividades, buscando en todo momento mejorar la calidad del desarrollo de software. La palabra Kanban tiene relación con dos palabras japonesas, kan, que significa “visual”, y ban, “tarjeta”, esto

debido a que la metodología utiliza principalmente tarjetas como recurso gráfico para para la gestión de equipos (Bermejo, 2011).

Para clasificar cada una de las tareas que el equipo tiene que desarrollar, Canvan propone una estructura flexible de tres fases: Todo (por hacer), para aquellas tareas que aún no se han realizado pero que se han definido como requerimientos por el cliente; In Progress (en progreso), sobre aquellas tareas que al menos un miembro del equipo se encuentra activamente trabajando sobre ellas; y Done (hecho), fase donde se desplaza una tarea cuando esta lista para su revisión o entrega. (Fadeke, 2023)

2.2.1.5 Selección de la metodología de desarrollo

A continuación, se presenta un cuadro comparativo entre las metodologías antes descritas, con los parámetros principales de: participación del cliente, tamaño recomendado del equipo de trabajo, recursos a utilizar, frecuencia de entregables y fases recomendadas.

Tabla 2:

Comparativa de metodologías

Metodología/ Parámetros	XP	Scrum	Kanban
Participación del cliente	Alta	Alta	Media
Equipo de trabajo	De 3 a 9 personas	De 5 a 9 personas	Preferencia a equipos de gran tamaño
Recursos por utilizar	Historias de usuario, plan de entregas, plan de iteraciones.	Historias de usuario, backlog del producto, backlog del sprint	Tablero Kanban, tarjetas de trabajo, WIP limits, análisis de métricas del flujo de trabajo.
Frecuencia de entregables	Entregas cortas (1-3 semanas) software funcional al final de cada iteración	Sprints de longitud fija (2-4 semanas)	Flujo constante de entregas a medida que se van completando las tareas.
Fases recomendadas	Exploración, planificación, iteraciones,	Planificación, desarrollo Sprint, revisión Sprint,	Visualización, establecimiento de límites de trabajo,

	producción, mantenimiento, muerte del proyecto	retrospectiva e incremento de sprint.	gestión del flujo de trabajo, optimización continua.

Considerando estos factores, y analizando los antecedentes seleccionados para el desarrollo de este trabajo, se determina que la mejor alternativa es la metodología Extreme Programming (XP), por ofrecer un desenvolvimiento más intuitivo y dedicado para equipos pequeños de trabajo. Este aspecto apoyará de manera positiva en el desarrollo de este trabajo, ya que el sistema requiere una entrega de valor continua con cliente, donde el mismo debe estar involucrado en sus etapas de desarrollo. Además, esta metodología establece una frecuencia de entregas cortas, claras y sencillas, principios fundamentales que permitirán agregar valor al proyecto en cortos periodos de tiempo.

2.2.2 Arquitectura basada en microservicios

2.2.2.1 Concepto

Al crear una solución de software, es indispensable plantear una descripción de la estructura del software, sus componentes y comunicación, esto es un procedimiento denominado diseño de software. Y aunque existen varios diseños que se adecuan a dependiendo de las necesidades del proyecto de software, este trabajo se centra en la arquitectura basada en microservicios. Esta arquitectura plantea una solución donde se requiere la combinación de diferentes servicios intercomunicados entre sí, a fin de realizar tareas complejas independientemente y formar aplicaciones de software en distintos sistemas (Red Hat, 2023).

2.2.2.2 API

Una API o también conocida como interfaz de programación de aplicaciones, es un software que recopila diferentes conjuntos reglas para intercomunicar una o varias aplicaciones mediante solicitudes y respuestas (IBM, 2022).

2.2.2.3 API REST

Existen varios tipos de API que de distinta forma intercomunican las aplicaciones, una de ellas son las API REST, donde generan solicitudes HTTP mediante las cuales ejecutan funciones

establecidas para la base de datos, es decir, crear, leer, actualizar y eliminar registros, para después devolver los datos de salida al cliente (Amazon AWS, 2020).

2.2.2.4 Bus de eventos

Con la finalidad de transmitir información entre los diferentes microservicios que componen la arquitectura, el bus de eventos es un componente que actúa como un canal de comunicación entre los diferentes microservicios. Este puede ser configurado y programado para reaccionar frente a peticiones de creación, edición, eliminación o consulta, y puede direccionar estos eventos, a uno o más microservicios (Amazon AWS, 2021).

2.2.3 Tecnologías y herramientas del desarrollo

2.2.3.1 Backend

2.2.3.1.1 Django REST framework

Django es un framework de desarrollo web que permite crear aplicaciones de forma rápida y eficiente, este utiliza una arquitectura basada en componentes y módulos reutilizables, ofreciendo a los desarrolladores una estructura clara y coherente para organizar su código. Su principal finalidad es la reducción significativa del tiempo de desarrollo web y aceleración del proceso de implementación de proyectos en línea (Amazon AWS, 2023).

Por su parte, Django REST framework es una herramienta de código abierto creada en Python para la creación de web APIs utilizando el framework web Django (Christie, 2024).

2.2.3.1.2 ASP.NET Web API

ASP.NET Web API es un framework utilizado para la creación de APIs utilizando los servicios http. Según la documentación proporcionada en la página de Microsoft .Net (Microsoft, 2024) “ASP.NET facilita la creación de servicios que lleguen a una amplia gama de clientes, incluidos exploradores y dispositivos móviles mediante experiencias web modernas” Este framework cuenta con las opciones clásicas de un API REST, enrutamiento, autenticación y serialización.

2.2.3.1.3 Java Spring Boot

Java Spring Boot es un marco de trabajo enfocado en la creación de aplicaciones para su ejecución en una máquina virtual Java (JVM). Su configuración y desarrollo permite la creación de microservicios y aplicaciones web de manera simplificada y rápida (IBM, 2023).

2.2.3.1.4 Selección de herramienta de desarrollo de API REST

A continuación, se presenta un cuadro comparativo entre las herramientas de desarrollo antes descritas, con los parámetros principales de: lenguaje de programación, seguridad, soporte, integración con tecnologías, costo y capacidad de integración con herramientas externas.

Tabla 3:

Comparativa de herramientas para backend

Herramienta/ Parámetros	Django REST Framework	ASP.NET Web API	Java Spring Boot
Lenguaje de programación	Python	C#	Java
Seguridad	Funciones de autenticación y autorización integradas.	Autenticación y autorización, integración con servicios de seguridad de .NET.	Autenticación y autorización, integración con servicios de seguridad de Java.
Soporte	Actualizaciones frecuentes, y mantenido principalmente por comunidad.	Soporte y mantenimiento son proporcionados por Microsoft	Actualizaciones frecuentes, y mantenido principalmente por comunidad.
Integración con otras tecnologías	Integración con sistema Django	Integración con sistema .NET	Integración con el ecosistema Spring
Costo	Gratuito y de código abierto	Gratuito, parte de la plataforma .NET	Gratuito y de código abierto
Capacidad de integración con herramientas externas	Facilidad de integración con otras funcionalidades y adaptable a proyectos externos a la herramienta.	Requiere conocimientos en prácticas del entorno .NET. E integración exclusiva con herramientas del	Requiere conocimientos en prácticas del entorno Spring. Necesidad de configuración e instalación de

		entorno.	entorno Spring.

Tomando en consideración estos parámetros, y analizando varios de los antecedentes seleccionados para el desarrollo de este trabajo, se determina que la mejor alternativa es el framework de desarrollo Django REST framework, ya que cuenta con facilidad de integración con otras herramientas como ‘swagger’ para la creación automática de documentación o ‘testeo’ para el desarrollo de pruebas unitarias y es ampliamente configurable frente a las otras herramientas. Estas capacidades aportarán valor al proyecto, ya que esta desarrollado con una expectativa escalable, donde a medida que los requerimientos de la empresa Courier vayan aumentando, el proyecto pueda seguir creciendo y aportando valor al negocio.

2.2.3.2 Frontend

2.2.3.2.1 Vue.js

Vue.js es uno de los framework de desarrollo web más utilizados en la actualidad. Según el último informe de Stack Overflow, Vue.js es considerado en 2023 el cuarto frameworks con más popularidad y uso entre la comunidad (Stack Overflow, 2023).

Este resultado no es de sorprender, ya que este framework de desarrollo web ofrece a los desarrolladores una sintaxis intuitiva y clara respecto a otras tecnologías, además de ser liviano y de carga rápida en comparación a otros frameworks además de contar con una amplia gama de bibliotecas y herramientas de desarrollo a su disposición (Barragan, 2021).

2.2.3.2.2 React

React, por otra parte, se corona como el framework de desarrollo más utilizado en la actualidad, según el último informe de Stack Overflow (2023).

Y aunque no es considerado como un framework de desarrollo como tal, sino más bien como una librería de Javascript, esta ha sido creada por Facebook como una herramienta para crear interfaces de usuario (UI) y llamadas componentes utilizando una arquitectura Modelo-Vista-Controlador (MVC).

2.2.3.2.3 Laravel

Laravel es un framework que utiliza PHP para la construcción de aplicaciones web, no obstante, aunque ha sido colocado en la parte de frontend para este estudio, Laravel tiene una concentración más amplia en herramientas para el backend.

Sin embargo, eso no lo ha limitado a ser uno de los framework para desarrollo de aplicaciones web más populares de la actualidad, esto debido a la cantidad de páginas web que se encuentran utilizando el lenguaje PHP en la actualidad (Cristancho, 2022).

2.2.3.2.4 Selección de herramienta de desarrollo de frontend

A continuación, se presenta un cuadro comparativo entre las herramientas de desarrollo antes descritas, con los parámetros principales de: lenguaje de programación, soporte, rendimiento, integración con otras herramientas y bibliotecas, facilidad de uso y curva de aprendizaje.

Tabla 4:

Comparativa de herramientas para frontend

Herramienta/ Parámetros	Vue.js	React	Laravel
Lenguaje de programación	JavaScript	JavaScript	PHP
Soporte	Actualizaciones frecuentes, y amplia comunidad de desarrolladores	Actualizaciones frecuentes, y amplia comunidad de desarrolladores	Actualizaciones frecuentes, y soporte oficial.
Rendimiento	Utilización de virtual DOM para optimizar el renderizado de la interfaz de usuario.	Utilización de virtual DOM para optimizar el renderizado de la interfaz de usuario.	Rendimiento escalable y optimizable mediante programación.
Integración con herramientas	Variedad de bibliotecas y herramientas de JavaScript para funcionalidades web.	Variedad de bibliotecas y herramientas de JavaScript para funcionalidades web.	Herramientas moderadamente más limitadas, enfocado más en el backend.

Facilidad de uso	Sintaxis intuitiva y flexibilidad en el manejo de scripts t depuración del código	Curva de aprendizaje moderada, necesidad de comprensión de JSX.	Curva de aprendizaje moderada alta, se necesita conocimiento en PHP y desarrollo backend.

Tomando en consideración estos parámetros, se escogió a Vue.js como la mejor alternativa para desarrollar el frontend de la aplicación web, ya que cuenta con una gran cantidad de bibliotecas compatibles y utilizables para el desarrollo de interfaces en JavaScript, las cuales facilitarán la interacción con la plataforma de envíos y seguimientos de paquetes tanto de clientes o empleados del sistema.

2.2.3.3 Bases de datos

2.2.3.3.1 Concepto

Una base de datos es la recopilación de datos bajo una estructura u organización que se almacenan de forma digital, están comúnmente controladas por un sistema de gestión de bases de datos (DBMS), los cuales permiten consultar y manipular los datos, de forma que sirve como interfaz entre la base de datos y sus programas (Oracle, 2018).

2.2.3.3.2 Diagrama conceptual

Un diagrama conceptual de la base de datos es una descripción de alto nivel del contenido de información de la base de datos, este representa un esquema completo de la base de datos, sus entidades y sus relaciones entre sí (Corcuera, 2018).

2.2.3.3.3 Diagrama lógico

El diagrama lógico de los datos amplia el diagrama conceptual, proporcionando el contenido de cada entidad y expone los diferentes tipos de conexiones que existen en el diagrama. Este ayuda a estableces lo requerimientos del negocio y proporciona una vista más detallada de la solución (Amazon AWS, 2022).

2.2.3.3.4 Diagrama físico

Un diagrama físico es la representación del modelo de base de datos que detalla la estructura de las diferentes tablas, sus componentes, sus relaciones, claves primarias y foráneas. Este es el más cercano al DBMS por lo que sus características dependerán del mismo. Estos se pueden crear a partir de un diseño lógico y se puede transformar en un diseño de base de datos real (IBM, 2021).

2.2.3.3.5 Selección de base de datos

En esta sección, se presenta un cuadro comparativo entre las diferentes bases de datos seleccionadas como opciones para el desarrollo de este trabajo de titulación, con los parámetros principales de: modelo de datos, seguridad, compatibilidad, soporte y costo

Tabla 5:

Comparativa de bases de datos

DBMS/ Parámetros	Firestore	SQL Server	MySQL
Modelo de datos	NoSQL (Documentos, Realtime Database)	Relacional SQL	Relacional SQL
Seguridad	Autenticación y autorización personalizadas	Múltiples capas de seguridad, roles y permisos	Múltiples capas de seguridad, roles y permisos
Compatibilidad	SDKs para varios lenguajes de programación	Integraciones con aplicaciones .NET	Amplia compatibilidad con lenguajes y plataformas
Soporte	Documentación oficial, foros de la comunidad, soporte técnico y servicios de consultoría.	Documentación oficial, foros de la comunidad, soporte técnico y servicios de consultoría.	Gran cantidad de foros y comunidad, soporte comercial.
Costo	Plan gratuito (Spark), Plan prepago (blaze)	Versión Desarrollo gratis, planes con licenciamiento	Gratuito y de código abierto

En base a estos factores, y analizando varios de los antecedentes seleccionados para el desarrollo de este trabajo, se determinó que la mejor alternativa es MySQL, por ofrecer una mayor compatibilidad con lenguajes y plataformas, además de ser una versión de código abierto con múltiples capas de seguridad y que permitirá al sistema manejar de forma efectiva y eficiente la información de envíos y seguimiento de paquetes.

CAPÍTULO III: METODOLOGÍA

3 Metodología de desarrollo aplicada

Tal como se presentó en la Tabla 1.4 del marco conceptual, la metodología XP o Extreme Programming fue la seleccionada para este estudio, ya que su forma de trabajo se enfoca en la entrega de producto de calidad incluso en equipos de trabajo tamaño reducido, así como en su notable flexibilidad en cuanto a la entrega y desarrollo de entregables.

La implementación de la arquitectura XP en este trabajo desarrolla las cuatro primeras fases de la metodología, empezando por la fase de planificación, donde se realizaron entrevistas con el equipo de trabajo para recolectar requisitos y definir historias de usuario.

Posteriormente en la fase de diseño, se crearon modelos e interfaces simples para una interacción efectiva del usuario, además se propuso una arquitectura que permite la escalabilidad entre interacciones y nuevas funcionalidades según los nuevos requerimientos del negocio.

En la fase de codificación, se llevaron a cabo cada una de las iteraciones del sistema, donde por cada una de las historias de usuario se describieron cada una de las tareas realizadas.

Finalmente, se ejecutó la fase de pruebas, donde se diseñaron las pruebas de aceptación basadas en las historias de usuario. Asimismo, se realizaron pruebas de carga para cada uno de los servicios, planteando un escenario de prueba y comprobando si los resultados fueron los esperados y cumplían con las expectativas.

3.1 Planificación

En esta fase, se establecen las bases del proyecto mediante la recopilación de requerimientos y necesidades del cliente documentadas en historias de usuario. La metodología XP fue de gran apoyo en la realización de para este caso, ya que facilitó la realización de entrevistas al equipo de trabajo de una pequeña empresa Courier.

El objetivo principal de estas entrevistas fue obtener una comprensión profunda del proceso de negocio y de sus principales componentes a ser automatizados, identificando de esta manera los

requerimientos del sistema. De forma que, al recompilar las ideas y comentarios por parte del usuario, se definieron las funcionalidades requeridas sin necesidad de utilizar lenguaje técnico o diseños complejos.

Al finalizar esta fase, se decidió que la estructura del sistema estaría constituida por módulos, ya que de esta forma el desarrollo del proyecto sería más escalable e intuitivo.

Los módulos por desarrollar son los siguientes:

- Autenticación y acceso.
- Administración de empleados.
- Gestión de prealertas.
- Gestión de paquetes.
- Reportes y analítica

3.1.1 Historias de usuarios

Para el sistema de gestión de procesos Courier, se determinan las siguientes historias de usuarios, según se muestra en la tabla 6:

Tabla 6:

Listado de historias de usuario

Nombre de la historia de usuario	Módulo
Autenticarse en el sistema	Autenticación y acceso
Administración de prealertas	Gestión de prealertas
Administración de paquetes	Gestión de paquetes
Administración de empleados	Administración de empleados
Generación de reporte de clientes con mayor cantidad de paquetes	Reportes y analítica
Generación de reportes de clientes con mayor volumen de compras	

En cooperación con los distintitos trabajadores de la empresa Courier se obtuvieron un total de 6 historias de usuario, que representan los procesos principales del sistema y sus requerimientos. A continuación, se detallan cada una de las historias.

Tabla 7:

Historia de usuario N°1 – Autenticación en el sistema:

<table>
<tr><th colspan="2">Historia de Usuario</th></tr>
<tr><td>Numero: RF01</td><td>Usuario: Administrador, cliente, empleado</td></tr>
<tr><td colspan="2">Nombre de la historia: Autenticación en el sistema</td></tr>
<tr><td>Prioridad del negocio: Alta</td><td rowspan="2">Riesgo en Negocio: Alta</td></tr>
<tr><td>Puntos estimados: 1</td></tr>
<tr><td colspan="2">Programador responsable: Daniel Bonilla</td></tr>
<tr><td colspan="2">Descripción:
Los usuarios pueden acceder mediante sus credenciales.
Los clientes pueden crear una cuenta y acceder a la información de sus paquetes</td></tr>
<tr><td colspan="2">Observaciones:
La sesión expira después de 1 hora de inactividad para garantizar la seguridad.</td></tr>
</table>

Tabla 8:

Historia de usuario N°2 – Administración de empleados

<table>
<tr><th colspan="2">Historia de Usuario</th></tr>
<tr><td>Numero: RF02</td><td>Usuario: Administrador</td></tr>
<tr><td colspan="2">Nombre de la historia: Administración de empleados</td></tr>
<tr><td>Prioridad del negocio: Media</td><td rowspan="2">Riesgo en Negocio: Media</td></tr>
<tr><td>Puntos estimados: 0.4</td></tr>
<tr><td colspan="2">Programador responsable: Daniel Bonilla</td></tr>
<tr><td colspan="2">Descripción:</td></tr>
</table>

• El administrador puede crear, modificar o desactivar empleados.
Observaciones: • Solo el administrador puede acceder a las funcionalidades de administración de empleados.

Tabla 9:

Historia de usuario N°3 – Administración de prealertas

Historia de Usuario	
Numero: RF03	**Usuario:** Administrador, empleado, cliente
Nombre de la historia: Administración de prealertas	
Prioridad del negocio: Alta	**Riesgo en Negocio:** Alta
Puntos estimados: 0.8	
Programador responsable: Daniel Bonilla	
Descripción: • El administrador y los empleados pueden marcar como recibidas o rechazar las prealertas. • El cliente puede crear y eliminar prealertas dirigidas a su casillero virtual.	
Observaciones: • El cliente puede eliminar la prealerta siempre y cuando no haya sido marcada como recibida. • Una prealerta se entiende como la notificación de un paquete a ser entregado en las bodegas de la empresa, la cual cambiará de estado según el paquete del cliente haya sido entregado o rechazado en bodega.	

Tabla 10:

Historia de usuario N°4 – Administración de paquetes

Historia de Usuario	
Numero: RF04	**Usuario:** Administrador, empleado, cliente
Nombre de la historia: Administración de paquetes	

Prioridad del negocio: Alta	**Riesgo en Negocio:** Alta
Puntos estimados: 1	
Programador responsable: Daniel Bonilla	
Descripción: • El administrador y los empleados del sistema pueden modificar el estado de los paquetes del cliente • El cliente puede visualizar el estado de su paquete. • El cliente puede subir la factura del pago del costo del flete.	
Observaciones: • El estado de un paquete puede variar en cuatro situaciones diferentes; pendiente de pago, en tránsito, en aduana, en proceso de entrega.	

Tabla 11:

Historia de usuario Nº5 – Generación reportes localidad

Historia de Usuario	
Numero: RF05	**Usuario:** Administrador
Nombre de la historia: Generar reportes de clientes con mayor cantidad de paquetes	
Prioridad del negocio: Media	**Riesgo en Negocio:** Baja
Puntos estimados: 0.4	
Programador responsable: Daniel Bonilla	
Descripción: • El administrador del sistema puede generar el informe con respecto a los clientes con mayor cantidad de paquetes ordenados desde la plataforma.	
Observaciones: • El informe debe parametrizarse en base a un periodo de tiempo a escoger.	

Tabla 12:

Historia de usuario Nº6 – Generar reportes de clientes con mayor volumen de compras

Historia de Usuario

Numero: RF06	**Usuario:** Administrador
Nombre de la historia: Generar reportes de clientes con mayor volumen de compras	
Prioridad del negocio: Media	**Riesgo en Negocio:** Baja
Puntos estimados: 0.4	
Programador responsable: Daniel Bonilla	
Descripción: • El administrador del sistema puede generar el informe con respecto a los clientes un mayor volumen de compras desde la plataforma.	
Observaciones: • El informe debe parametrizarse en base a un periodo de tiempo a escoger.	

3.1.2 Definición de iteraciones

Una vez establecidas las historias de usuarios, en la Tabla 13 se definen las diferentes iteraciones del proyecto en base a los diferentes parámetros de la tabla, en conjunto con sus puntos estimados, donde un punto es equivalente a una semana de desarrollo por tres horas diarias. La primera iteración se centra en la autenticación y acceso de usuarios, seguida de la administración de empleados, prealertas y paquetes en la segunda iteración. Finalmente, la tercera iteración se enfoca en la generación de reportes basados en el volumen de compras y la cantidad de paquetes.

Tabla 13:

Definición de iteraciones

Nº de historia	**Nombre de la historia**	**Puntos estimados (1 pto = 1 semana)**	**Iteración**
RF01	Autenticación en el sistema	1	1
RF02	Administración de empleados	0.7	2
RF03	Administración de prealertas	0.6	2

RF04	Administración de paquetes	0.7	2
RF05	Generación de reporte de clientes con mayor cantidad de paquetes	0.5	3
RF06	Generación de reportes de clientes con mayor volumen de compras	0.5	3

3.1.3 Plan de entregas

Una vez desarrollado la definición de iteraciones, es necesario definir un plan de entregas en el que consten las fechas de entrega de cada una de las historias de usuario. Para ello se presenta la Tabla 14, presenta las semanas de desarrollo donde se hará entrega de los avances del sistema.

Tabla 14:

Plan de entregas

Entrega	Abril		Mayo				Junio	F. inicio	F. fin
	3s	4s	1s	2s	3s	4s	1s		
Autenticación en el sistema								15/04/2024	21/04/2024
Administración de prealertas								22/04/2024	12/05/2024
Administración de paquetes								13/05/2024	19/05/2024
Administración de empleados								20/05/2024	26/05/2024
Generación de reporte de clientes con mayor cantidad de paquetes								27/05/2024	28/05/2024
Generación de reportes de clientes con mayor volumen de compras								29/05/2024	31/05/2024
Revisión de errores en el Desarrollo de								01/06/2024	03/06/2024

Software									

CAPÍTULO IV: DISEÑO

4.1 Diseño aplicado en la metodología

Como una metodología de diseño ágil, XP fomenta a los usuarios a crear diseños simples y entendibles, que sean fáciles de modificar a futuro, esto en base a un principio fundamental de la metodología, la simplicidad. Este enfoque tiene como objetivo el procurar minimizar el tiempo y esfuerzo en el desarrollo en sus demás iteraciones, sin la necesidad de reestructurar una gran parte el sistema.

Este concepto se refleja en el desarrollo de este trabajo al presentar modelos que, aunque simples, cumplen con la función requerida por el negocio, y da apertura a seguir mejorando y escalando a base de más iteraciones. De igual manera, se crearon interfaces simples e intuitivas con el objetivo de garantizar que los usuarios finales puedan interactuar de manera fácil y efectiva con el sistema.

4.1.1 Diseño arquitectónico del sistema

El presente trabajo utiliza un diseño arquitectónico moderno que no solo satisfaga las necesidades actuales de la empresa Courier, sino que también sea escalable y útil a futuro. Para ello se escogió una arquitectura basada en microservicios, que tal como fue explicado en el marco teórico, consiste en la división de la aplicación en pequeños servicios independientes, intercomunicados entre sí por interfaces, estas interfaces son servicios API Rest, desarrollados en el framework Django Rest Framework. Además, las interfaces de este sistema fueron desarrolladas en el framework de desarrollo de interfaces Vue.js.

Esta arquitectura permite al sistema adaptarse a la infraestructura y escalabilidad que pueda enfrentar el negocio frente a una gran carga de usuarios, manteniendo la disponibilidad ante la caída de cualquiera de sus servicios y manteniendo la simplicidad en cada uno de sus módulos. Además, es posible integrar fácilmente nuevas tecnologías, módulos y funcionalidades dejando abierta la capacidad de adaptarse a los cambios dentro del entorno tecnológico.

4.1.2 Arquitectura basada en microservicios

Ampliando el concepto de la arquitectura basada en microservicios, este tipo de arquitectura se fundamenta en la división de aplicaciones grandes en elementos más pequeños que puedan ser independientes entre sí, y que interconectados, puedan brindar nuevas funcionalidades al usuario sin tener que modificar la programación inicial del sistema (RedHat, 2022).

Gracias a este enfoque, esta arquitectura proporciona cualidades atractivas para los sistemas de información, como la escalabilidad modular, donde únicamente aquellos servicios que experimentan alta demanda sean escalados, resultando en una utilización más eficiente de los recursos. Además de una gran resiliencia, permitiendo que el sistema mantenga sus operaciones incluso cuando uno de sus componentes ha fallado o se haya caído.

Estas ventajas son peculiarmente beneficiosas para un sistema Courier, donde sus diferentes servicios tienen demandas variables y es indispensable el minimizar el impacto que puede tener la caída de uno de estos servicios en la operación general. Esto con la principal finalidad de mantener el mayor tiempo posible el servicio activo para los usuarios y así cumplir con las expectativas de los clientes.

Por tal razón, para el desarrollo de este proyecto, se diseñaron dos microservicios: uno orientado a la autenticación del sistema y otro para la gestión de paquetes. Cada microservicio fue desarrollado con Django REST framework y está enlazado directamente a su correspondiente base de datos MySQL.

La intercomunicación entre los servicios de autenticación y gestión de paquetes se realiza a través de las API REST, estas interfaces de programación permiten la comunicación con el sistema para obtener y procesar datos (RedHat, 2023). Django REST framework maneja este intercambio de datos mediante solicitudes HTTP, donde cada solicitud contendrá una acción a realizar en el servicio y devolverá una respuesta al cliente en formato JSON.

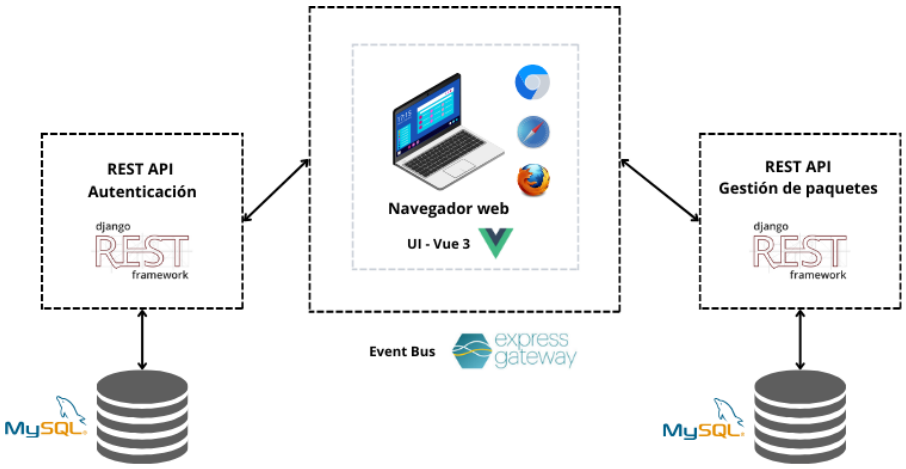
Cada una de estas peticiones no se conecta directamente con la interfaz del cliente, que ha sido desarrollada en el framework de desarrollo web Vue.js. En su lugar, las peticiones son canalizadas

hacia el bus de datos. Este componente, desarrollado en Express gateway, funciona como intermediario, recibiendo, administrando y procesando las solicitudes entrantes.

Así entonces, la Figura 1 ilustra la arquitectura de microservicios que se utiliza en el desarrollo del proyecto:

Ilustración 1:

Arquitectura de microservicios utilizada en el desarrollo del proyecto



4.2 Tarjetas CRC

Las tarjetas CRC son entregables de la metodología XP, estas representan objetos en el sistema o modelo a implementar. Su estructura es simple y se encuentra enlazada con las siglas de su nombre, es decir; clases, responsabilidades y colaboración. Su principal finalidad es representar el diseño del proyecto, de manera que se encuentre recopilada cada una de las clases en el sistema, en conjunto con sus responsabilidades, colaboradores, atributos y operaciones, convirtiéndose en una aproximación al diagrama de clases.

Tabla 15:

Tarjeta CRC - Cliente

CLIENTE	
RESPONSABILIDADES	**COLABORADORES**
Atributos: • Id • Número de Identificación	• Pedido • Paquete

(CI, pasaporte) • Nombres • Apellidos • Email • Ciudad • Dirección • Teléfono • Contraseña • Código postal • Número de casillero • Activo • Fecha de creación **Operaciones:** • Autenticarse • Crear cliente • Actualizar Cliente • Consultar Cliente	

Tabla 16:

Tarjeta CRC - Pedido

PEDIDO	
RESPONSABILIDADES	**COLABORADORES**
Atributos: • Id • Código • Tienda • Carrier • Cliente • Empleado • Receptor • Descripción • Observación • Código de seguimiento • Valor de la compra • Factura • Fecha de creación • Estado del pedido **Operaciones:** • Crear pedido • Actualizar pedido • Consultar pedido • Cambiar estado • Recibir pedido • Rechizar pedido	• Cliente • Empleado • Tienda • Carrier • Paquete

Tabla 17:

Tarjeta CRC - Tienda

TIENDA	
RESPONSABILIDADES	**COLABORADORES**
Atributos: • Id • Nombre • Descripción • Email • Teléfono • Activo **Operaciones:** • Crear tienda • Actualizar tienda • Deshabilitar tienda	• Pedido

Tabla 18:

Tarjeta CRC - Carrier

CARRIER	
RESPONSABILIDADES	**COLABORADORES**
Atributos: • Id • Nombre • Descripción • Email • Teléfono • Activo **Operaciones:** • Crear carrier • Actualizar carrier • Deshabilitar carrier	• Pedido

Tabla 19:

Tarjeta CRC - Empleado

Empleado	
RESPONSABILIDADES	**COLABORADORES**
Atributos: • Id • Número de Identificación (CI, pasaporte) • Nombres	• Paquete • Pedido

• Apellidos • Email • Ciudad • Dirección • Teléfono • Contraseña • Es_Administrador • Cargo • Activo • Fecha de creación **Operaciones:** • Autenticarse • Crear empleado • Actualizar empleado • Deshabilitar empleado • Consultar empleado • Consultar pedido • Consultar informes	

Tabla 20:

Tarjeta CRC - Paquete

PAQUETE	
RESPONSABILIDADES	**COLABORADORES**
Atributos: • Id • Empleado • Cliente • Código • Dirección • Ciudad • Número Postal • Categoría • Peso • Estado del paquete • Costo del flete • Comprobante de pago • Prueba de pago • Fecha en transito • Fecha en aduana • Fecha en entrega • Fecha recibida **Operaciones:** • Crear paquete • Consultar paquete • Cambiar estado del paquete	• Pedido • Cliente • Empleado

Tabla 21:

Tarjeta CRC - Usuario

Usuario	
RESPONSABILIDADES	**COLABORADORES**
Atributos: • Id • Número de Identificación (CI, pasaporte) • Nombres • Apellidos • Email • Ciudad • Dirección • Teléfono • Contraseña • Código postal • Es_Administrador • Activo • Fecha de creación **Operaciones:** • Autenticarse • Crear usuario • Actualizar usuario • Consultar usuario	• Cliente • Empleado

4.3 Diagrama conceptual

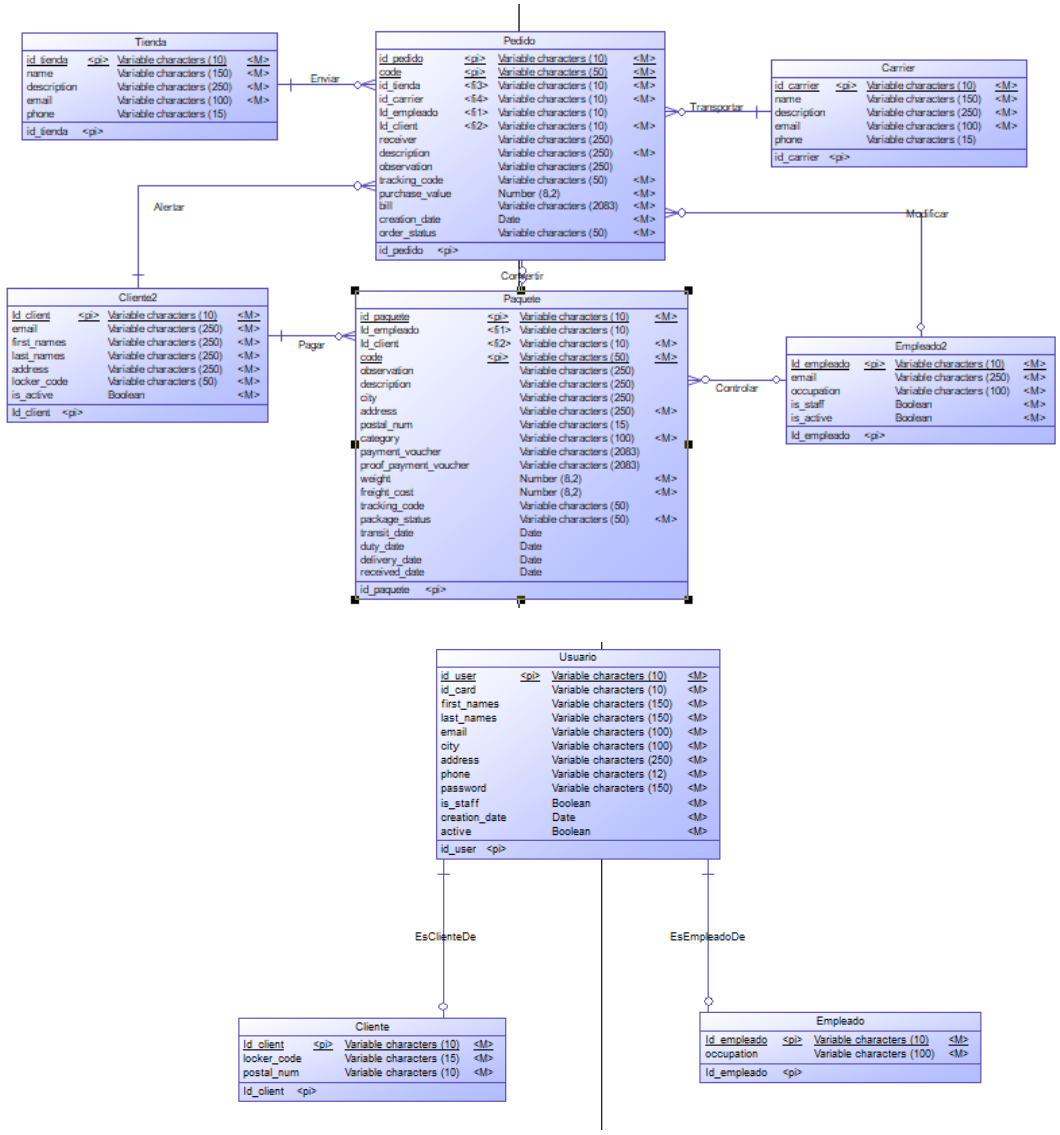
Un diagrama de conceptual es una técnica utilizada para representar las diferentes entidades que conforman el modelo de un sistema, describiendo entre ellas las diferentes relaciones que ocupan para representar la estructura interna de un proyecto.

A continuación, se presenta el diagrama de clases del sistema, recopilando los datos

en cada una de las tarjetas CRC, separando respectivamente cada tabla con su microservicio.

Ilustración 2:

Diagramas conceptuales del sistema

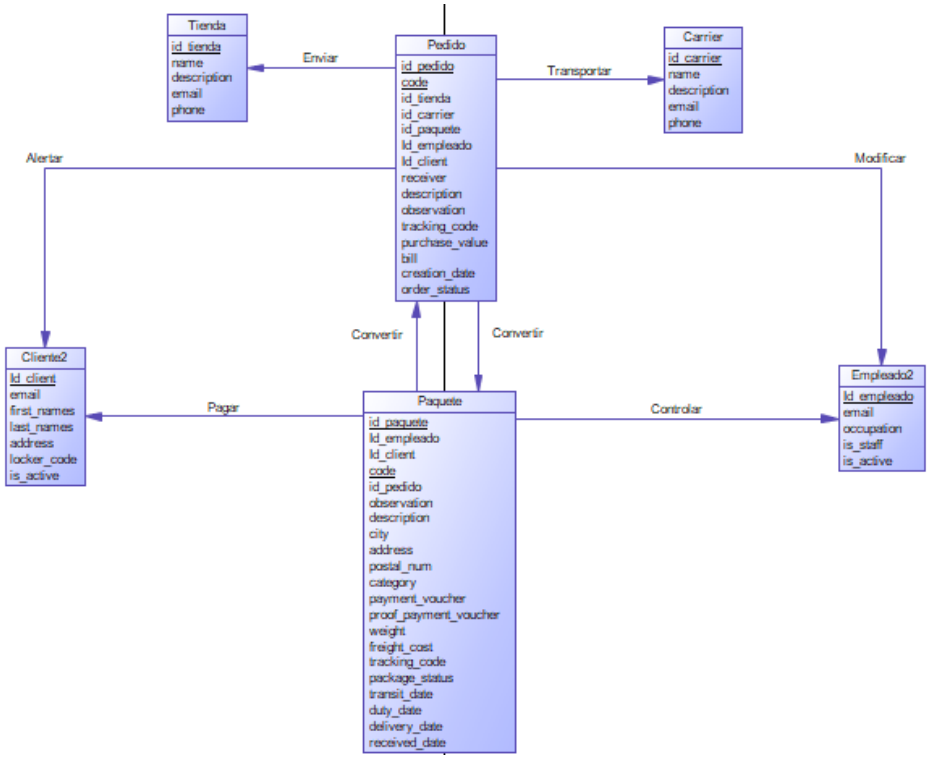


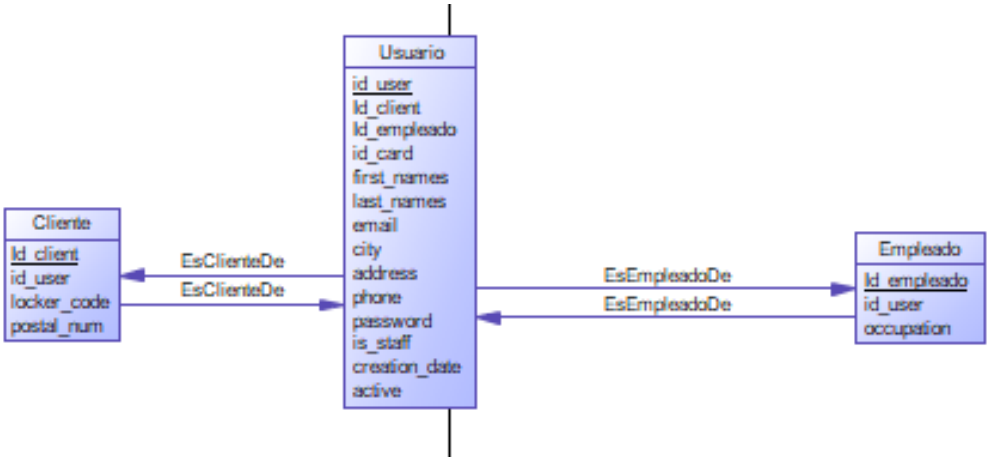
4.4 Diagrama Físico de base de datos

Una vez desarrollado el diagrama de clases, se presenta el diagrama físico de la solución, separando respectivamente cada grafico con su microservicio. Este detalla la estructura de las diferentes tablas, componentes, relaciones y claves primarias de la base de datos utilizadas para el desarrollo de este trabajo de titulación.

Ilustración 3:

Diagramas físicos de las bases de datos





4.5 Diseño de las interfaces de usuario

Este trabajo busca crear interfaces simples e intuitivas para el usuario, utilizando colores claros y una tipografía redondeada para mantener la legibilidad del sistema en todo momento. Cabe resaltar que este será un prototipo del sistema, por lo que las imágenes utilizadas son referenciales y probablemente serán modificas en el producto final.

4.5.1 Interfaz de registro al sistema

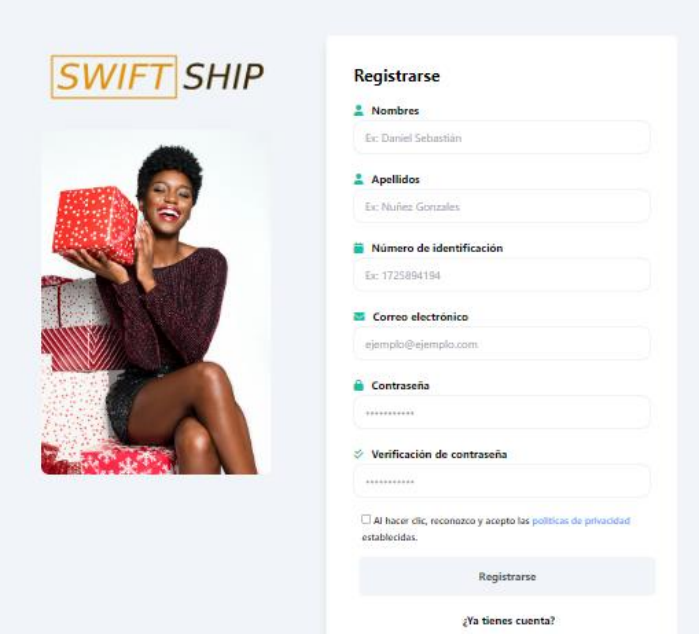
El registro en el sistema será exclusivamente para clientes. Se solicitarán diversos datos para registrar toda la información necesaria del cliente.

4.5.1.1 Clientes

Se solicita información del usuario como su nombre completo, dirección, número celular, email, código postal y una contraseña para ingresar al sistema. La ilustración 4 muestra el registro de clientes.

Ilustración 4:

Interfaz de registro de clientes



4.5.2 Interfaz de acceso al sistema

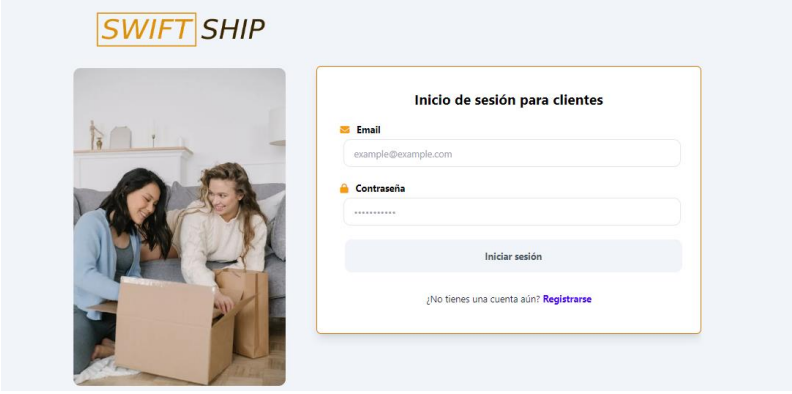
El acceso al sistema se realizará como cliente o como usuario de este, los clientes podrán crear las prealertas y visualizar sus paquetes, mientras que los clientes podrán modificar el estado de las prealertas y los paquetes según vaya transcurriendo el giro del negocio.

4.5.2.1 Clientes

Se accede al sistema mediante el correo electrónico y contraseña proporcionados en el registro del usuario. La ilustración 5 muestra la interfaz de acceso para clientes.

Ilustración 5:

Interfaz de acceso al sistema para clientes

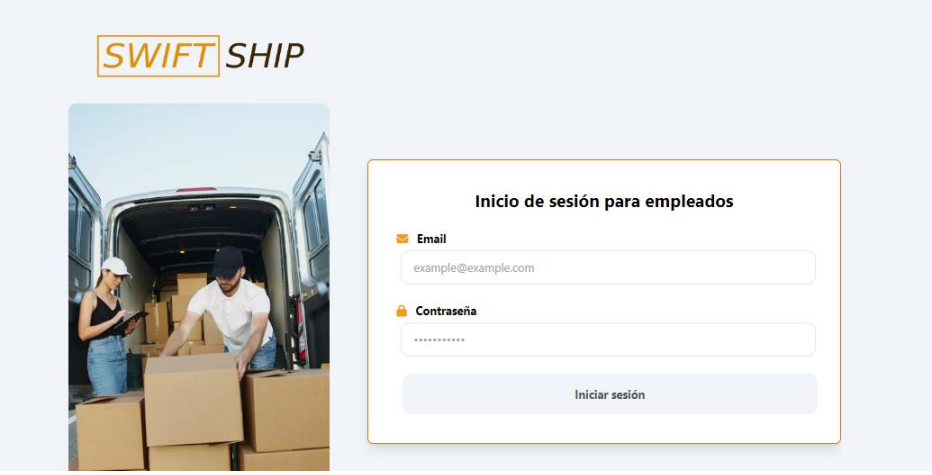


4.5.2.2 Empleados

Se puede acceder al sistema mediante el usuario y contraseña proporcionados en el registro del usuario. La ilustración 6 muestra la interfaz de acceso para empleados.

Ilustración 6:

Interfaz de acceso al sistema para empleados

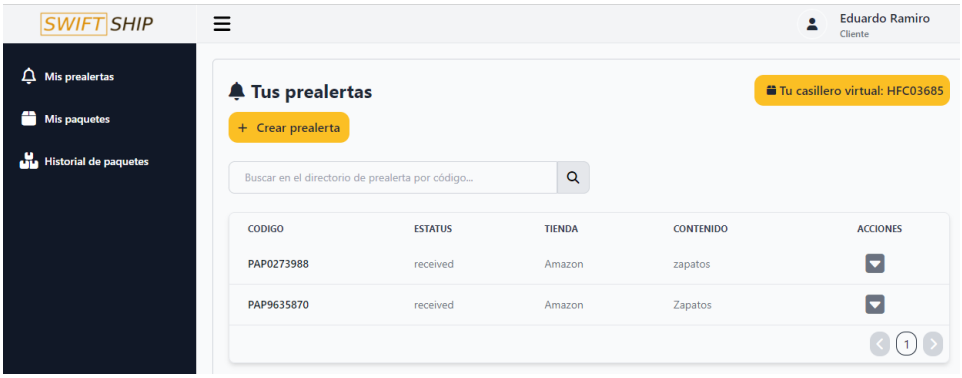


4.5.3 Interfaz ingreso prealertas

Esta interfaz permite ingresar los datos de una prealerta, esto con el objetivo que sea recibida en la bodega de la empresa y administrada como tal. La ilustración 7 muestra la interfaz de ingreso de prealertas.

Ilustración 7:

Interfaz de ingreso de prealertas

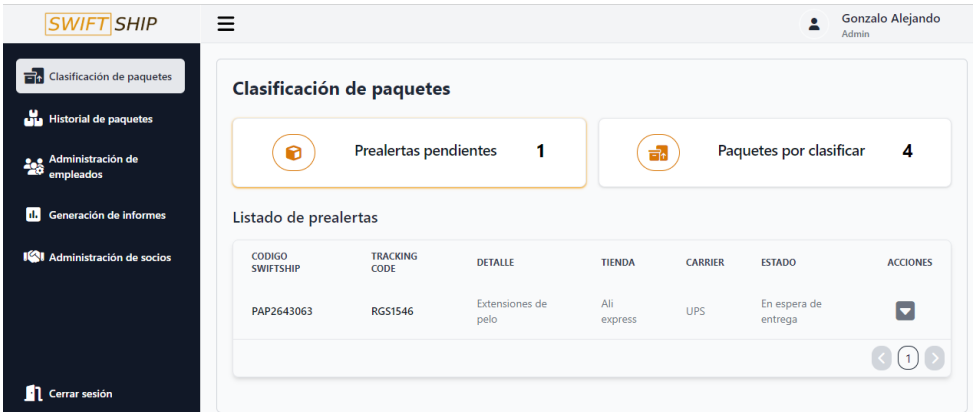


4.5.4 Interfaz de clasificación de prealertas y paquetes

Una vez accedido al sistema, en la sección principal de administración del cliente observan los datos de todos los paquetes y prealertas pendientes por clasificar, el empleado en cuestión cuenta con los permisos para cambiar el estado de cada uno de estos paquetes y así enviar la información al cliente. La ilustración 8 muestra la interfaz de clasificación de prealertas y paquetes.

Ilustración 8:

Interfaz de clasificación de prealertas y paquetes

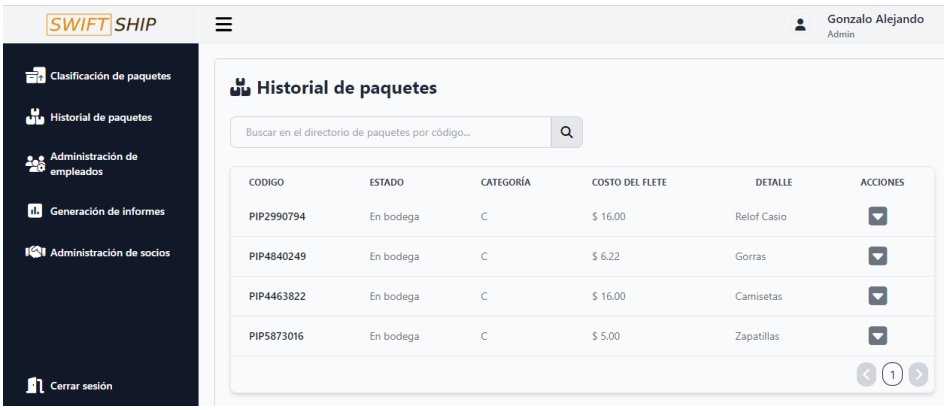


4.5.5 Interfaz de historial de paquetes

Esta interfaz cuenta con toda la información en detalle de los paquetes ingresados en el sistema, además, incluye un buscador por código para encontrar paquetes en específico. La ilustración 9 muestra la interfaz de historial de paquetes.

Ilustración 9:

Interfaz de historial de paquetes



4.5.6 Interfaz de visualización de paquetes

En esta interfaz el usuario visualiza el estado de todos sus paquetes, además de poder ser clasificados mediante el estado como tal dando clic en cada uno de los distintos estados. Cada uno de estos estados cuenta con un contador para brindar información adicional del estado de sus paquetes. La ilustración 10 muestra la interfaz de visualización de paquetes.

Ilustración 10:

Interfaz de visualización de paquetes



4.5.7 Interfaz de administración de empleados

Para los usuarios de sistema es necesaria una credencial de acceso, esta interfaz contará con elementos para la creación y edición de las cuentas de estos usuarios. La ilustración 11 muestra la interfaz de ingreso de administración de empleados.

Ilustración 11:

Interfaz de administración de empleados

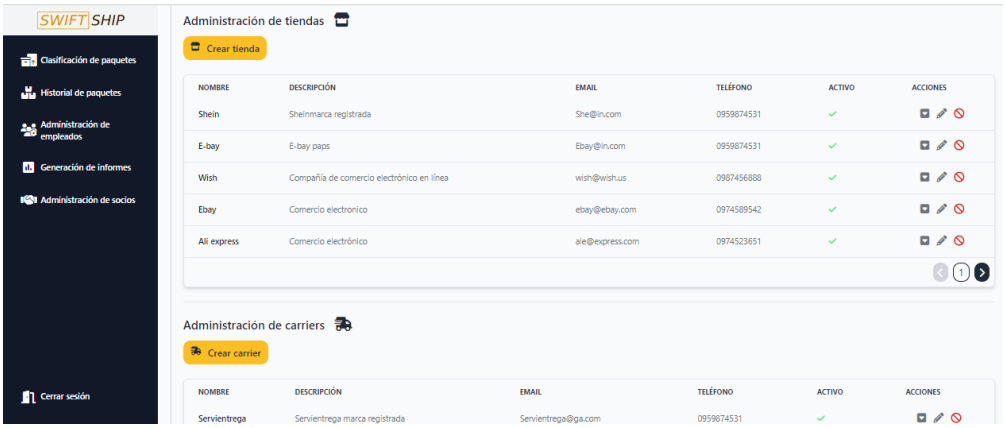


4.5.8 Interfaz de administración de socios

Un servicio Courier está en constante actualización con respecto a las tiendas y Carrier con los que trabaja, por lo cual, esta interfaz permite crear, modificar y deshabilitar estos aspectos. La ilustración 12 muestra la interfaz de administración de socios.

Ilustración 12:

Interfaz de administración de socios

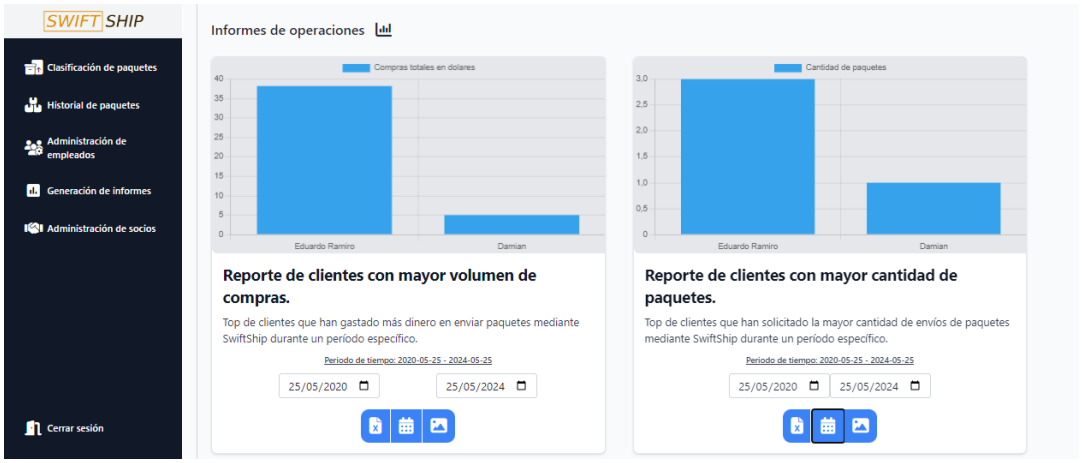


4.5.9 Interfaz de reportes

Si es que el usuario cuenta con el rol de administrador, entonces puede visualizar sección de reportes en la barra lateral izquierda. Dentro de la interfaz se presentan las estadísticas de prealertas recibidas en el día y paquetes entregados en el día. Además de contar con las secciones para generar los diferentes reportes recopilados como requerimientos en la sección de historias de usuario. Esta interfaz permite escoger el rango de fechas con las cuales se generará el reporte de volumen de compras de los clientes y el reporte de clientes con mayor cantidad de paquetes respectivamente, en conjunto con las opciones de descargar un archivo Excel de estos datos, elegir el periodo desde el inicio de los tiempos, o descargar la gráfica generada por el sistema. La ilustración 13 muestra la interfaz de reportes.

Ilustración 13:

Interfaz de reportes


SWIFT SHIP
Clasificación de paquetes
Historial de paquetes
Administración de empleados
Generación de informes
Administración de socios
Cerrar sesión
Informes de operaciones
Compras totales en dólares
40
35
30
25
20
15
10
5
0
Eduardo Ramiro
Damian
Reporte de clientes con mayor volumen de compras.
Top de clientes que han gastado más dinero en enviar paquetes mediante SwiftShip durante un período específico.
Período de tiempo: 2020-05-25 - 2024-05-25
25/05/2020
25/05/2024
Cantidad de paquetes
3,0
2,5
2,0
1,5
1,0
0,5
0
Eduardo Ramiro
Damian
Reporte de clientes con mayor cantidad de paquetes.
Top de clientes que han solicitado la mayor cantidad de envíos de paquetes mediante SwiftShip durante un período específico.
Período de tiempo: 2020-05-25 - 2024-05-25
25/05/2020
25/05/2024

CAPÍTULO V: DESARROLLO

5.1 Fase de iteraciones

En esta fase se lleva a cabo el desarrollo de cada una de las iteraciones del desarrollo del sistema web. Estas iteraciones fueron definidas y ponderadas en las etapas anteriores de este trabajo, y se han ordenado en base a la prioridad que representa en el giro de negocio del sistema Courier.

La metodología XP recalca la importancia de desarrollar iteraciones cortas y frecuentes, aplicadas en este trabajo, las iteraciones se centraron en cada una de las historias de usuario, detallando así cada una de las tareas y desafíos involucrados en su desarrollo.

Al finalizar cada iteración, se ejecutaban y revisaban las diferentes pruebas de aceptación y de carga, las cuales validaban con el usuario el rendimiento y ejecución esperados del sistema. Fue gracias a esta retroalimentación cíclica que se incorporaron diversos cambios en la forma en que se procesa y presenta la información, siguiendo un enfoque iterativo que evolucionaba con cada una de las necesidades y expectativas del negocio.

5.1.1 Primera iteración

5.1.1.1 Historias

RF01: Que la aplicación permita el registro y el ingreso de usuarios al sistema.

5.1.1.2 Tareas

Debido a que la solución propuesta es una arquitectura basada en microservicios, la primera tarea fue establecer el ambiente de trabajo, creando todos los proyectos en las distintas aplicaciones, instalando librerías, paquetes necesarios y finalmente levantando todos los servicios a utilizar.

Una vez finalizado, el primer servicio y componente creado es la página de registro y autenticación de usuarios. Este componente se encargará de registrar toda la información necesaria del cliente o empleado y de redirigirlo en base a las credenciales expuestas.

Para mantener la seguridad del usuario, la autenticación persistente se manejará mediante JSON Web Token, o mejor conocido como JWT. El token tendrá una permanencia de una hora, tras la cual expirará y el usuario deberá iniciar sesión nuevamente.

Ya que existen varios roles en la aplicación, la tabla 22 recopila las características o permisos con la que cuenta cada clase.

Tabla 22

Roles de Usuario

Rol	Tipo	Características
Administrador	Accesos y permisos a paquetes, prealertas, administración de empleado y generación de informes	Capacidad de aceptar o rechazar prealertas. Capacidad de administrar paquetes. Capacidad de administrar empleados. Capacidad de administrar socios. Capacidad de visualizar y generar informes.
Empleado	Accesos y permisos a paquetes y prealertas	Capacidad de aceptar o rechazar prealertas. Capacidad de administrar paquetes.
Cliente	Acceso únicamente a sus paquetes y prealertas	Capacidad de crear y eliminar sus prealertas Capacidad de visualizar sus paquetes.

5.1.2 Segunda iteración

5.1.2.1 Historias

RF02: Que la aplicación permita la administración de empleados.

RF03: Que la aplicación permita la administración de prealertas.

RF04: Que la aplicación permita la administración de paquetes.

5.1.2.2 Tareas

Para esta iteración se desarrolló el componente de administración de empleados, únicamente accesible mediante usuarios con el rol de is_staff. Este componente permite al usuario crear y modificar toda la información de los empleados, además de controlar si un empleado debe tener permisos de administrador para gestionar otros elementos del sistema.

Por otro lado, para la administración de prealertas y paquetes, se crearon dos componentes para cada sección; una para el lado del cliente, y otro para el lado de los empleados. Al igual que lo dispuesto en las historias de usuario, el cliente puede crear sus prealertas, visualizar el estado de estas y subir los pagos correspondientes si es necesario. Mientras que, los empleados, gestionan el estado de las prealertas y paquetes, aceptándolos o rechazándolos según sea el caso.

5.1.3 Tercera iteración

5.1.3.1 Historias

RF05: Que el sistema permita generar reportes de clientes con la mayor cantidad de paquetes

RF06: Que el sistema permita generar reportes de clientes con el mayor volumen de compras.

5.1.3.2 Tareas

Para esta tarea se creó un componente únicamente accesible para los administradores del sistema, es decir aquellos que cuentan con el rol de is_staff. Este componente utiliza la librería de chart.js para crear gráficos en base a data tipo JSON proporcionada por el servicio de gestión de paquetes. Para crear el grafico, el usuario tiene que seleccionar un periodo de tiempo entre dos fechas, así se habilitarán las opciones de descargar la información de este gráfico en Excel o la propia ilustración en formato png.

5.2 Fase de pruebas

En esta fase se realizan las diferentes pruebas de aceptación y de carga de cada módulo desarrollado.

5.2.1 Primera iteración

5.2.1.1 Pruebas de aceptación

Tabla 23

Caso de prueba RF01_P01

Prueba de aceptación	
Código: RF01_P01	Historia de usuario: RF01
Nombre: Registro de clientes	
Descripción: Disponer de un formulario donde los clientes se puedan registrar en el sistema.	
Condiciones: Ninguna	
Pasos: 1. Desde la pantalla de inicio hacer clic en Registrarse 2. Rellenar el formulario con todos los datos del cliente. 3. Presionar botón de Registrarse	
Resultados esperados: Se registra el cliente nuevo y se redirige a la pantalla principal.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 24

Caso de prueba RF01_P02

Prueba de aceptación	
Código: RF01_P02	Historia de usuario: RF01
Nombre: Autenticación de clientes	
Descripción: Disponer de un formulario donde los clientes puedan ingresar en el sistema con sus credenciales.	
Condiciones: Tener una cuenta creada con su correo y contraseña.	
Pasos: 1. Desde la pantalla de inicio rellenar el formulario. 2. Presionar botón de Iniciar Sesión.	
Resultados esperados: El cliente inicia sesión y se redirige al panel de administración de paquetes.	

Evaluación de la prueba: Prueba satisfactoria

Tabla 25

Caso de prueba RF01_P03

Prueba de aceptación	
Código: RF01_P03	Historia de usuario: RF01
Nombre: Autenticación de empleados	
Descripción: Disponer de un formulario donde los empleados puedan ingresar en el sistema con sus credenciales.	
Condiciones: Tener una cuenta creada con su correo y contraseña.	
Pasos: 1. Desde la pantalla de inicio rellenar el formulario. 2. Presionar botón de Iniciar Sesión.	
Resultados esperados: El empleado inicia sesión y se redirige al panel de clasificación de paquetes.	
Evaluación de la prueba: Prueba satisfactoria	

5.2.1.2 Prueba de carga servicio de autenticación

5.2.1.2.1 Escenario de prueba:

En esta prueba se evalúa el módulo de autenticación, el objetivo es evaluar el rendimiento del sistema en horas pico, simulando 100 usuarios concurrentes que intentan autenticarse de manera simultánea durante un período de 10 segundos.

5.2.1.2.1 Resultados de la Prueba:

Ilustración 14

Resultados prueba de carga autenticación

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento
Login Auth	100	881	455	1404	143,73	0,00%	9,5/sec
Total	100	881	455	1404	143,73	0,00%	9,5/sec

El sistema paso satisfactoriamente esta prueba, cumpliendo en su totalidad las peticiones de

inicio de sesión, mostrando un rendimiento regular bajo la prueba de carga, con un 100% de peticiones exitosas y un tiempo de respuesta promedio de 881 ms, tiempo un poco superior a lo aceptable para el tiempo de carga de una aplicación web, no obstante, cumple con el objetivo del sistema.

5.2.2 Segunda iteración

5.2.2.1 Pruebas de aceptación

Tabla 26

Caso de prueba RF03_P01

Prueba de aceptación	
Código: RF03_P01	Historia de usuario: RF03
Nombre: Ingreso de prealertas	
Descripción: Disponer de una pantalla donde se puedan subir las prealertas.	
Condiciones: Tener una cuenta creada con su correo y contraseña.	
Pasos: 1. Desde la pantalla de inicio dar clic al botón de crear prealerta. 2. Rellenar el formulario y subir la factura con todos los datos del paquete. 3. Presionar botón de crear prealerta.	
Resultados esperados: La prealerta se crea, la ventana se recarga y se visualiza la prealerta creada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 27

Caso de prueba RF03_P02

Prueba de aceptación	
Código: RF03_P02	Historia de usuario: RF03
Nombre: Visualización de prealertas	
Descripción: Disponer de una pantalla donde se puedan visualizar en detalle una prealerta.	
Condiciones:	

Tener una cuenta creada con su correo y contraseña. Contar con al menos una prealerta creada.
Pasos: 1. Desde la pantalla de inicio dar clic al icono en forma de flecha descendente al lado de la prealerta. 2. Visualizar la información de la prealerta.
Resultados esperados: En pantalla aparece una ventana modal con toda la información en detalle de la prealerta.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 28

Caso de prueba RF03_P03

Prueba de aceptación	
Código: RF03_P03	Historia de usuario: RF03
Nombre: Búsqueda de prealertas	
Descripción: Disponer de una pantalla donde se permita buscar las prealertas por código.	
Condiciones: Tener una cuenta creada con su correo y contraseña. Tener una prealerta creada con el código a buscar.	
Pasos: 1. Desde la pantalla de inicio dar clic al campo de búsqueda. 2. Colocar el código de la prealerta requerida. 3. Presionar botón con icono de búsqueda.	
Resultados esperados: Se carga las prealertas que coinciden con el código de búsqueda.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 29

Caso de prueba RF03_P04

Prueba de aceptación	
Código: RF03_P04	Historia de usuario: RF03
Nombre: Eliminación de prealertas	

Descripción: Disponer de una pantalla donde se permita eliminar las prealertas por código.
Condiciones: Tener al menos una prealerta creada y que no haya sido marca como obtenida hasta el momento. Tener una cuenta creada con su correo y contraseña.
Pasos: 1. Desde la pantalla de inicio dar clic al icono en forma de balde de basura. 2. Confirmar la eliminación de la prealerta.
Resultados esperados: Se elimina la prealerta del listado de prealertas.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 30

Caso de prueba RF03_P05

Prueba de aceptación	
Código: RF03_P05	Historia de usuario: RF03
Nombre: Recepción de prealerta	
Descripción: Disponer de una pantalla donde se permita marcar como recibidas las prealertas generadas por el usuario.	
Condiciones: Tener al menos una prealerta creada y pendiente de clasificar. Tener una cuenta de empleado creada con su correo y contraseña.	
Pasos: 1. Desde la pantalla de inicio dar clic al icono en forma de flecha descendente al lado de la prealerta. 2. Presionar el botón “recibir” en la ventana modal. 3. Completar los datos del formulario para la creación del paquete. 4. Confirmar la recepción del paquete	
Resultados esperados: Se marca como recibida la prealerta del listado de prealertas, se crea un nuevo paquete en la sección de paquetes.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 31

Caso de prueba RF03_P06

Prueba de aceptación	
Código: RF03_P06	Historia de usuario: RF03
Nombre: Rechazo de prealerta	
Descripción: Disponer de una pantalla donde se permita marcar como rechazadas las prealertas generadas por el usuario.	
Condiciones: Tener al menos una prealerta creada y pendiente de clasificar. Tener una cuenta de empleado creada con su correo y contraseña.	
Pasos: 1. Desde la pantalla de inicio dar clic al icono en forma de flecha descendente al lado de la prealerta. 2. Presionar el botón “rechazar” en la ventana modal. 3. Completar los datos del formulario para el rechazo del paquete. 4. Confirmar el rechazo del paquete	
Resultados esperados: Se marca como rechazada la prealerta del listado de prealertas.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 32

Caso de prueba RF04_P01

Prueba de aceptación	
Código: RF04_P01	Historia de usuario: RF04
Nombre: Visualización en detalle del paquete	
Descripción: Disponer de una pantalla donde se permita al cliente ver en detalle toda la información y el estado del paquete	
Condiciones: Tener al menos un paquete creado. Tener una cuenta de cliente creada con su correo y contraseña.	
Pasos: 1. Desde la pantalla de inicio dar clic a la pestaña de ‘Mis	

paquetes’ 2. Presionar al icono en forma de fecha descendente al lado del paquete.
Resultados esperados: Visualizar en detalle toda la información y estado del paquete.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 33

Caso de prueba RF04_P02

Prueba de aceptación	
Código: RF04_P02	Historia de usuario: RF04
Nombre: Modificación del estado del paquete	
Descripción: Disponer de una pantalla donde se permita marcar los diferentes estados por los que transcurre un paquete.	
Condiciones: Tener al menos un paquete creado y pendiente de clasificar. Tener una cuenta de empleado creada con su correo y contraseña.	
Pasos: 1. Desde la pantalla de inicio dar clic a la sección de paquetes por clasificar. 2. Presionar al icono en forma de flecha descendente al lado del paquete. 3. Presionar el botón “actualizar estado” en la ventana modal. 4. Seleccionar el estado actual del paquete y agregar si es que se tienen alguna observación. 5. Presionar el botón de completar	
Resultados esperados: Se cambia el estado del paquete dentro del listado y se actualiza su fecha de estado en la interfaz de usuario.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 34

Caso de prueba RF04_P03

Prueba de aceptación	
Código: RF04_P03	Historia de usuario: RF04

Nombre: Subida del comprobante de pago al paquete
Descripción: Disponer de una pantalla donde se permita al cliente subir el comprobante de pago de su respectiva factura.
Condiciones: Tener al menos un paquete creado de clasificar. Tener una cuenta de cliente creada con su correo y contraseña.
Pasos: 3. Desde la pantalla de inicio dar clic a la pestaña de 'Mis paquetes' 4. Presionar al icono en forma de factura al lado del paquete. 5. Presionar el texto resaltado 'aquí' para ver la factura del cliente. 6. Presionar el botón de seleccionar archivo en la parte inferior para subir el comprobante de pago. 7. Presionar el botón de subir comprobante
Resultados esperados: Se oculta el ícono de factura pendiente en el listado de paquetes.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 35

Caso de prueba RF04_P04

Prueba de aceptación	
Código: RF04_P04	Historia de usuario: RF04
Nombre: Búsqueda de paquetes	
Descripción: Disponer de una pantalla donde se permita buscar los paquetes por código.	
Condiciones: Tener una cuenta de cliente creada con su correo y contraseña. Tener al menos un paquete creada con el código a buscar.	
Pasos: 1. Desde la pantalla de inicio dar clic a la pestaña de historial de paquetes. 2. Colocar el código de la prealerta requerida en el campo de búsqueda.	

3. Presionar botón con icono de búsqueda.
Resultados esperados: Se carga los paquetes que coinciden con el código de búsqueda.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 36

Caso de prueba RF02_P01

Prueba de aceptación	
Código: RF02_P01	Historia de usuario: RF02
Nombre: Creación de cuenta de empleado	
Descripción: Disponer de una pantalla con un formulario que permita crear empleados.	
Condiciones: Tener una cuenta de administrador creada con su correo y contraseña.	
Pasos: 1. Desde la pantalla de inicio dar clic a la pestaña de administración de empleados. 2. Pulsar el botón de crear empleado. 3. Rellenar los datos del formulario. 4. Pulsar el botón de crear empleado.	
Resultados esperados: Se crea una nueva cuenta de empleado, visible desde la tabla del componente y accesible por el empleado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 37

Caso de prueba RF02_P02

Prueba de aceptación	
Código: RF02_P02	Historia de usuario: RF02
Nombre: Creación de cuenta de administrador	
Descripción: Disponer de una pantalla con un formulario que permita crear administradores.	
Condiciones: Tener una cuenta de administrador creada con su correo y contraseña.	
Pasos:	

1. Desde la pantalla de inicio dar clic a la pestaña de administración de empleados. 2. Pulsar el botón de crear empleado. 3. Rellenar los datos del formulario. 4. En la sección de cargo seleccionar el cargo de administrador. 5. Pulsar el botón de crear empleado.
Resultados esperados: Se crea una nueva cuenta de administrador, visible desde la tabla del componente y accesible por el administrador.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 38

Caso de prueba RF02_P03

Prueba de aceptación	
Código: RF02_P03	Historia de usuario: RF02
Nombre: Visualización en detalle de cuenta de empleado	
Descripción: Disponer de una pantalla que permita visualizar en detalle la información del empleado.	
Condiciones: Tener una cuenta de administrador creada con su correo y contraseña. Tener al menos una cuenta de empleado o administrador creada.	
Pasos: 1. Desde la pantalla de inicio dar clic a la pestaña de administración de empleados. 2. Pulsar el icono de una flecha descendente junto al empleado.	
Resultados esperados: Se visualiza en detalle la cuenta del empleado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 39

Caso de prueba RF02_P04

Prueba de aceptación	
Código: RF02_P04	Historia de usuario: RF02
Nombre: Modificación de cuenta de empleado	

Descripción: Disponer de una pantalla con un formulario que permita modificar empleados.
Condiciones: Tener una cuenta de administrador creada con su correo y contraseña. Tener al menos una cuenta de empleado o administrador creada.
Pasos: 1. Desde la pantalla de inicio dar clic a la pestaña de administración de empleados. 2. Pulsar el icono de un lápiz junto al empleado. 3. Cambiar los datos en el formulario 4. Pulsar el botón con incono de diskette (guardar)
Resultados esperados: Se modifican los datos de la cuenta de empleado, visible desde la tabla del componente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 40

Caso de prueba RF02_P05

Prueba de aceptación	
Código: RF02_P05	Historia de usuario: RF02
Nombre: Deshabilitar cuenta de empleado	
Descripción: Disponer de una interfaz que permita deshabilitar cuentas de empleado	
Condiciones: Tener una cuenta de administrador creada con su correo y contraseña. Tener al menos una cuenta de empleado o administrador creada.	
Pasos: 1. Desde la pantalla de inicio dar clic a la pestaña de administración de empleados. 2. Pulsar el icono de prohibición junto al empleado. 3. Confirmar deshabilitación del empleado.	
Resultados esperados: La cuenta del empleado queda suspendida y no es accesible.	
Evaluación de la prueba: Prueba satisfactoria.	

5.2.2.2 Prueba de carga servicio de gestión de paquetes

5.2.2.2.1 Escenario de prueba:

En esta prueba se evalúa el módulo de gestión de paquetes, el objetivo es evaluar el rendimiento del sistema en horas pico, simulando 100 usuarios concurrentes que intentan crear sus paquetes de manera simultánea durante un período de 10 segundos.

5.2.2.2.1 Resultados de la Prueba:

Ilustración 15

Resultados prueba de creación de paquetes.

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento
Create Packages	100	34	27	63	6,59	0,00%	10,1/sec
Total	100	34	27	63	6,59	0,00%	10,1/sec

El sistema paso satisfactoriamente esta prueba, cumpliendo en su totalidad las peticiones de creación de paquetes, mostrando un rendimiento bueno bajo la prueba de carga, con un 100% de peticiones exitosas y un tiempo de respuesta promedio de 34ms, tiempo sumamente bueno para la generación de peticiones.

5.2.3 Tercera iteración

5.2.3.1 Pruebas de aceptación

Tabla 41

Caso de prueba RF05_P01

Prueba de aceptación	
Código: RF05_P01	Historia de usuario: RF05
Nombre: Descargar informe de reporte de clientes con mayor volumen de compras	
Descripción: Disponer de una interfaz que permita seleccionar un periodo de tiempo para generar un reporte de los clientes con mayor volumen de compras.	

Condiciones: Tener una cuenta de administrador creada con su correo y contraseña. Tener al menos un usuario que haya desarrollado un pedido en el sistema.
Pasos: 1. Desde la pantalla de inicio dar clic a la pestaña de generación de informes. 2. Seleccionar el periodo de tiempo en el componente de reporte por volumen de compras. 3. Pulsar el botón de descarga archivo Excel.
Resultados esperados: Se genera la gráfica y se descargan los datos de los clientes con mayor volumen de compras.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 42

Caso de prueba RF06_P01

Prueba de aceptación	
Código: RF06_P01	Historia de usuario: RF06
Nombre: Descargar informe de reporte de clientes con mayor cantidad de paquetes ordenados.	
Descripción: Disponer de una interfaz que permita seleccionar un periodo de tiempo para generar un reporte de los clientes con mayor cantidad de paquetes ordenados.	
Condiciones: Tener una cuenta de administrador creada con su correo y contraseña. Tener al menos un usuario que haya desarrollado un pedido en el sistema.	
Pasos: 1. Desde la pantalla de inicio dar clic a la pestaña de generación de informes. 2. Seleccionar el periodo de tiempo en el componente de reporte por volumen de compras. 3. Pulsar el botón de descarga archivo Excel.	
Resultados esperados: Se genera la gráfica y se descargan los datos de los clientes con mayor cantidad de paquetes ordenados.	
Evaluación de la prueba: Prueba satisfactoria.	

5.2.3.2 Prueba de carga servicio de consulta de reportes

5.2.3.2.1 Escenario de prueba:

En esta prueba se evalúa el módulo de consulta de paquetes, el objetivo es evaluar el rendimiento del sistema en horas pico, simulando 100 empleados concurrentes que intentan consultar los informes de paquetes de manera simultánea durante un período de 10 segundos.

5.2.3.2.1 Resultados de la Prueba:

Ilustración 16

Resultados prueba de carga consulta de informes de paquetes.

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento
Get reports	100	10	8	20	2,91	0,00%	10,1/sec
Total	100	10	8	20	2,91	0,00%	10,1/sec

El sistema paso satisfactoriamente esta prueba, cumpliendo en su totalidad las peticiones de creación de paquetes, mostrando un rendimiento bueno bajo la prueba de carga, con un 100% de peticiones exitosas y un tiempo de respuesta promedio de 10ms, tiempo sumamente bueno para la obtención de paquetes en un periodo de tiempo determinado.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones:

- Tras completar el desarrollo de una aplicación web para la gestión de los principales procesos logísticos de una empresa de Courier, se concluyó que los principales procesos de estas empresas pueden agruparse en cuatro atapas clave: recepción de paquetes, clasificación de paquetes, envió y logística del paquete. Este trabajo automatiza estas fases desarrollando un sistema de información que permite a los trabajadores gestionar la clasificación de los paquetes, mediante una interfaz intuitiva que aborde todas las etapas de logística de un envió. Además, proporciona a los clientes una experiencia sencilla y transparente respecto al seguimiento y estado de sus paquetes.
- Una vez desarrollado el levantamiento de requerimientos de los principales procesos logísticos, características, funciones y servicios necesarios de la empresa, se identificó como este paso en la metodología de desarrollo fue un factor invaluable. Esto gracias a que permitió entender en profundidad las necesidades de los usuarios finales, además de alinear una solución efectiva para todos aquellos involucrados en el proyecto, facilitando de esta manera el diseño de un sistema efectivo y funcional a la realidad del negocio.
- Al determinar las características funcionales y la interfaz de usuario para el aplicativo web, se aseguró de manera integral que la aplicación sea clara y manejable para un operador de la empresa Courier, permitiendo al empleado mantener un control y gestión efectivo de los diferentes paquetes manejados por la empresa, minimizando errores y fallos en la gestión de entregas y recepción. Sucesos que podrían acontecer al manejar estos procesos de manera manual en hojas de cálculo o archivos de texto plano y que afectan de manera negativa la relación con el cliente final, generando inseguridad e insatisfacción al servicio en general. De esta forma, la aplicación no solo optimiza las operaciones logísticas, sino que también mejora la confianza y satisfacción del cliente final con el servicio.

- Tras finalizar el prototipo funcional, resulta claro que, el uso del framework Vue.js para el desarrollo de interfaces, fue de suma utilidad, ya que permitió desarrollar entornos visuales más cómodos, intuitivos y accesibles desde cualquier dispositivo, facilitando de esta forma su control y gestión para los administradores y operativos del sistema. En conjunto, este framework ofrece una gran cantidad de componentes que fueron reutilizados alrededor del proyecto, ahorrando tiempo en el desarrollo, manteniendo un estándar de diseño y facilitando la capacidad de escalabilidad a futuro.

Recomendaciones

- Dado que este trabajo es un prototipo funcional, se recomienda completar los pasos sugeridos por la metodología, especialmente en las fases de pruebas e implementación. Estas acciones son cruciales para identificar áreas de mejora, optimizar el rendimiento del sistema e identificar posibles vulnerabilidades que puedan producirse al implementar el sistema. Además, es crucial planificar detalladamente la capacitación de los operativos del sistema, esto con el objetivo de evitar un rechazo de este y obtener una integración sin contratiempos.
- Gracias al impacto positivo que tuvo la utilización de frameworks de trabajo en el desarrollo del proyecto, se recomienda seguir utilizando Vue.js en la creación de interfaces de usuario dinámicas y responsivas, así como Django para la creación de aplicaciones web. La combinación de ambos marcos de trabajo ha demostrado formar un proyecto con componentes escalables, de manejo sencillo y que puedan adaptarse fácilmente a las necesidades del usuario y del negocio.
- Ya que este proyecto tiene el objetivo de escalar en conjunto con la empresa, se sugiere explorar la incorporación de módulos adicionales que puedan aportar un mayor valor al negocio. Entre las cuales se pueden destacar las notificaciones en tiempo real, para una comunicación instantánea con los clientes u operadores, o funciones de inteligencia artificial para la automatización de la clasificación de paquetes o incluso la implementación de modelos

predictivos para prever picos en la demanda y mejorar la respuesta del negocio ante una mayor actividad comercial.

- En búsqueda de crear módulos y funcionalidades que puedan agregar una ventaja competitiva al negocio, en un corto periodo de tiempo, es indispensable la utilización de una metodología ágil. Este enfoque no solo nos permitirá la capacidad de entregar resultados rápidos y efectivos, sino que también se alineará con las verdaderas necesidades del cliente, adaptando continuamente el producto en respuesta a la retroalimentación del mismo para garantizar un resultado final que realmente agregue valor y satisfaga las necesidades del entorno cambiante al que se enfrentan los negocios Courier.

REFERENCIAS

Amazon AWS. (6 de Febrero de 2020). *Amazon AWS.* Obtenido de Amazon AWS: https://aws.amazon.com/es/compare/the-difference-between-frontend-and-backend/#:~:text=El%20front%20end%20es%20aquello,permiten%20que%20la%20aplicaci%C3%B3n%20funcione.

Amazon AWS. (9 de Mayo de 2022). *Amazon AWS*. Obtenido de Amazon AWS: https://aws.amazon.com/es/compare/the-difference-between-logical-and-physical-data-model/

Amazon AWS. (4 de Junio de 2023). *Amazon AWS*. Obtenido de Amazon AWS: https://aws.amazon.com/es/what-is/django/

Barragan, A. (26 de Noviembre de 2021). *OpenWebinars.* Obtenido de OpenWebinars: https://openwebinars.net/blog/que-es-vue-js-y-que-lo-diferencia-de-otros-frameworks/#qu%C3%A9-es-vue-js

Bello, E. (28 de Abril de 2021). *IEBS.* Obtenido de Escuela de Negocios de la Innovación y los Emprendedores: https://www.iebschool.com/blog/que-es-el-xp-programming-agile-scrum/

Christie, T. (24 de Enero de 2024). *Django REST framework*. Obtenido de Django REST framework: https://www.django-rest-framework.org/

Corcuera, P. (22 de Octubre de 2018). Diseño de Bases de Datos Modelo Conceptual Entidad Relación. *Diseño de Bases de Datos Modelo Conceptual Entidad Relación*. Cantabria, Cantabria, España. Obtenido de https://personales.unican.es/corcuerp/BDT/Slides/BD_ModeloRelacional.pdf

Fadeke, A. (05 de Agosto de 2023). *Doist Inc*. Obtenido de Todoist: https://todoist.com/es/productivity-methods/kanban

GooApps. (27 de Octubre de 2022). *GooApps.* Obtenido de GooApps: https://gooapps.es/2022/10/27/las-5-mejores-metodologias-de-desarrollo-de-software/

Hurtado, J. (03 de Diciembre de 2021). *Logo IEBS*. Obtenido de Logo IEBS:

https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/

IBM. (16 de Marzo de 2021). *IBM*. Obtenido de IBM: https://www.ibm.com/docs/es/radfws/9.6.1?topic=modeling-physical-data-models

IBM. (12 de Septiembre de 2022). *IBM.* Obtenido de IBM: https://www.ibm.com/es-es/topics/rest-apis

Marvin, L. (18 de Septiembre de 2020). *OpenWebinars*. Obtenido de OpenWebinars: https://openwebinars.net/blog/extreme-programming-que-es-y-como-aplicarlo/

Montalvo, A. (2021). *El Universo*. Obtenido de El Universo: https://www.eluniverso.com/noticias/economia/couriers-estiman-que-importar-bajo-4x4-costara-entre-15-y-60-menos-por-eliminacion-de-tasa-de-42-nota/

Oracle. (2 de Febrero de 2018). *Oracle.* Obtenido de Oracle: https://www.oracle.com/mx/database/what-is-database/

Red Hat. (4 de Agosto de 2023). *Red Hat.* Obtenido de Red Hat: https://www.redhat.com/es/topics/cloud-native-apps/what-is-service-oriented-architecture

Sandoval, E. (21 de Septiembre de 2023). *Escuela Britanica de artes creativas y tecnología.* Obtenido de Ebac: https://ebac.mx/blog/frameworks#:~:text=Un%20framework%2C%20o%20un%20marco,la%20l%C3%B3gica%20de%20tu%20app.

Servicio Nacional de Aduana del Ecuador. (12 de Marzo de 2022). *Aguana gob ec.* Obtenido de Aguana gob ec: https://aduana.gob.ec/servicio-al-ciudadano/envios-courier-postal/

Stack Overflow. (24 de Febrero de 2023). *Stack Overflow.* Obtenido de Stack Overflow: https://survey.stackoverflow.co/2023/