



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Carrera de Ingeniería en Sistemas

Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless)

IT system for order management in physical coffee shop locations in the city of Loja applying distributed architecture (microservices and/or serverless)

Trabajo de Titulación, previa a
la obtención del título de
Ingeniera en Sistemas

AUTORA:

Maria Anabel Encalada Córdova

DIRECTOR:

Ing. Wilman Patricio Chamba Zaragocín, Mg.Sc.

Loja – Ecuador

2024

Certificación

Loja, 16 de agosto de 2023

Ing. Wilman Chamba Zaragocín.
DIRECTOR DE TRABAJO DE TITULACIÓN

CERTIFICO

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Titulación denominado: **“Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless)”**, previo a la obtención del título de **Ingeniera en Sistemas**, de la autoría del estudiante **Maria Anabel Encalada Córdova**, con **cédula de identidad Nro. 0707031662**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. Wilman Chamba Zaragocín.
DIRECTOR DE TRABAJO DE TITULACIÓN

Autoría

Yo, **María Anabel Encalada Córdova**, declaro ser autor del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mí Trabajo de Titulación en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

Cédula: 0707031662

Fecha: 29 de enero de 2024

Correo Electrónico: maria.a.encalada@unl.edu.ec

Celular: (+593) 099 794 0592

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo del Trabajo de Titulación

Yo **Maria Anabel Encalada Córdova**, declaro ser autor del Trabajo de Titulación denominado: **Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless)** como requisito para optar el título de **Ingeniero en Sistemas**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los veinte y nueve días del mes de enero de dos mil veinticuatro.

Firma:

Autora: Maria Anabel Encalada Córdova

Cédula de identidad: 0707031662

Fecha: 29 de enero de 2024

Dirección: Machala 88-05 y Guaranda, Loja - Ecuador

Correo Electrónico: maria.a.encalada@unl.edu.ec

Teléfono: (+593) 99 794 0592

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Wilman Patricio Chamba Zaragocín, Mg.Sc

Dedicatoria

Con profundo cariño y gratitud, dedico mi trabajo de titulación a mis amados padres, quienes han sido la fuente inagotable de inspiración y apoyo en cada paso de mi camino académico. Su amor incondicional, sacrificio y constante aliento han sido el motor que impulsó mi determinación y perseverancia. Este logro no solo es un testimonio de mi esfuerzo, sino también del amor y dedicación que ustedes han vertido en mí. Con este trabajo, honro su incansable devoción y sueños compartidos. Este éxito es un reflejo de la solidez de nuestros lazos familiares y un tributo a la confianza que depositaron en mí desde el principio. A ustedes, mis queridos padres, les dedico este logro con todo mi corazón.

Maria Encalada

Agradecimiento

En este trascendental capítulo de mi vida, quiero expresar mi profundo agradecimiento a mi amada familia. Mamá y papá, su inquebrantable apoyo y amor incondicional han sido mi mayor fortaleza. Mis queridos hermanos y hermana, su aliento constante y la confianza que depositaron en mí me motivaron a superar cada obstáculo. A mi familia extendida y amigos/as cercanos, su presencia en cada etapa de mi camino, brindando consuelo y alegría, me ha recordado la importancia de la comunidad en la realización de los sueños. Un agradecimiento especial se reserva para mi ángel, quien, sin nombre, pero con un corazón generoso, fue un pilar fundamental en mi travesía hacia este logro.

Mi gratitud se extiende a los respetados profesores de esta prestigiosa Universidad, quienes no solo compartieron su conocimiento experto, sino que también guiaron mi crecimiento académico y personal. Es imposible no mencionar al Ing. Wilman Chamba, mi director de tesis, cuya confianza en mis capacidades y apoyo constante me permitieron enfrentar los desafíos con determinación. Su orientación iluminó mi camino y sus valiosos consejos me impulsaron a alcanzar un nivel más alto de excelencia.

Al concluir este camino lleno de aprendizaje y dedicación, reflexiono sobre las palabras de apoyo y el respaldo inquebrantable que he recibido. Cada gesto de aliento, cada conversación inspiradora y cada muestra de cariño han dejado una huella imborrable en mi corazón. Hoy, al celebrar este logro, quiero expresar mi sincero agradecimiento a mi familia, amigos/as y profesores, así como a mi ángel, por creer en mí cuando más lo necesitaba. Este logro no solo es mío, sino de todos ustedes que han estado a mi lado en este emocionante viaje. ¡Gracias por ser mi fuente de fuerza y motivación constante!

Maria Encalada

Índice de Contenidos

Portada	i
Certificación	ii
Autoría	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de Contenidos	vii
Índice de tablas:.....	ix
Índice de figuras:.....	ix
Índice de anexos:.....	x
1. Título	1
2. Resumen	2
2.1. Abstract.....	3
3. Introducción	4
4. Marco Teórico	6
4.1. Antecedentes.....	6
4.1.1. Cafetería universitaria.....	6
4.1.2. Artesanale.....	6
4.1.3. Otros.....	7
4.2. Ingeniería de Software.....	7
4.2.1. Metodologías de desarrollo de software.....	7
4.2.2. Metodología ICONIX.....	9
4.3. Tecnologías de desarrollo de software.....	11
4.3.1. JavaScript.....	11
4.3.2. Microservicios y Arquitectura Distribuida.....	13
4.4. Trabajos relacionados.....	15
4.4.1. Menú de tableta Menulux.....	15
4.4.2. Menú de entrega y recogida en línea.....	16
4.4.3. Eatsa.....	16
5. Metodología	17
5.1. Área de estudio.....	17
5.2. Procedimiento.....	17
5.2.1. Construir el modelo de arquitectura física y lógica en base a los requerimientos del usuario utilizando el Modelamiento de desarrollo de software dirigido por ICONIX.....	17
5.2.2. Implementar la arquitectura de la solución informática en una aplicación web mediante el uso de tecnologías JavaScript y servicios de la nube con la metodología ICONIX.....	17

5.2.3.	Comparar los procesos de toma de pedidos tradicional y web en un ambiente de pruebas simulado dentro de la FEIRNR para determinar los beneficios de la actualización.	18
5.3.	Recursos.....	18
5.3.1.	Recursos Científicos.....	18
5.3.2.	Recursos Técnicos.....	19
5.3.3.	Recursos de Hardware y Software.....	19
5.4.	Participantes.....	19
6.	Resultados.....	20
6.1.	Objetivo 1: Construir el modelo de arquitectura física y lógica en base a los requerimientos del usuario utilizando el Modelamiento de desarrollo de software dirigido por ICONIX.....	20
6.1.1.	Análisis y diseño preliminar.....	20
6.1.2.	Diseño de la arquitectura física y lógica.....	24
6.2.	Objetivo 2: Implementar la arquitectura de la solución informática mediante el uso de tecnologías JavaScript y la metodología ICONIX.....	44
6.2.1.	Configuración de espacio de trabajo.....	44
6.2.2.	Servicios.....	47
6.2.3.	Servicio FrontEnd.....	51
6.2.4.	Realizar pruebas de funcionamiento del software.....	57
6.3.	Objetivo 3: Comparar los procesos de toma de pedidos tradicional y web en un ambiente de pruebas simulado dentro de la FEIRNR para determinar los beneficios de la actualización.....	59
6.3.1.	Planificación del caso de prueba para la simulación manual versus web.....	59
6.3.2.	Ejecución de la simulación.....	61
6.3.3.	Establecer un análisis cuantitativo de los resultados.....	63
7.	Discusión.....	68
7.1.	Objetivo 1: Construir el modelo de arquitectura física y lógica en base a los requerimientos del usuario utilizando el Modelamiento de desarrollo de software dirigido por ICONIX.....	68
7.2.	Objetivo 2: Implementar la arquitectura de la solución informática mediante el uso de tecnologías JavaScript y la metodología ICONIX.....	69
7.3.	Objetivo 3: Realizar pruebas de funcionamiento del software.....	69
8.	Conclusiones.....	71
9.	Recomendaciones.....	72
9.1.	Trabajos Futuros.....	72
10.	Bibliografía.....	73
11.	Anexos.....	76

Índice de tablas:

Tabla 1. Comparación de metodologías de desarrollo de software.....	8
Tabla 2. Requerimientos funcionales.....	21
Tabla 3. Requerimientos funcionales.....	21
Tabla 4. Requerimientos funcionales.....	22
Tabla 5. Requerimientos funcionales.....	22
Tabla 6. Glosario de términos.....	23
Tabla 7. Descripción de caso de uso - Iniciar sesión.....	25
Tabla 8. Descripción de caso de uso - Registrar cuentas de usuario.....	26
Tabla 9. Descripción de caso de uso - Registrar cuenta de usuario	28
Tabla 10. Descripción de caso de uso - Registrar Categoría	29
Tabla 11. Descripción de caso de uso - Registrar producto	30
Tabla 12. Descripción de caso de uso - Agregar productos al pedido	31
Tabla 13. Descripción de caso de uso - Generar comprobante de pago	31
Tabla 14. Servicios, objetivo y estilo arquitectónico	42
Tabla 15. Planificación de desarrollo de servicios	44
Tabla 16. Comité del proyecto.....	59
Tabla 17. Cronograma de actividades	60
Tabla 18. Banco de preguntas de la encuesta de satisfacción.....	61
Tabla 19. Escenarios de prueba para la comparación del proceso tradicional vs web	62
Tabla 20. Promedio de aceptación de la Encuesta de Satisfacción	64
Tabla 21. Descripción de caso de uso - Ver información de cuenta de usuario.....	106
Tabla 22. Resumen de resultados de la Encuesta de Satisfacción.....	143

Índice de figuras:

Figura 1. Modelado del dominio – Usuarios	24
Figura 2. Modelado del dominio – negocio.....	24
Figura 3. Diagrama de casos de uso - General.....	25
Figura 4. Diagrama de casos de uso - Gestión de cuentas de usuarios.....	26
Figura 5. Diagrama de casos de uso - Gestión de cuenta de usuario	28
Figura 6. Diagrama de casos de uso - Gestión de categorías	29
Figura 7. Diagrama de casos de uso - Gestión de productos	30
Figura 8. Diagrama de casos de uso - Gestión de pedido	31
Figura 9. Diagramas de secuencia - Registrar cuentas de usuario.....	33
Figura 10. Descripción de caso de uso - Generar comprobante de pago.....	34
Figura 11. Diagramas de secuencia - Registrar categoría	35
Figura 12. Diagramas de secuencia - Registrar producto	35
Figura 13. Diagramas de secuencia - Agregar producto al pedido	36
Figura 14. Diagramas de secuencia - Generar comprobante de pago	36
Figura 15. Diagrama de paquetes de microservicios	37
Figura 16. Modelo estático final del dominio	38
Figura 17. Modelo estático final de usuarios	39
Figura 18. Modelo estático final de productos.....	39
Figura 19. Modelo estático final de pedidos	40
Figura 20. Diagrama de componentes de microservicios	41
Figura 21. Representación gráfica del comportamiento del bus de eventos.....	41
Figura 22. Diagrama de despliegue de microservicios.....	43
Figura 23. Espacio de trabajo general.....	44
Figura 24. Configuración de archivo Docker	45

Figura 25. Espacio de trabajo de configuración del clúster Kubernetes	45
Figura 26. Configuración de puerta de enlace Ingress.....	46
Figura 27. Manifiestos Kubernetes de Deployment y Service.....	47
Figura 28. Espacio de trabajo de servicio órdenes	48
Figura 29. Modelo Orden en el servicio de órdenes.....	48
Figura 30. Controlador para obtener todas las órdenes en el servicio de órdenes.....	49
Figura 31. Rutas del servicio de órdenes	50
Figura 32. Directorio del bus de eventos.....	50
Figura 33. Escuchador de cambios en el registro de productos	51
Figura 34. Publicador de cambios en el registro de producto	51
Figura 35. Espacio de trabajo de FrontEnd.....	52
Figura 36. Middleware de autenticación.....	52
Figura 37. Espacio de trabajo de componentes contenedores de páginas.....	53
Figura 38. Componente contenedor por defecto de la aplicación	53
Figura 39. Consumo de API Rest.....	54
Figura 40. Peticiones de tipo lectura (GET)	54
Figura 41. Peticiones de tipo escritura (POST y PUT).....	55
Figura 42. Espacio de trabajo de páginas	55
Figura 43. Página Cambiar clave	56
Figura 44. Espacio de trabajo del proyecto de intercambio de recursos.....	57
Figura 45. Estructura de archivos de prueba	57
Figura 46. Resultado de pruebas unitarias/integración del servicio de usuarios	58
Figura 47. Resultado de pruebas unitarias/integración del servicio de productos	58
Figura 48. Resultado de pruebas unitarias/integración del servicio de pedidos	58
Figura 49. Resumen del tiempo empleado en el registro y actualización de menú en el método tradicional frente al proceso web.....	65
Figura 50. Resumen del tiempo empleado en la toma de pedidos en el método tradicional frente al proceso web.	65
Figura 51. Resumen del tiempo empleado en el cobro por consumo en grupo o individual en el método tradicional frente al proceso web.....	66
Figura 52. Promedio del tiempo empleado en el registro y actualización de menú en el método tradicional frente al proceso web.....	66
Figura 53. Promedio del tiempo empleado en la toma de pedidos en el método tradicional frente al proceso web.	67
Figura 54. Promedio del tiempo empleado en el cobro por consumo en grupo o individual en el método tradicional frente al proceso web.....	67

Índice de anexos:

Anexo 1. Entrevistas para la obtención de requerimientos	76
Anexo 2. Menús de Cafeterías	83
Anexo 3. Modelado de la arquitectura lógica	97
Anexo 4. Manual de instalación del sistema informático	127
Anexo 5. Modelo de Encuesta de Satisfacción	135
Anexo 6. Resultados de la Encuesta de Satisfacción	137
Anexo 7. Acta de validación de pruebas del sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless).....	144
Anexo 8. Manual de usuario del sistema informático	160
Anexo 9. Certificado de traducción del resumen.....	172

1. Título

Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless)

IT system for order management in physical coffee shop locations in the city of Loja applying distributed architecture (microservices and/or serverless)

2. Resumen

Las tecnologías han ganado lugar en la cotidianidad del día a día y esto se hizo mucho más evidente durante la pandemia cuando muchos procesos debieron ser gestionados por medio de TICs. Por otro lado, las cafeterías enfrentaron dificultades en la gestión de sus pedidos con las demandas de seguridad sanitaria que representaba su reactivación. Considerando esto surgió la siguiente pregunta de investigación: ¿Cómo gestionar el proceso de toma de pedidos en los locales físicos de Cafeterías? Por ello, el presente Trabajo de Titulación (TT) tiene como objetivo principal el desarrollar una aplicación informática para la gestión de pedidos en los locales físicos de cafeterías utilizando arquitecturas distribuidas (Microservicios y/o serverless), aplicando las actividades de la metodología ICONIX y tecnologías como: el lenguaje de programación JavaScript, Kubernetes y Docker para su despliegue. Para cumplir con este objetivo, se consideraron 3 fases. En la primera fase, se hizo un análisis de casos de estudio que sirvieran como referencia para determinar los requerimientos en los procesos de la gestión de pedidos en cafeterías, se hizo un modelado de software dirigido por la metodología ICONIX, obteniendo los diagramas UML que representaron la arquitectura física y lógica. En la segunda fase se implementó la arquitectura obtenida en una aplicación web usando el stack MERN (MongoDB, Express.js, React, Node.js), dando como resultado una solución informática funcional que cumplió con los requerimientos y los modelos obtenidos durante la primera fase. La fase final consistió en comparar los procesos de gestión de pedidos tradicional y web en un ambiente de pruebas simulado que permitió dar respuesta a la pregunta de investigación planteada.

Palabras claves: *Microservicios, Docker, Kubernetes, ICONIX, arquitectura distribuida.*

Abstract

Technologies have gained a prominent role in everyday life, and this became even more evident during the pandemic when many processes had to be managed through ICTs. On the other hand, cafés faced challenges in managing their orders due to the sanitary safety demands associated with their reactivation. Considering this, the following research question emerged: How to manage the order-taking process in physical café locations? Therefore, the main objective of this Thesis Work (TW) is to develop a computer application for order management in physical café locations using distributed architectures (Microservices and/or serverless), applying activities from the ICONIX methodology and technologies such as the JavaScript programming language, Kubernetes, and Docker for deployment.

To achieve this objective, three phases were considered. In the first phase, an analysis of case studies was conducted to serve as references for determining the requirements in the processes of order management in cafés. Software modeling was carried out guided by the ICONIX methodology, resulting in UML diagrams that represented the physical and logical architecture. In the second phase, the obtained architecture was implemented in a web application using the MERN stack (MongoDB, Express.js, React, Node.js), resulting in a functional software solution that met the requirements and models obtained during the first phase. The final phase involved comparing traditional and web-based order management processes in a simulated test environment, which allowed for addressing the research question posed.

Keywords: *Microservices, Docker, Kubernetes, ICONIX, distributed architecture.*

3. Introducción

Las cafeterías dentro de la ciudad de Loja, Ecuador, se caracterizan por la venta de platos, bebidas y postres tanto nacionales como internacionales. Sin embargo, durante el periodo de pandemia, estos establecimientos tuvieron que ajustarse a medidas sanitarias entre ellas el distanciamiento social. Esto, poco a poco ha ido creando una nueva necesidad de una alternativa para la toma de pedidos. Se han visto varias y diversas opciones, como menús digitalizados, menús de un solo uso, hojas emplastizadas o grandes impresiones de materiales resistentes al alcohol desinfectante o productos desinfectantes.

Esto propició que la atención al cliente se deba actualizar y emplear Tecnologías de la Información y Comunicación (TICs) como valor agregado. En virtud de ello nace el Trabajo de Titulación (TT): “Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitecturas distribuidas (microservicios y/o serverless)”. Esta solución permitió a varias cafeterías gestionar sus pedidos con mejores respuestas en tiempos empleados en la atención a sus clientes, así mismo, ofreció una plataforma para mejorar el control y visualización del estado de las órdenes.

Por lo anterior, el presente TT fue propuesto con el fin de dar respuesta a la pregunta de investigación: ¿Cómo gestionar el proceso de toma de pedidos en los locales físicos de Cafeterías? El desarrollo tuvo como objetivo principal “Desarrollar una aplicación informática para la gestión de pedidos en los locales físicos de cafeterías utilizando arquitecturas distribuidas (Microservicios y/o serverless)”. Su consecución se logró por medio de 3 objetivos específicos, estos fueron 1) Construir el modelo de arquitectura física y lógica en base a los requerimientos del usuario utilizando el Modelamiento de desarrollo de software dirigido por ICONIX, 2) Implementar la arquitectura de la solución informática en una aplicación web mediante el uso de tecnologías JavaScript y servicios de la nube con la metodología ICONIX y 3) Comparar los procesos de toma de pedidos tradicional y web en un ambiente de pruebas simulado dentro de la FEIRNR para determinar los beneficios de la actualización.

Para el desarrollo de los objetivos específicos se empezó la etapa de Modelado: análisis y diseño descrita por la metodología ICONIX dando como resultado el modelo de arquitectura física y lógica en base al análisis requerimientos obtenido del estudio de 5 cafeterías. Posteriormente se siguieron las fases de Inicialización y Construcción establecidas por ICONIX desarrollando un sistema informático con el uso del stack MERN y la utilización de Kubernetes y Docker para la implementación de Microservicios. Por último, en cuanto a comparar los procesos de gestión de pedidos tradicional y web, se determinó realizar pruebas en un ambiente simulado planteando escenarios que describían las actividades más

relevantes de la metodología propuesta por el autor Pablo González [1]. Todos estos apartados se los puede evidenciar en la sección de Resultados del presente TT.

En cuanto al desarrollo del Trabajo de Titulación se encuentra organizado de la siguiente manera:

La sección Marco Teórico, está compuesta por los antecedentes referentes al tema y trabajos relacionados; la sección Metodología se detalla el contexto en donde se realizó el TT, el procedimiento y los recursos que fueron utilizados; la sección Resultados contiene el desarrollo de los tres objetivos específicos planteados para TT donde se evidencian: los requerimientos funcionales y no funcionales, el desarrollo de la metodología ICONIX, diseño y documentación de la arquitectura lógica y física de la solución a través de los modelos de dominio, casos de uso, diagrama de secuencia, diagrama de clases, diagrama de componentes y diagrama de despliegue, la codificación aplicando una arquitectura de Microservicios; y finalmente la ejecución de los pruebas de integración y funcionales que demuestran el cumplimiento de los requerimientos funcionales y no funcionales del sistema; la sección Discusión, contiene un análisis de los resultados obtenidos de aplicar las fases de la metodología ICONIX, el proceso ordenado que se llevó, las tecnologías que se adaptaron en el desarrollo de la aplicación y las pruebas ejecutadas; la sección Conclusiones abarca los acontecimientos más sobresalientes del proyecto, entre ellos se concluye que el sistema ofrece una alternativa estandarizada o común que puede ser usada por cualquier cafetería dentro y fuera de la Universidad Nacional de Loja. Además, también puede ser usada por restaurantes y bares en la gestión de sus pedidos favoreciendo el distanciamiento social y administración adecuada de su negocio; y finalmente en la sección Recomendaciones posee las sugerencias y posibles trabajos futuros aplicables para la continuidad del proyecto.

4. Marco Teórico

En esta sección, se exponen los conceptos fundamentales que constituyen los pilares del desarrollo del presente Trabajo de Titulación. Además, se llevó a cabo una observación detallada de diversas cafeterías y establecimientos de comida, con el propósito de adquirir un entendimiento inicial del flujo de procesos que guía la interacción con sus clientes.

4.1. Antecedentes

4.1.1. Cafetería universitaria

La cafetería universitaria asociada a la carrera de Turismo, inaugurada en el año 2022 con la reintegración de las actividades académicas presenciales en la Universidad Nacional de Loja, desempeña un rol vital en el ecosistema estudiantil. Su operación, influenciada por el dinamismo propio del ámbito universitario, se caracteriza por ofrecer un menú ininterrumpido, en sintonía con las necesidades y horarios cambiantes de la comunidad estudiantil. Esta interacción constante con un público diverso y en movimiento, proporciona una perspectiva esencial para comprender las dinámicas y desafíos que enfrenta la gestión de pedidos en un entorno de cafetería universitaria [2].

4.1.2. Artesanale

Artesanale es una red de restaurantes y cafeterías dentro de la ciudad de Loja, Ecuador. Se caracteriza por la venta de platos, bebidas y postres tanto nacionales como internacionales, dentro de sus productos más característicos están: Pizzas, crepes, helados, tamales, humitas y amplia variedad de café. Además, a lo largo de los años que llevan en el mercado, han buscado innovar y atraer a más clientes con platos y con servicios modernos.

Antes del inicio de la pandemia mundial causada por el COVID-19, contaban con 4 locales físicos (24 de mayo & Zoilo Rodríguez, 10 de agosto entre Sucre y Bolívar, Antonio Eguiguren y 18 de noviembre, Av. Gobernación de Mainas y 18 de Noviembre dentro de supermaxi) desde donde atendían a sus clientes, con la llegada del confinamiento, en marzo el 2020 y debido a la complejidad de atención que representó el distanciamiento social, uno de sus locales tuvo que cerrar (Antonio Eguiguren y 18 de noviembre) y en los 3 restantes se redujo el flujo de visitantes.

Como respuesta a las necesidades de cuidado hacia el personal como a los clientes, se han implementado menús digitalizados, mismos que consisten en documentos en formato PDF alojados en Google Drive a los que se accede mediante códigos QR ubicados en los locales físicos de Artesanale, Estas cartas digitales, sólo son de tipo informativas, es decir, no

se puede seleccionar algún producto desde la vista del menú, además de estar implementados con imágenes propias de cada ítem.

Sin embargo, un factor a favor de Artesanale es el conocimiento propio que tienen de sus menús, con categorías claras y productos fáciles de buscar con la clasificación actual, siendo estas: platos típicos, sándwiches, pastas y lasañas, ensaladas, gelatos, postres y dulces, pizzas, crepes y bebidas. Por cada categoría, se detalla el nombre del producto, su precio y, en algunos casos, una descripción de lo que contiene el plato.

4.1.3. Otros

Al abordar los modelos que han sido implementados para presentar los menús a raíz de la pandemia del COVID-19 en las cafeterías locales, se evidencia una tendencia hacia la adopción de menús digitalizados, accesibles a través de códigos QR ubicados en las mesas, así como la utilización de hojas emplastizadas con la lista de platos y bebidas. En ambos enfoques, prevalece un compromiso compartido por salvaguardar tanto a los colaboradores como a los clientes, ofreciendo una experiencia de atención adaptada a la situación actual [3].

Además, se han identificado casos en los que la crisis sanitaria ha brindado la oportunidad de reevaluar y modificar recursos, tal es el ejemplo de la sustitución de los tradicionales "individuales" de mesa por menús de un solo uso. Esta innovación consiste en imprimir la oferta gastronómica en hojas simples y en tonos de gris, y tras la toma del pedido, la hoja no se retira, sino que se mantiene disponible para ser utilizada como parte de la sobremesa. Estas adaptaciones revelan la capacidad de la industria para reimaginar sus prácticas en función de las circunstancias actuales, manteniendo siempre la atención en la seguridad y satisfacción de los clientes [4].

4.2. Ingeniería de Software

4.2.1. Metodologías de desarrollo de software

Las metodologías ayudan a guiar el trabajo a desarrollar y son básicamente estructuras que detallan pasos a seguir para conseguir un producto. De este conjunto de métodos que se especifica en una metodología, es de donde se comienza a realizar entregables o insumos del software que se pretende construir [5].

Actualmente, existen varias metodologías por las cuales se puede optar al desarrollar, pero la decisión radica en las necesidades observadas en el primer análisis del problema. Por lo tanto, a continuación, se verá una comparación de una metodología tradicional, una ágil y una que es pesada-ligera o semi-ágil, es decir, hereda características de las metodologías tradicionales y de las ágiles.

Tabla 1. Comparación de metodologías de desarrollo de software

	RUP	XP	ICONIX
Características	<p>Secuencial</p> <p>Estructurada</p> <p>El cliente no forma parte activa del proceso, al final de todo el desarrollo se hace entrega del producto terminado</p> <p>Ideal para proyectos bien definidos y sin posibles cambios.</p> <p>Costo de cambio elevado.</p>	<p>Define roles en el proyecto (programadores, cliente, entrenador o líder, rastreador, probador)</p> <p>Flexible al cambio</p> <p>Basado en iteraciones</p> <p>El cliente forma parte del equipo.</p> <p>Costo de cambio bajo.</p>	<p>Prioridad a la fase de análisis y diseño</p> <p>Desarrollo basado en iteraciones</p> <p>El cliente forma parte del equipo</p> <p>Responde al cambio en tiempo de ejecución.</p> <p>Costo de cambio bajo.</p>
Fases	<p>Inicio</p> <p>Elaboración</p> <p>Construcción</p> <p>Transición.</p>	<p>Planificación</p> <p>Diseño</p> <p>Codificación</p> <p>Pruebas</p> <p>Lanzamiento.</p>	<p>Análisis de requerimientos</p> <p>Análisis y diseño preliminar</p> <p>Diseño detallado</p> <p>Implementación.</p>
Procesos	<p>Modelado de negocio</p> <p>Requerimientos</p> <p>Análisis y diseño</p> <p>Implementación</p> <p>Pruebas</p> <p>Entrega</p> <p>Configuración de cambios y administración</p> <p>Administración de proyecto</p> <p>Ambiente.</p>	<p>Requerimientos en historias de usuario</p> <p>Criterios de pruebas de adaptación</p> <p>Plan de iteración</p> <p>Diseño simple</p> <p>Programación en parejas</p> <p>Pruebas de adaptación</p> <p>Incremento del software.</p>	<p>Modelado del dominio</p> <p>Prototipado</p> <p>Modelo de casos de uso</p> <p>Descripción de casos de uso</p> <p>Diagramas de robustez</p> <p>Diagramas de secuencia</p> <p>Escribir/generar código.</p>
Conclusión	<p>Las fases respetan un orden secuencial</p> <p>Un error en alguna de las fases, puede representar fallas más grandes en las fases próximas</p>	<p>Ofrece una respuesta al cambio a medida que se da el proceso de desarrollo.</p> <p>Prioriza la comunicación con el cliente.</p>	<p>Es una metodología ágil que presta atención y cuidado a la fase de análisis y diseño, antes de comenzar a codificar.</p> <p>Se ajusta a cualquier tipo de proyecto, sea</p>

	<p>Tiempos de espera largos hasta que el cliente pueda ver resultados.</p> <p>Es posible que el resultado final no sea lo que el cliente realmente espera.</p>	<p>Se adapta a las condiciones del problema.</p>	<p>este grande o pequeño.</p> <p>Es flexible y adopta el modelo de desarrollo iterativo, es decir, entrega incrementos a medida que se avanza.</p>
--	--	--	--

Por lo antes mencionado, se opta por el uso de ICONIX en el presente trabajo, debido a que incluye las mejores características de RUP y XP, siendo flexible al cambio, basada en iteraciones, sin descuidar la importancia de tener un buen análisis y diseño antes de la fase de codificación [6].

4.2.2. Metodología ICONIX

Esta metodología es de tipo pesado-ligero, porque está compuesta de características de RUP y XP y propone el uso de un subconjunto de determinado de lo más importante y útil de UML [7]. Para mejorar sus procesos, se salta u omite, por ejemplo, los diagramas de colaboración y de estado, por considerarlos redundantes.

Los diagramas que sí son considerados dentro de esta metodología son:

- Diagramas de clases, tanto en el análisis (dominio del problema) y diseño.
- Casos de uso (diagramas y textos).
- Los diagramas de secuencia.
- Diagramas de robustez.

Los diagramas de robustez son un tipo de diagrama UML que proporcionan una representación visual de la interacción entre objetos y la distribución de responsabilidades. Sin embargo, estos no son estrictamente necesarios en la metodología. Esto se debe a que los diagramas de secuencia ya proporcionan información sobre las interacciones entre los objetos de un sistema. En particular, los diagramas de secuencia muestran las cadenas de mensajes que se intercambian entre los objetos durante la ejecución de un caso de uso. Esto proporciona una visión más completa de cómo se comportan los objetos del sistema [7]. Por lo tanto, la omisión de los diagramas de robustez en la metodología ICONIX no representa una pérdida significativa de información. De hecho, puede simplificar el proceso de desarrollo al reducir la cantidad de documentación que se debe crear.

Asimismo, esta metodología no abarca de manera directa y a detalle las cuestiones de alto nivel, es decir, la organización del equipo de trabajo. Un sistema que cumpla con los requerimientos es lo más importante en ICONIX [8]. Por lo que difiere de las metodologías pesadas ya que no exige mucha documentación, como sí sucede en RUP, y se centra más bien en conseguir el código fuente tan pronto como sea posible, como lo hace XP [9].

4.2.2.1. Fases de la metodología ICONIX

La metodología no impone ningún tipo de restricción en cómo está organizado un proyecto en cuanto a sus fases. Lo que la metodología sí exige es que previo a la codificación del proyecto exista un análisis de los requisitos y un trabajo de diseño [7]. Sin embargo, propone de manera general las siguientes fases:

- Modelado: análisis y diseño
- Planificación: inicialización
- Planificación: construcción

Modelado: análisis y diseño

Aquí se sitúa la esencia fundamental de la metodología, ya que en esta etapa convergen tres de los componentes fundamentales de ICONIX [10]. Esta fase se desglosa en tres subfases, a saber:

Análisis de requerimientos: el primer levantamiento de las necesidades a cubrir con el proyecto, es decir, identificar objetos del mundo real con diagramas de modelado de dominio, reglas y casos de uso.

Análisis preliminar: depurar la lista de requerimientos y construir la ficha de casos de uso, se puede valer de algunas interfaces de usuario para una mejor comprensión tanto para el equipo como para el cliente.

Diseño detallado: ideal para determinar el comportamiento de los objetos mediante modelos de interacción, es decir, diagramas de secuencia, modelos estructurales, con ayuda de diagramas de paquetes y de clases, también se define la arquitectura física de la solución mediante diagramas de componentes y de despliegue.

Planificación: inicialización

Esta etapa marca el inicio del proyecto, donde el equipo comienza a explorar los requerimientos y casos de uso, al mismo tiempo que se familiariza con las tecnologías que se implementarán en el desarrollo del producto. Durante esta fase, se lleva a cabo la creación del prototipo del sistema, una tarea fundamental para visualizar y validar conceptos clave. Es

importante destacar que esta fase se ejecuta únicamente una vez en todo el proyecto, estableciendo así las bases sólidas para las fases subsiguientes.

Planificación: construcción

Aquí es donde más se destaca la incidencia de XP en esta metodología, ya que su dinámica consiste en ofrecer pequeñas series de lanzamientos (esto en las metodologías ágiles, se denomina “incremento”). El mantenimiento se considera parte de esta fase puesto que desde la primera versión que llega al cliente el producto entra en mantenimiento.

4.3. Tecnologías de desarrollo de software

4.3.1. JavaScript

JavaScript es un lenguaje de programación que permite implementar funciones complejas en las páginas web. Cada vez que un sitio web hace algo más que mostrar información estática, como mostrar actualizaciones de contenido oportunas, mapas interactivos, gráficos 2D/3D animados, video jukeboxes de desplazamiento, etc., puede apostar que JavaScript está probablemente involucrado [11]. Es la tercera capa de la tarta de tecnologías web estándar, dos de las cuales son HTML (HyperText Markup Language - Lenguaje de marcado de hipertexto) y CSS (Cascading Style Sheets - Hojas de estilo en cascada).

JavaScript, el estándar de la web, desde sus inicios en 1995 hasta la actualidad se ha posicionado como uno de los lenguajes de programación con mayor aceptabilidad en el año 2021, tanto que el 68.62% de 58,031 desarrolladores profesionales lo usan, según los datos de Stack Overflow [12], sitio de consultas relacionadas a desarrollo de software, siendo el top 1, mientras que lenguajes de programación como Python y Java se ubican en las posiciones 4 y 7 respectivamente. Además, 13,719 empresas de todo el mundo usan JavaScript en sus desarrollos, según Stackshare [13], entre las empresas que más destacan están Paypal, Walmart, Instagram, LinkedIn, Netflix, Groupon y otros.

Adicionalmente y considerando las nuevas tendencias de los usuarios por el uso de herramientas basadas en la web [14], JavaScript y sus marcos de trabajo se han impuesto en los últimos años como la herramienta favorita para desarrollar software del lado del cliente [15][16]. Pero no solo aplica a *frontend*, sino que también se ha vuelto popular en la construcción de los servicios y APIs REST (*Representational State Transfer* - Transferencia de Estado Representacional) con su entorno de ejecución Node.js.

4.3.1.1. Node.js

Es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones *network* escalables. Asimismo, pueden atenderse muchas conexiones simultáneamente. Por cada conexión, se activa la devolución de llamada (*callback* por su traducción al inglés), pero si no hay trabajo que hacer, Node.js se entrará en reposo [17].

JavaScript es el lenguaje de programación más popular dentro del desarrollo web, y gracias a Node.js se ha utilizado ampliamente en el lado del servidor, siendo a día de hoy el más popular junto al framework Express.js, el mismo ocupa el puesto 7 de los repositorios de mayor aceptación en la plataforma GitHub. Con 23 millones de descargas semanales, ExpressJS ha llegado al top 5 en 2021 en favoritos, según Stack Overflow, y el puesto 4 entre los más queridos. Además, Node.js ocupa el lugar 8 entre los puestos de trabajo más demandados en USA [18].

Express.js

Express es una infraestructura de aplicaciones web de Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. Con miles de métodos de programa de utilidad, Protocolo de Transferencia de Hipertexto (HTTP) y middleware a su disposición, la creación de una API sólida es rápida y sencilla. Express.js proporciona una delgada capa de características de aplicación web básicas, que no ocultan las características provenientes de Node.js [19].

Express.js se ha convertido en uno de los frameworks más utilizados dentro del desarrollo de aplicaciones web, gracias a que es minimalista y flexible, y al ser de código abierto ha permitido que miles de desarrolladores en el mundo contribuyan para agregar cada vez nuevas funcionalidades, incrementando su poderío. Se encuentra alojado en el entorno de ejecución de Node.js, pero equipando un conjunto versátil de funcionalidades muy útiles para la eficaz construcción de aplicaciones web y móviles, lo que lo convierte en la herramienta más preferida por los programadores [20][21].

4.3.1.2. React

Es una librería de JavaScript, dos de las aplicaciones web más utilizadas en el mundo están desarrolladas con React, estas son Facebook e Instagram, por supuesto entre muchas otras, que utilizan la capacidad interactiva que proporciona esta librería. Si el objetivo es lanzar una web robusta, dinámica y escalable con un rendimiento fiable, la elección debe ser React.js. Otra característica a destacar es la capacidad de traer elementos reutilizables [22].

Esto permite a los desarrolladores reutilizar estos componentes en varias aplicaciones. Por último, destacar que ofrece un conjunto de paquetes dedicados al diseño UI (*User Interfaces*), como por ejemplo Material-UI, el cual se inspira en el diseño de materiales de Google [23].

Una de las librerías más jóvenes de JavaScript, React, desde su liberación en el 2013 a la comunidad por parte de Facebook, sus creadores, se ha convertido en una de las opciones favoritas para desarrollar del lado del cliente siendo el top 1, entre los desarrolladores que participan de la comunidad Stack Overflow, seguido de Vue.js y Angular, frameworks de JavaScript, mientras que Django, Flask y Laravel ocupan las posiciones 3, 6 y 15 respectivamente. Incluso muchos sitios de aficionados al desarrollo de software mencionan a React entre sus 3 preferidos [12]. Además, según datos de GitHub, React es el segundo marco web con más estrellas en sus repositorios. Según *Job Openings* en los Estados Unidos, los puestos más demandados son para desarrolladores de React.

4.3.2. Microservicios y Arquitectura Distribuida

4.3.2.1. Docker

En el año 2013, Docker revolucionó el paisaje del desarrollo al introducir el innovador concepto de contenerización. Esta novedosa perspectiva se ha convertido en un estándar esencial con notables beneficios para los desarrolladores. La contenerización facilita la creación de entornos altamente controlados, permitiendo la encapsulación de sistemas operativos, aplicaciones web y otros componentes en unidades autónomas y portátiles denominadas contenedores[24]. Esta plataforma, notablemente compatible con los sistemas operativos Linux y Windows, ha habilitado una gestión más eficiente y ágil de los recursos informáticos, fomentando la reproducibilidad, escalabilidad y aislamiento de componentes en el entorno de desarrollo y despliegue. La proliferación de Docker ha otorgado a los equipos de desarrollo una herramienta poderosa para agilizar el ciclo de desarrollo y mejorar la consistencia en el despliegue de aplicaciones, factores cruciales en la implementación exitosa de una arquitectura distribuida basada en microservicios o modelos serverless [25].

4.3.2.2. Kubernetes

Kubernetes, una destacada plataforma de orquestación de contenedores de código abierto, fue inicialmente desarrollada por Google y actualmente está bajo el cuidado y la evolución continua de la Cloud Native Computing Foundation. Esta poderosa herramienta ha redefinido la manera en que las aplicaciones son desplegadas, escaladas y administradas de manera automatizada en entornos de arquitecturas distribuidas [26].

Al abrazar Kubernetes, se abren las puertas hacia un enfoque más eficiente y eficaz en la gestión de componentes. Al transformar los procesos asociados con la incorporación de

complementos y bibliotecas en el despliegue de aplicaciones, Kubernetes ofrece una metodología consistente y escalable para gestionar el ciclo de vida de software. Su enfoque modular y flexible permite a los equipos de desarrollo definir y administrar el flujo de trabajo de forma coherente, independientemente de la complejidad del entorno. Además de su capacidad para facilitar la orquestación de aplicaciones, Kubernetes brinda herramientas para supervisar y garantizar la alta disponibilidad de los servicios, adaptándose a las fluctuantes demandas y permitiendo una adaptación ágil ante cambios repentinos en el tráfico [27].

Esta plataforma se fundamenta en un modelo que se vale de componentes fundamentales para gestionar las diversas actividades inherentes al ciclo de desarrollo de software. Su arquitectura modular, que comprende elementos como nodos, pods, servicios y controladores, define la base para un enfoque coherente en la creación y administración de aplicaciones en arquitecturas distribuidas. La adopción de Kubernetes, junto con la comprensión de sus conceptos centrales, se erige como un recurso clave para la implementación exitosa de sistemas distribuidos basados en microservicios o modelos serverless [28].

4.3.2.3. AWS ECS

ECS (Amazon Elastic Container Service) ofrece un robusto respaldo para contenedores Docker y Fargate, presentando su propio motor para la ejecución de contenedores. Este enfoque libera al usuario de tareas costosas como la selección de instancias, aprovisionamiento y la configuración de conjuntos de clústeres de máquinas virtuales [29].

Además de su enfoque simplificado en la gestión de contenedores, ECS se destaca por su alta escalabilidad. Esto otorga a los usuarios la capacidad de implantar y gestionar su propio software de orquestación de contenedores, ejercer control sobre la administración y la escalabilidad de clústeres y máquinas virtuales, y programar la ejecución precisa de contenedores en dichas entidades virtuales [30].

4.3.2.4. Azure Kubernetes Service

Azure Kubernetes Service (AKS) constituye una herramienta de gran robustez para la gestión de contenedores, posibilitando un escalado eficiente tanto de aplicaciones como de la infraestructura subyacente [31].

La característica distintiva de AKS radica en su capacidad para realizar el aprovisionamiento de clústeres de manera fluida y flexible. Este proceso puede llevarse a cabo a través de la propia plataforma de Azure, la interfaz de línea de comandos de Azure o mediante herramientas de infraestructura como código, como Azure Resource Manager y

Terraform. Esta diversidad de opciones otorga a los usuarios un mayor control sobre la configuración y despliegue de sus clústeres, a la vez que fomenta una adaptación cómoda a sus preferencias y requerimientos específicos [32].

4.3.2.5. DigitalOcean

DigitalOcean se erige como una destacada solución en el ámbito de la administración de infraestructura y despliegue de aplicaciones. A través de su plataforma, los usuarios pueden acceder a una serie de servicios y herramientas diseñados para simplificar y agilizar el proceso de gestión de entornos virtuales. Un aspecto central de la oferta de DigitalOcean es su servicio de máquinas virtuales, conocidas como "droplets", que permiten desplegar aplicaciones y servicios en entornos escalables y altamente configurables. Además de los droplets, la plataforma ofrece una variedad de soluciones complementarias, incluyendo servicios de almacenamiento, bases de datos y redes, que potencian la capacidad de los usuarios para construir y operar aplicaciones de manera efectiva [33].

Uno de los rasgos distintivos de DigitalOcean es su enfoque en la simplicidad y la usabilidad. A través de su interfaz intuitiva y amigable, los usuarios pueden configurar y gestionar sus recursos de manera eficiente, incluso si carecen de una experiencia técnica avanzada [34]. Además, DigitalOcean brinda la posibilidad de automatizar tareas a través de herramientas como Terraform y Ansible, lo que contribuye a una gestión más ágil y reproducible de la infraestructura. Con su enfoque centrado en el usuario y su gama de servicios versátiles, DigitalOcean emerge como una opción atractiva para quienes buscan una solución integral y accesible para sus necesidades de despliegue y gestión de aplicaciones en entornos de nube [35].

4.4. Trabajos relacionados

4.4.1. Menú de tableta Menulux

Menulux ofrece una aplicación de menú digital para tabletas que le permite crear y utilizar su menú en cuestión de minutos.

Proporciona logotipos y pantallas de acceso exclusivos para su negocio, así como fotos y descripciones bien elegidas de sus productos. También puede preparar diferentes menús para días especiales y festivos, así como organizar páginas de menú personalizadas para cada menú.

Soporte de idiomas extranjeros: Prepare contenidos en todos los idiomas que desee presentar en su menú. A continuación, cambie el idioma del menú con un solo toque, tocando

el icono de selección de idioma situado en la parte inferior izquierda de la pantalla de acceso al menú.

Actualizaciones automáticas de la aplicación: Modifica tus menús, las fotos de los productos, el precio y la descripción como quieras. El menú de la tableta Menulux actualiza automáticamente todo su contenido [36].

4.4.2. Menú de entrega y recogida en línea

Reduzca su dependencia de los socios del canal de entrega y empiece a aceptar entregas y recogidas directas en línea. Cree un menú de entrega que sea diferente del menú de su restaurante y conecte el enlace del menú de entrega a su sitio web y compártanlo en sus canales de redes sociales.

4.4.3. Eatsa

No hay cajeros ni camareros, no hay un solo humano, al menos a la vista. Al entrar a Eatsa se pueden ver unas estaciones con iPad para realizar los pedidos y una pared llena de compartimentos, como si fuesen hornos de microondas. Lo único que hay que hacer es pasar la tarjeta de crédito por la terminal del iPad, y por medio de una aplicación se puede elegir el platillo con una amplia variedad de opciones y con la posibilidad de personalizarlo.

Paso siguiente es esperar frente a la pared de puertas transparentes LCD, donde se puede observar animaciones hasta que la pantalla se ponga en negro y aparezca un plato con el nombre del cliente en pantalla. Eso es todo, no hay interacciones, no se puede ver cómo cocinan la comida, nadie cruza palabras, todo se hace de forma automática.

5. Metodología

En esta sección se muestra la metodología usada en el desarrollo del presente proyecto de trabajo de titulación. A continuación, encontrará, en la sección 5.1., el contexto donde se desarrolló el proyecto, sección 5.2., proceso bajo el cual se abordaron los objetivos planteados al inicio del proyecto, sección 5.3., recursos utilizados y en la sección 5.4., los participantes o involucrados en el presente.

5.1. Área de estudio

El presente trabajo de titulación se llevó a cabo en 5 cafeterías de la ciudad de Loja durante el período académico abril - septiembre 2022 contemplado por la Universidad Nacional de Loja.

5.2. Procedimiento

Los objetivos planteados se alcanzaron con la realización de las siguientes actividades:

5.2.1. Construir el modelo de arquitectura física y lógica en base a los requerimientos del usuario utilizando el Modelamiento de desarrollo de software dirigido por ICONIX.

- Análisis y diseño preliminar: se realizó el modelado de requerimientos (véase **sección 6.1.1.1 Modelado de requerimientos**) y el diagrama del dominio (véase **sección 6.1.1.2 Modelado del diagrama del dominio**).
- Diseño de la arquitectura física y lógica: se modeló casos de uso, la interacción mediante diagramas de secuencia, modelado estructural mediante diagrama de paquetes y diagrama de clases y la arquitectura física (véase **sección 6.1.2 Diseño de la arquitectura física y lógica**).

5.2.2. Implementar la arquitectura de la solución informática en una aplicación web mediante el uso de tecnologías JavaScript y servicios de la nube con la metodología ICONIX.

- Implementación: en esta etapa se desarrollaron los módulos de gestión de usuarios, menús y pedidos (véase **sección 6.2.2 Servicios** y **sección 6.2.3 Servicio FrontEnd**).
- Pruebas de funcionamiento e integración de componentes de software: con el fin de supervisar la correcta integración entre módulos y comportamiento parcial y

general del sistema se realizaron pruebas de funcionamiento e integración (véase **sección 6.2.4 Realizar pruebas de funcionamiento del software**).

5.2.3. Comparar los procesos de toma de pedidos tradicional y web en un ambiente de pruebas simulado dentro de la FEIRNR para determinar los beneficios de la actualización.

- Planificación del caso de prueba para la simulación manual versus web (véase **sección 6.3.1 Planificación del caso de prueba para la simulación manual versus web**).
- Ejecución de la simulación de los casos de prueba establecidos (véase **sección 6.3.2 Ejecución de la simulación**).
- Análisis cuantitativo de los resultados con el fin de medir las respuestas del sistema en un entorno de pruebas (véase **sección 6.3.3 Establecer un análisis cuantitativo de los resultados**).

5.3. Recursos

En la realización del presente trabajo de titulación, fueron necesarios los siguientes recursos:

5.3.1. Recursos Científicos

5.3.1.1. Estudios de caso

Se analizó el proceso seguido por varias cafeterías dentro de la ciudad de Loja con el fin de determinar el comportamiento general, similitudes y diferencias entre una y otra en atención al cliente, menús y difusión.

5.3.1.2. Investigación Bibliográfica

Revisión de casos similares en un contexto más amplio, así como revisión de soluciones existentes (véase **sección 4.4 Trabajos relacionados**).

5.3.1.3. Observación directa

Se visitó las cafeterías, se tomó nota de cambios en menús que fueron evidentes en sus medios de comunicación y/o medios digitales, así como, también, se identificó diferencias en los comportamientos de los grupos consumidores (véase **Anexo 2. Menús de Cafeterías**).

5.3.2. Recursos Técnicos

5.3.2.1. Entrevistas

Estas permitieron adquirir información destacada sobre el funcionamiento de las cafeterías, identificación de procesos clave y una visión más profunda de su gestión (véase **Anexo 1. Entrevistas para la obtención de requerimientos**).

5.3.2.2. Encuestas

Aplicadas a clientes de las cafeterías observadas, con el fin de identificar sus necesidades y posibles puntos de mejora (véase **Anexo 6. Resultados de la Encuesta de Satisfacción**).

5.3.3. Recursos de Hardware y Software

5.3.3.1. Hardware

- **Laptop:** equipo que fue necesario para las distintas investigaciones que requería este proyecto, así como para cumplir con el desarrollo de solución informática.

5.3.3.2. Software

- **Mendeley Reference Manager:** programa informático que permitió la gestión de las fuentes bibliográficas.
- **Visual Studio Code:** editor de código utilizado para la implementación de la Aplicación Web Progresiva.
- **ClickUp:** software utilizado para la gestión de tareas y actividades del proyecto, permitió el seguimiento de los objetivos y la planificación de horarios.
- **MERN Stack:** conjunto de software de JavaScript compuesto por *MongoDB*, *Express.js*, *React* y *Node.js* que permitió construir la solución informática.

5.4. Participantes

El trabajo de titulación se realizó por la estudiante Maria Anabel Encalada Córdova bajo dirección del ing. Wilman Chaman, docente de la Universidad Nacional de Loja. Además, con la participación de los encargados de las cafeterías observadas.

6. Resultados

Con el fin de exponer la evidencia de los resultados obtenidos del desarrollo de los objetivos del presente Trabajo de Titulación se presentan los siguientes resultados:

6.1. Objetivo 1: Construir el modelo de arquitectura física y lógica en base a los requerimientos del usuario utilizando el Modelamiento de desarrollo de software dirigido por ICONIX

Para dar cumplimiento al primer objetivo del presente TT, se llevaron a cabo 2 iteraciones con las siguientes actividades: Modelado de requerimientos y diagrama conceptual, documentación mínima en el marco de la metodología ICONIX. Los resultados obtenidos de este proceso se presentan a continuación:

6.1.1. Análisis y diseño preliminar

En esta fase se realizó una entrevista de entendimiento y conocimiento general a 5 propietarios de cafeterías dentro de la ciudad de Loja, estas fueron aplicadas dentro de las siguientes cafeterías: Artesanale, Cafetería universitaria de la carrera de turismo de la Universidad Nacional de Loja, El Emporio Lojano, El rincón de Borgoña y La llorona, mismas que se encuentran en el **Anexo 1. Entrevistas para la obtención de requerimientos**. En base a estas entrevistas se logró un entendimiento del modelo de negocio y relación entre personal de cafetería y clientes con respecto a la pandemia del COVID-19. De estas entrevistas se identificaron conceptos claves en el área productiva y los primeros requerimientos.

Para efectos de más detalles del menú, platos, bebidas, órdenes y proceso de pago se realizó observación directa sobre las cartas físicas que disponen las cafeterías mencionadas y a los procesos ejecutados por los miembros de los locales mientras atendían a clientes, los menús de las cafeterías observadas se encuentran en el **Anexo 2. Menús de Cafeterías**. Con la información obtenida, se definieron más detalles importantes para el modelado de requerimientos y el diagrama conceptual.

6.1.1.1. Modelado de requerimientos

Con la información obtenida de entrevistas y visitas a las cafeterías observadas, se presenta el modelado de requerimientos (véase **Anexo 3. Modelado de la arquitectura lógica** Anexo 3. Modelado de), compuesto de requerimientos funcionales y no funcionales.

A continuación, se muestran los requerimientos funcionales obtenidos para la construcción de la solución informática.

El sistema permitirá al administrador:

Tabla 2. Requerimientos funcionales

Código de requerimiento	Descripción	Categoría
RF05	Registrar cuentas de usuario. Los datos a registrar son: nombre, apellido, cédula, correo electrónico y clave.	Visible
RF06	Registrar roles a las cuentas de usuario, estos serán: Administrador, Cocinero, Mesero y Cliente.	Visible
RF07	Actualizar los roles de usuario.	Visible
RF08	Buscar cuentas de usuario por la dirección de correo electrónico o cédula.	Inferido
RF09	Dar de baja cuentas de usuario	Visible

El sistema permitirá al cliente:

Tabla 3. Requerimientos funcionales

Código de requerimiento	Descripción	Categoría
RF10	Registrar su cuenta de usuario. Los datos obligatorios a registrar son: nombre, apellido, correo electrónico y clave, mientras que opcionales: cédula, dirección y número de celular.	Visible
RF11	Visualizar el menú disponible.	Visible
RF12	Elegir un plato del menú y ver los detalles del producto, estos son: denominación, precio y descripción (opcional).	Visible
RF13	Seleccionar platos y/o bebidas del menú	Visible
RF14	Quitar productos de su pedido, antes de que este sea confirmado.	Visible
RF15	Y al mesero generar un pedido con los platos seleccionados y el número de mesa.	Visible
RF16	Ver el detalle de su pedido, así como el valor a pagar por el consumo.	Visible
RF17	Dar por terminado su consumo, cuando esto haya pasado, el sistema debe notificar al cajero para que haga el cobro.	Inferido

El sistema permitirá al cocinero:

Tabla 4. Requerimientos funcionales

Código de requerimiento	Descripción	Categoría
RF18	Visualizar los pedidos pendientes de preparar, realizados por los clientes	Visible
RF19	Actualizar el estado de los pedidos a despachado cuando este ha sido terminado	Inferido

El sistema permitirá al mesero:

Tabla 5. Requerimientos funcionales

Código de requerimiento	Descripción	Categoría
RF20	Registrar categorías al menú, los datos a registrar son: nombre, descripción.	Visible
RF21	Actualizar el nombre de las categorías del menú	Inferido
RF22	Buscar categorías por denominación.	Inferido
RF23	Dar de baja categorías	Visible
RF24	Registrar platos o bebidas en el menú, los datos a guardar son: nombre, descripción (opcional) y precio.	Visible
RF25	Actualizar la información de los productos.	Visible
RF26	Buscar productos por nombre.	Inferido
RF27	Dar de baja productos.	Visible
RF28	Visualizar el detalle los pedidos que han sido despachados a través del número de mesa o nombre	Visible
RF29	Calcular el valor a pagar, con impuestos, de pedidos en estado despachado	oculto
RF30	Generar un comprobante de pago en base al detalle de pedidos despachados	Visible

6.1.1.2. Modelado del diagrama del dominio

Una vez hecho el levantamiento de información y escritos los requerimientos, en la Tabla 6 se presenta el glosario de términos.

Tabla 6. Glosario de términos

Término	Concepto
Administrador	Rol encargado de la gestión de usuarios y sus roles (mesero, cocinero y administrador)
Cliente	Actor que navega por el menú de la cafetería, realiza pedidos y gestiona su propia cuenta de usuario.
Mesero	Actor encargado de asistir al cliente en la realización de los pedidos, pagos por consumo, así como de actualizar menús y productos.
Cocinero	Actor encargado de procesar los pedidos.
Menú	Lista de categorías con productos asociados. Por ejemplo, bebidas calientes es la categoría que tiene los productos café y té.
Producto	Son los platos y bebidas que están disponibles en los menús de la cafetería.
Categoría	Es un agrupador de productos que comparten ciertas características en común que los hace ser parte del grupo.
Pedido	Es la orden de compra que lanza un cliente.
Comprobante de venta	Es un registro que contiene los detalles de consumo de un cliente y su información si así el visitante lo desea o puede generarse con datos por defecto. No tiene valor tributario.
Pago	Es el registro que da constancia del pago efectuado por parte del cliente por concepto de consumo.

Con el glosario de términos presentado en la

Tabla 6 y el levantamiento de requerimientos funcionales, se presenta el modelo del dominio dividido en 2 partes: Figura 1 contiene el modelado conceptual de seguridad que contiene el detalle del usuario, roles y privilegios, y en la Figura 2 se muestra el modelado del negocio con los flujos esperados. De color verde se detalla la lista de roles, hasta el corte de observación y alcance del proyecto.

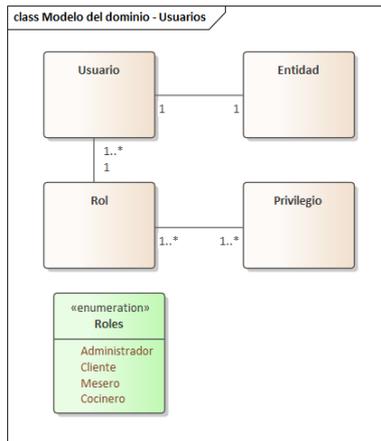


Figura 1. Modelado del dominio – Usuarios

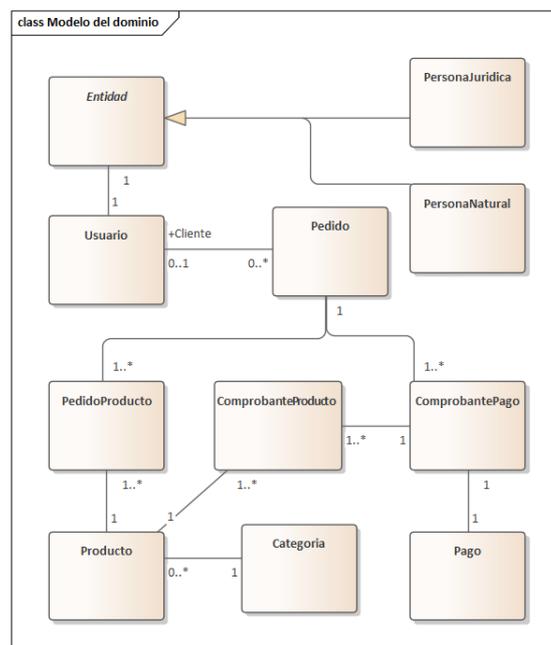


Figura 2. Modelado del dominio – negocio

En la Figura 2 se observa la relación de conceptos en los flujos propios del modelo de negocios. Así como la relación entre cada objeto.

6.1.2. Diseño de la arquitectura física y lógica

Para una mejor definición del flujo de acciones y comportamientos tanto de actores como de conceptos, se presentan varios diagramas donde se detallan flujos de secuencia de acciones y procesos. Adicional y después de dos iteraciones, se definieron 3 servicios (usuarios, productos y pedidos) que se irán presentando desde la vista de integración como monolito y desde el diseño propio e individual de cada uno de estos con su naturaleza de microservicios. En las próximas páginas se muestran los diagramas que forman parte de la arquitectura lógica, para más detalles ver en el **Anexo 3. Modelado de la arquitectura lógica.**

6.1.2.1. Modelado de casos de uso

Con la definición anterior de conceptos y actores, se desarrollan diagramas de casos de uso que representan con que entidades se relaciona cada rol, así como las actividades que puede o no hacer.

En la Figura 3 se presenta el diagrama de caso de uno general.

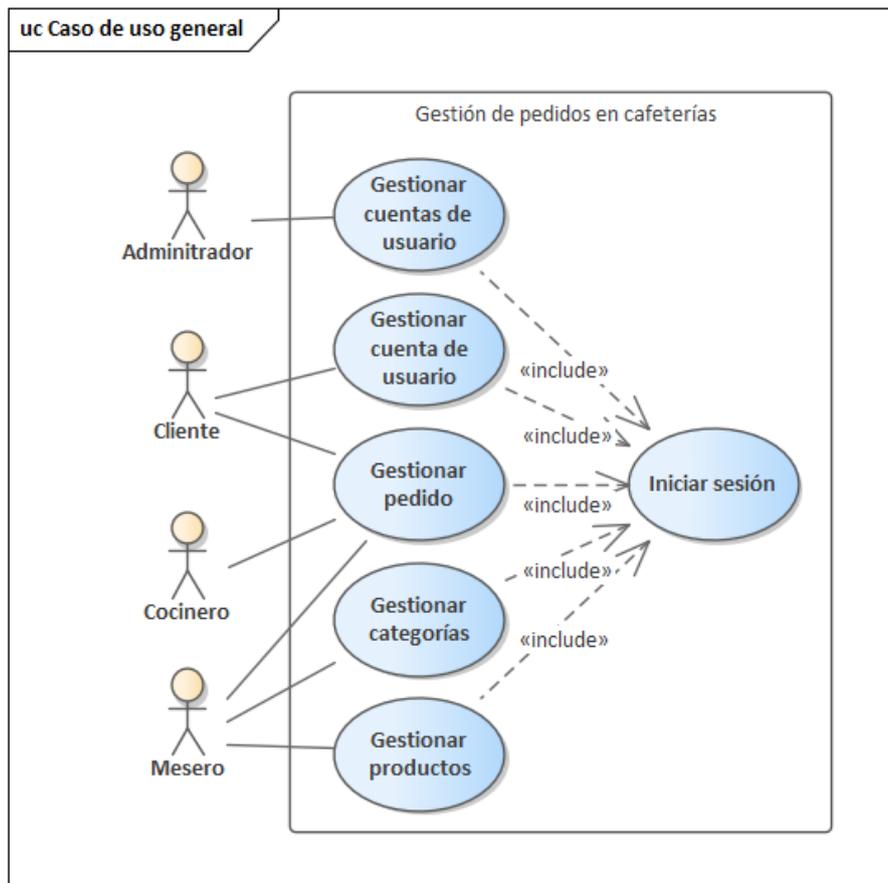


Figura 3. Diagrama de casos de uso - General

En la Tabla 7 se presenta la descripción de caso de uso iniciar sesión, que aplica a todos los usuarios y para todas las acciones, con la excepción del registro de cuentas de cliente y ver menú, en esos casos no será necesario haber accedido al sistema con credenciales de usuario.

Tabla 7. Descripción de caso de uso - Iniciar sesión

Identificador	CU01
Nombre	Iniciar sesión
Actores	Todos los usuarios del sistema.
Descripción	Autenticación del usuario en el sistema.
Precondiciones	Tener cuenta registrada en el sistema.

	Estar en la página "Iniciar sesión".
Flujo Principal	<p>El usuario ingresa sus credenciales de acceso (correo y clave).</p> <p>El usuario pulsa el botón "Iniciar sesión".</p> <p>El sistema verifica la información ingresada en el formulario.</p> <p>El sistema autentica al usuario y redirige a la página principal con acceso a las funciones autorizadas a su rol.</p> <p>El caso de uso finaliza.</p>
Flujo Alternativo	<p>Campos obligatorios vacíos</p> <p>4.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.</p> <p>Credenciales de acceso no válidas</p> <p>5.1. El sistema muestra un mensaje indicando que las credenciales no son correctas.</p>

Servicio de usuarios

Los diagramas de caso de uso detallados o explotados se presentan a continuación. En la Figura 4, el caso de uso Gestionar cuentas de usuarios, se ven las acciones que realizaría un administrador.

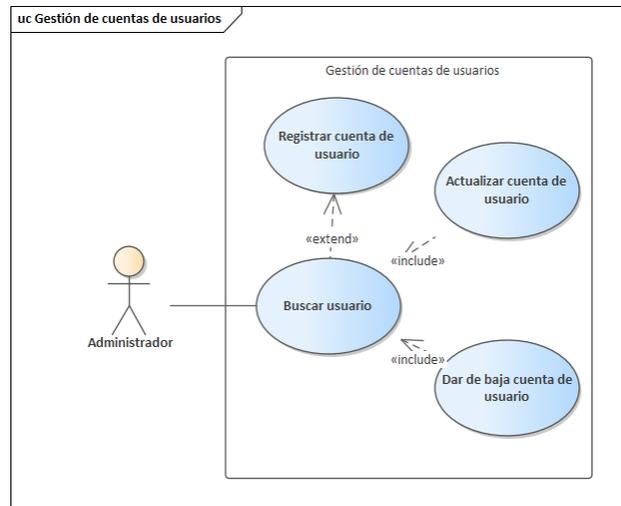


Figura 4. Diagrama de casos de uso - Gestión de cuentas de usuarios

Tabla 8. Descripción de caso de uso - Registrar cuentas de usuario

Identificador	CU02
Nombre	Registrar cuentas de usuario
Actor	Administrador
Descripción	Registro de nuevas cuentas de usuario
Precondición	Iniciar sesión

	Encontrarse en la página Listar usuarios
Flujo normal	<p>El administrador pulsa el botón "Registrar usuario".</p> <p>El sistema muestra un formulario registrar usuario.</p> <p>El administrador ingresa la información del usuario (nombre, apellido, dirección de correo electrónico)</p> <p>El administrador asigna un rol</p> <p>El administrador pulsa el botón "Guardar".</p> <p>El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.</p> <p>El sistema verifica que la dirección de correo electrónico no esté registrada.</p> <p>El sistema guarda los cambios y muestra un mensaje indicando que el usuario se ha registrado con éxito.</p>
Flujo alterno	<p>Campos obligatorios vacíos</p> <p>6.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.</p> <p>Validación de cédula incorrecta</p> <p>6.1. El sistema muestra un mensaje de alerta indicando que el número de cédula de la entidad es incorrecto cuando no es de tipo pasarte</p> <p>Formato de correo incorrecto</p> <p>6.1. El sistema muestra un mensaje indicando que la dirección de correo electrónico no cumple el formato.</p> <p>Registro existente</p> <p>7.1. El sistema muestra un mensaje indicando que hay un usuario registrado con la dirección de correo electrónico ingresada.</p>

Los casos de uso descritos correspondientes a la Figura 5, responden a la acción que realiza el cliente de gestión de su propia cuenta, mientras que lo descrito con referencia a los casos de uso de la Figura 4, son las acciones del administrador con la gestión de cuentas de todos los roles.

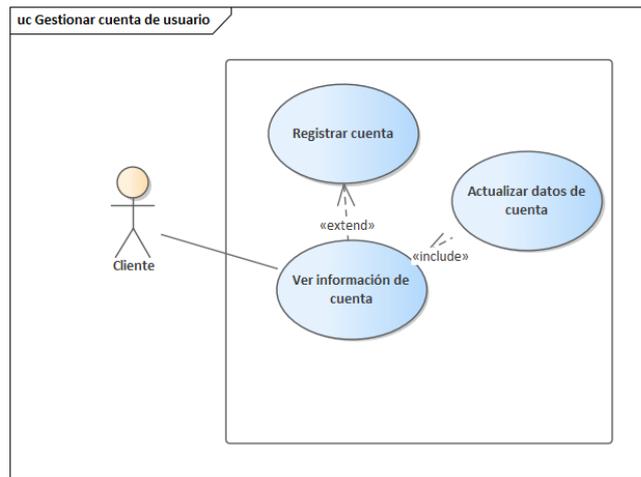


Figura 5. Diagrama de casos de uso - Gestión de cuenta de usuario

Tabla 9. Descripción de caso de uso - Registrar cuenta de usuario

Identificador	CU06
Nombre	Registrar cuenta de usuario
Actor	Cliente
Descripción	Registro de cuenta de usuario con rol cliente
Precondición	Encontrarse en la página Registrar cuenta
Flujo normal	<p>El cliente ingresa la información del usuario (nombre, apellido, dirección de correo electrónico).</p> <p>El cliente pulsa el botón "Guardar".</p> <p>El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.</p> <p>El sistema verifica que la dirección de correo electrónico no esté registrada.</p> <p>El sistema guarda los cambios y muestra un mensaje indicando que el usuario se ha registrado con éxito.</p>
Flujo alterno	<p>Campos obligatorios vacíos</p> <p>4.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.</p> <p>Validación de cédula incorrecta</p> <p>4.1. El sistema muestra un mensaje de alerta indicando que el número de cédula de la entidad es incorrecto cuando no es de tipo pasarte</p> <p>Formato de correo incorrecto</p> <p>4.1. El sistema muestra un mensaje indicando que la dirección de correo electrónico no cumple el formato.</p> <p>Registro existente</p> <p>5.1. El sistema muestra un mensaje indicando que hay un usuario registrado con la dirección de correo electrónico ingresada.</p>

Servicio de Productos

Los diagramas de casos de uso explotados correspondientes al servicio de Productos se pueden ver en la Figura 6 y Figura 7 respectivamente.

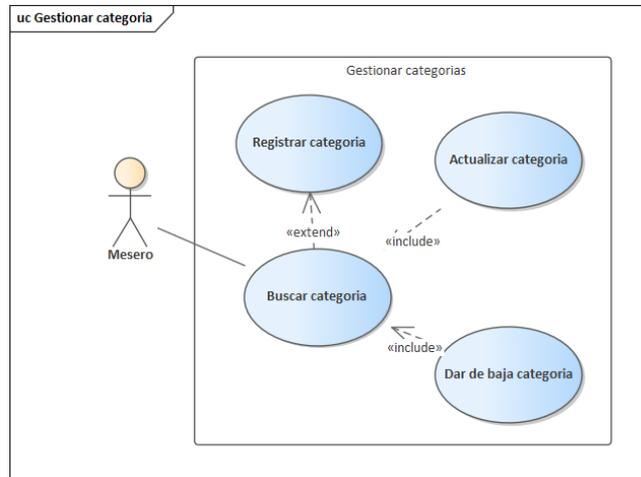


Figura 6. Diagrama de casos de uso - Gestión de categorías

Tabla 10. Descripción de caso de uso - Registrar Categoría

Identificador	CU09
Nombre	Registrar categoría
Actor	Mesero
Descripción	Registro de nuevas categorías
Precondición	Iniciar sesión Encontrarse en la página Listar categorías
Flujo normal	El mesero pulsa el botón "Registrar categoría". El sistema muestra un formulario registrar categoría. El mesero ingresa la información de la categoría (nombre y descripción) El mesero pulsa el botón "Guardar". El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. El sistema guarda la categoría muestra un mensaje indicando que el registro ha sido exitoso.
Flujo alterno	Campos obligatorios vacíos 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.

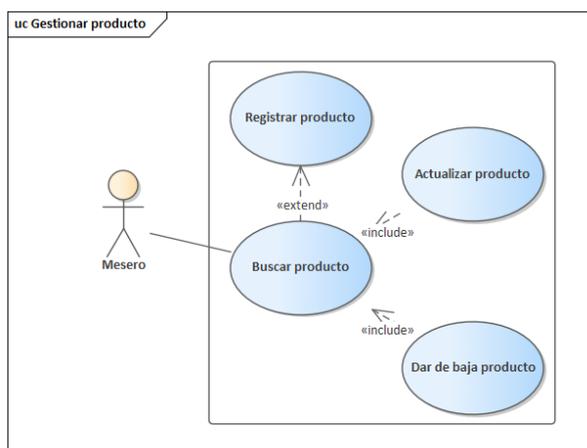


Figura 7. Diagrama de casos de uso - Gestión de productos

Tabla 11. Descripción de caso de uso - Registrar producto

Identificador	CU13
Nombre	Registrar producto
Actor	Mesero
Descripción	Registro de nuevos productos
Precondición	Iniciar sesión Encontrarse en la página Listar productos
Flujo normal	El mesero pulsa el botón “Registrar producto”. El sistema muestra un formulario de registro de producto. El mesero ingresa la información (nombre, descripción y precio). El mesero pulsa el botón “Guardar”. El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. El sistema guarda el producto y muestra un mensaje indicando que el registro ha sido exitoso.
Flujo alternativo	Campos obligatorios vacíos 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.

Servicio de pedidos

Este servicio cuenta de una acción general que involucra a varios actores. Las acciones y la interacción entre usuarios se presentan en la Figura 8.

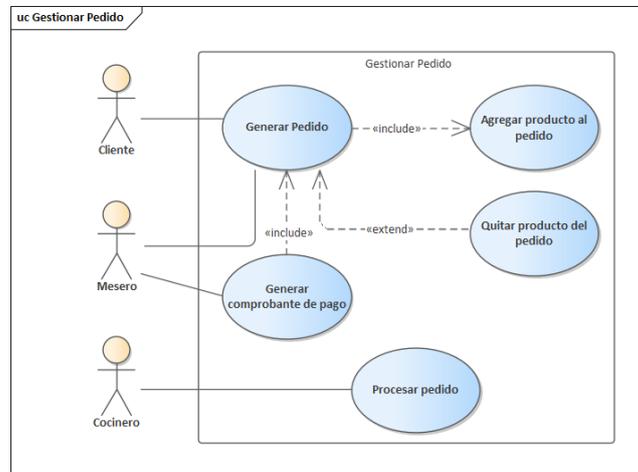


Figura 8. Diagrama de casos de uso - Gestión de pedido

Tabla 12. Descripción de caso de uso - Agregar productos al pedido

Identificador	CU17
Nombre	Agregar productos al pedido
Actor	Mesero o cliente
Descripción	Agregar productos a la lista de un pedido
Precondición	Haberse cumplido el caso de uso Iniciar sesión Encontrarse en la página pública del menú
Flujo normal	El usuario pulsa el botón "Agregar a pedido". El sistema muestra la opción de elegir la cantidad de productos. El usuario presiona el botón "Agregar" El sistema agrega a la lista del pedido y muestra un mensaje indicando que la operación se realizó con éxito.
Flujo alterno	

Tabla 13. Descripción de caso de uso - Generar comprobante de pago

Identificador	CU21
Nombre	Generar comprobante de pago
Actor	Mesero
Descripción	El mesero cobra por los productos consumidos
Precondición	Haberse cumplido el caso de uso Iniciar sesión Tener un pedido con estado procesado.

Flujo normal	<p>El mesero pulsa el botón “Generar comprobante de pago”.</p> <p>El sistema muestra un formulario para registrar datos del cliente, del pago y el detalle de consumo.</p> <p>El mesero completa el formulario.</p> <p>El mesero, una vez realizado y verificado el pago, pulsa el botón “Guardar”.</p> <p>El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.</p> <p>El sistema guarda el comprobante de pago y el pago.</p> <p>El sistema envía el comprobante de pago vía correo electrónico.</p> <p>El sistema actualiza el estado del pedido a “Terminado” y muestra un mensaje indicando que la operación se ha realizado con éxito.</p>
Flujo alterno	<p>Consumidor final</p> <p>2.1. El cliente puede decidir que en su comprobante no salgan sus datos, sino información por defecto.</p> <p>7.1. El sistema no envía comprobante de pago.</p> <p>Datos diferentes a los del registro de usuario (de existir)</p> <p>3.1. El cliente solicita al mesero que se ingresen otros datos para el comprobante de pago.</p> <p>3.2. El mesero registra los datos indicados por el cliente.</p> <p>Campos obligatorios vacíos</p> <p>6.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.</p>

6.1.2.2. Modelado de interacción mediante diagrama de secuencia

Una vez descrito el comportamiento de los actores con relación a los casos de uso, se presentan diagramas de secuencia que detallan la interacción entre las diferentes capas del sistema. Cada diagrama, está compuesto por el flujo normal que debería seguir y en texto de color rojo, se presenta el flujo alterno. La documentación de los diagramas, está dividida por los servicios previamente mencionados.

Servicio de usuarios

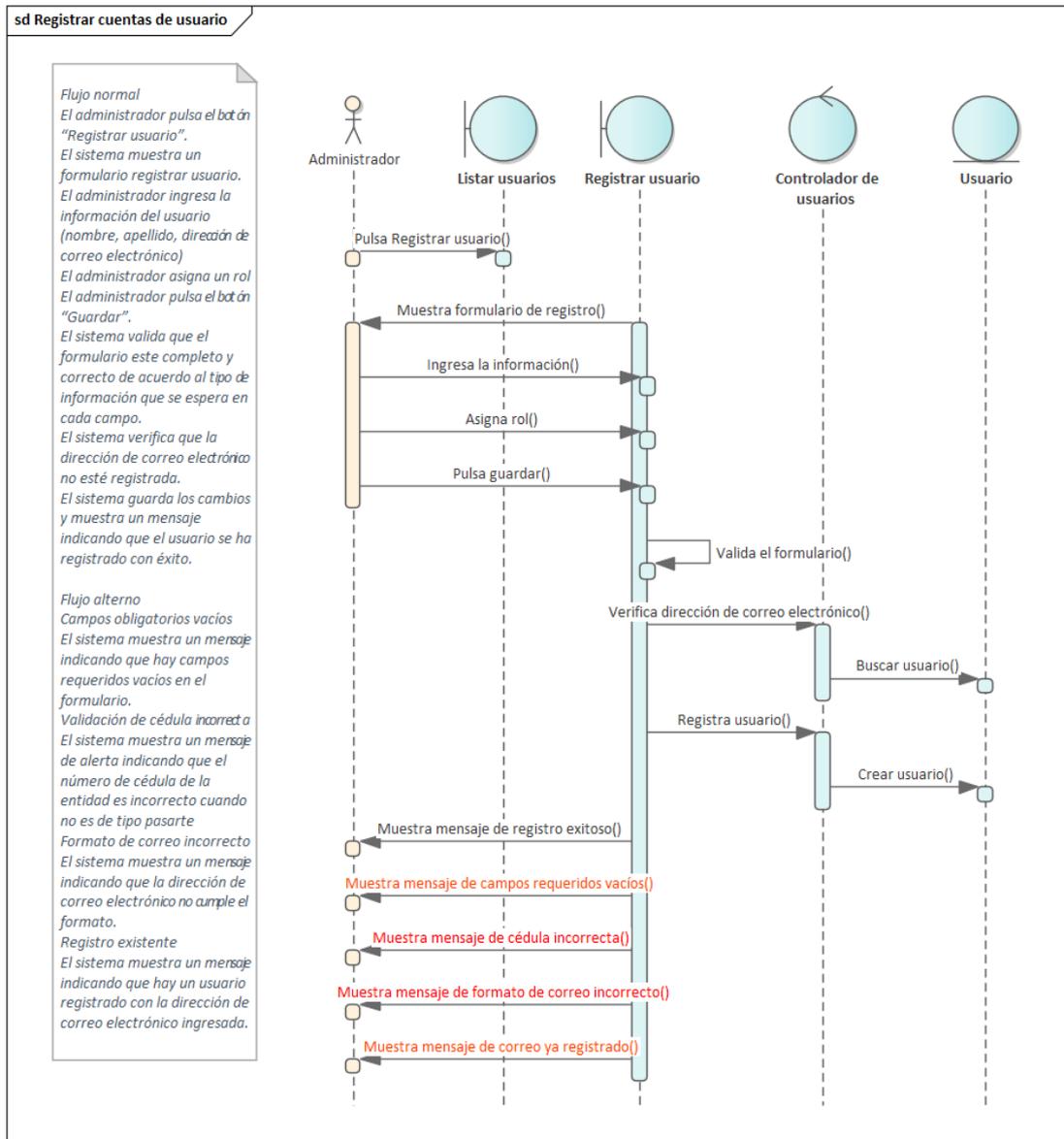


Figura 9. Diagramas de secuencia - Registrar cuentas de usuario

sd Registrar cuenta cliente

Flujo normal
 El sistema muestra un formulario registrar usuario.
 El cliente ingresa la información del usuario (nombre, apellido, dirección de correo electrónico).
 El cliente pulsa el botón "Guardar".
 El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.
 El sistema verifica que la dirección de correo electrónico no esté registrada.
 El sistema guarda los cambios y muestra un mensaje indicando que el usuario se ha registrado con éxito.

Flujo alterno
 Campos obligatorios vacíos
 4.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.
 Validación de cédula incorrecta
 4.1. El sistema muestra un mensaje de alerta indicando que el número de cédula de la entidad es incorrecto cuando no es de tipo pasarte
 Formato de correo incorrecto
 4.1. El sistema muestra un mensaje indicando que la dirección de correo electrónico no cumple el formato.
 Registro existente
 5.1. El sistema muestra un mensaje indicando que hay un usuario registrado con la dirección de correo electrónico ingresada.

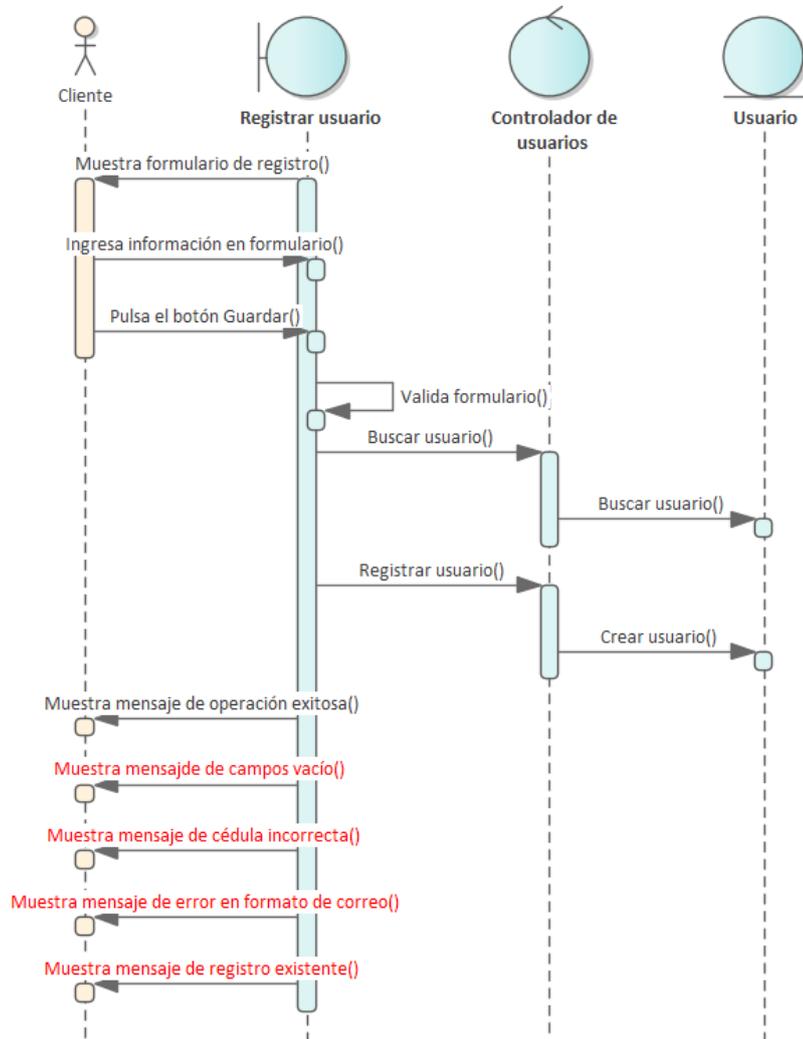


Figura 10. Descripción de caso de uso - Generar comprobante de pago

Servicio de productos

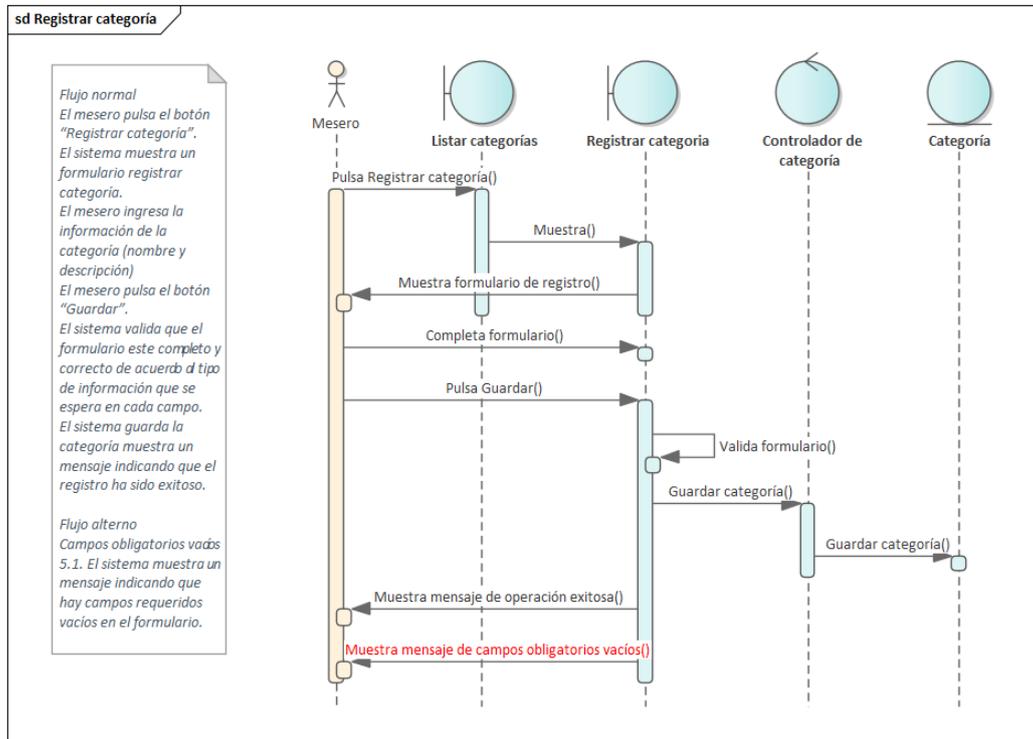


Figura 11. Diagramas de secuencia - Registrar categoría

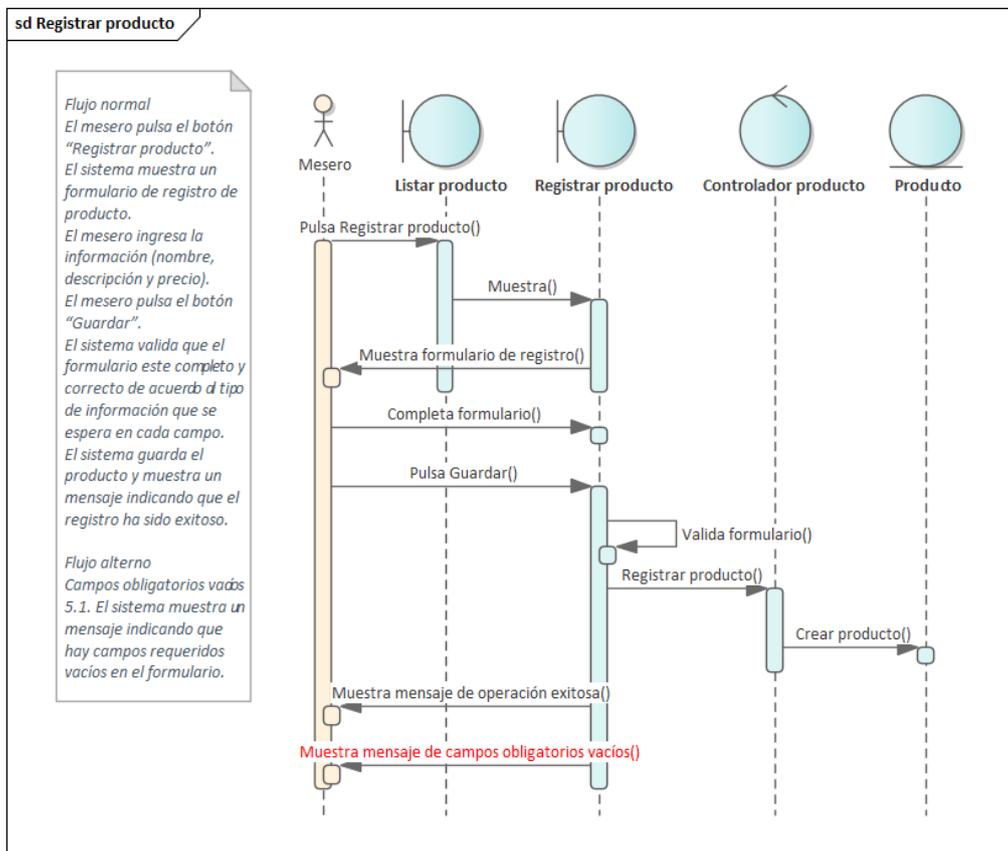


Figura 12. Diagramas de secuencia - Registrar producto

Servicio de pedidos

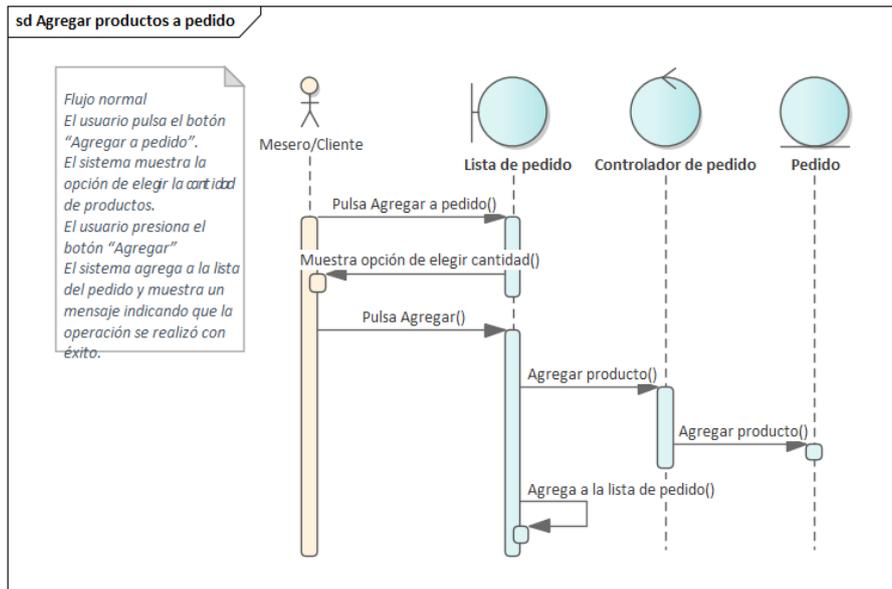


Figura 13. Diagramas de secuencia - Agregar producto al pedido

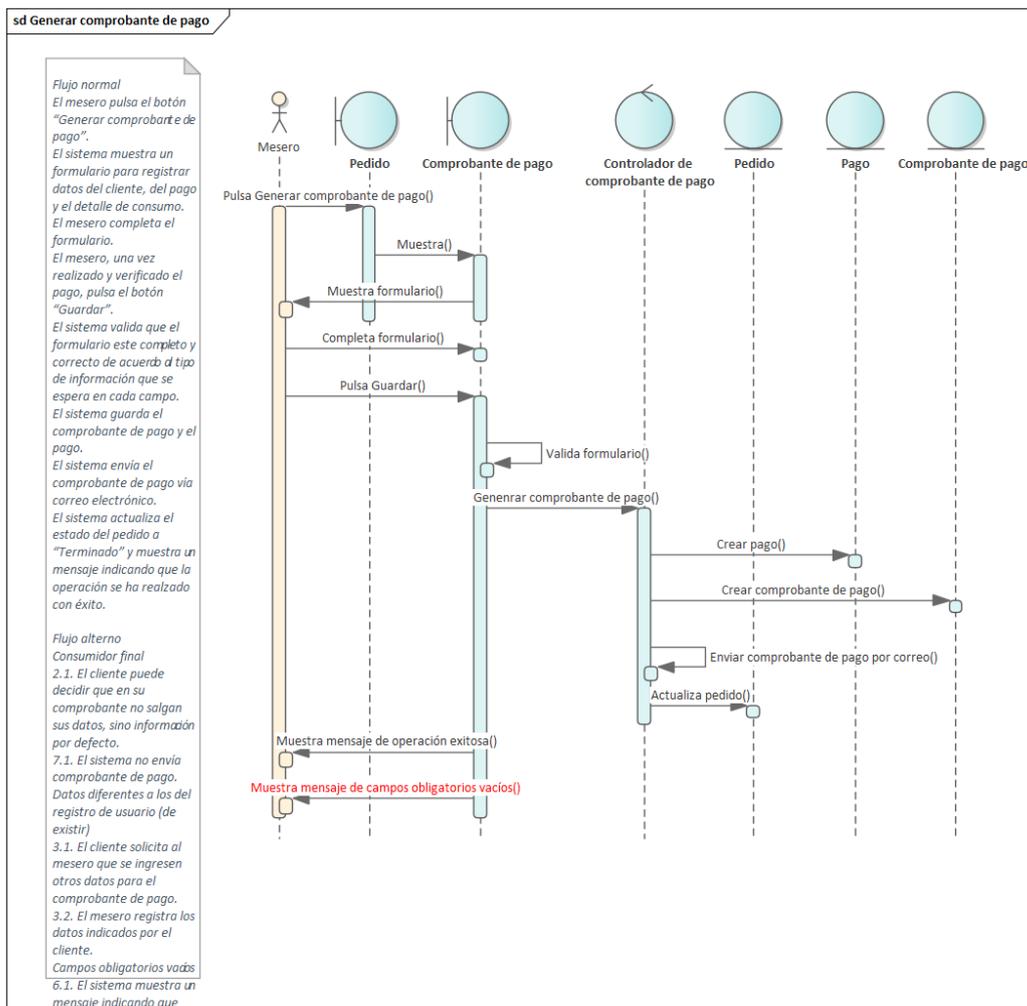


Figura 14. Diagramas de secuencia - Generar comprobante de pago

6.1.2.3. Modelado estructural mediante diagrama de paquetes y diagrama de clases

En la Figura 15. Diagrama de paquetes de microservicios se presenta el diagrama de paquetes que alberga los servicios identificados. Para cada microservicio se proporciona un detalle de los modelos que lo componen; asimismo, se incluye el paquete que engloba la construcción del FrontEnd. Cada servicio responde a un estilo arquitectónico basado en capas.

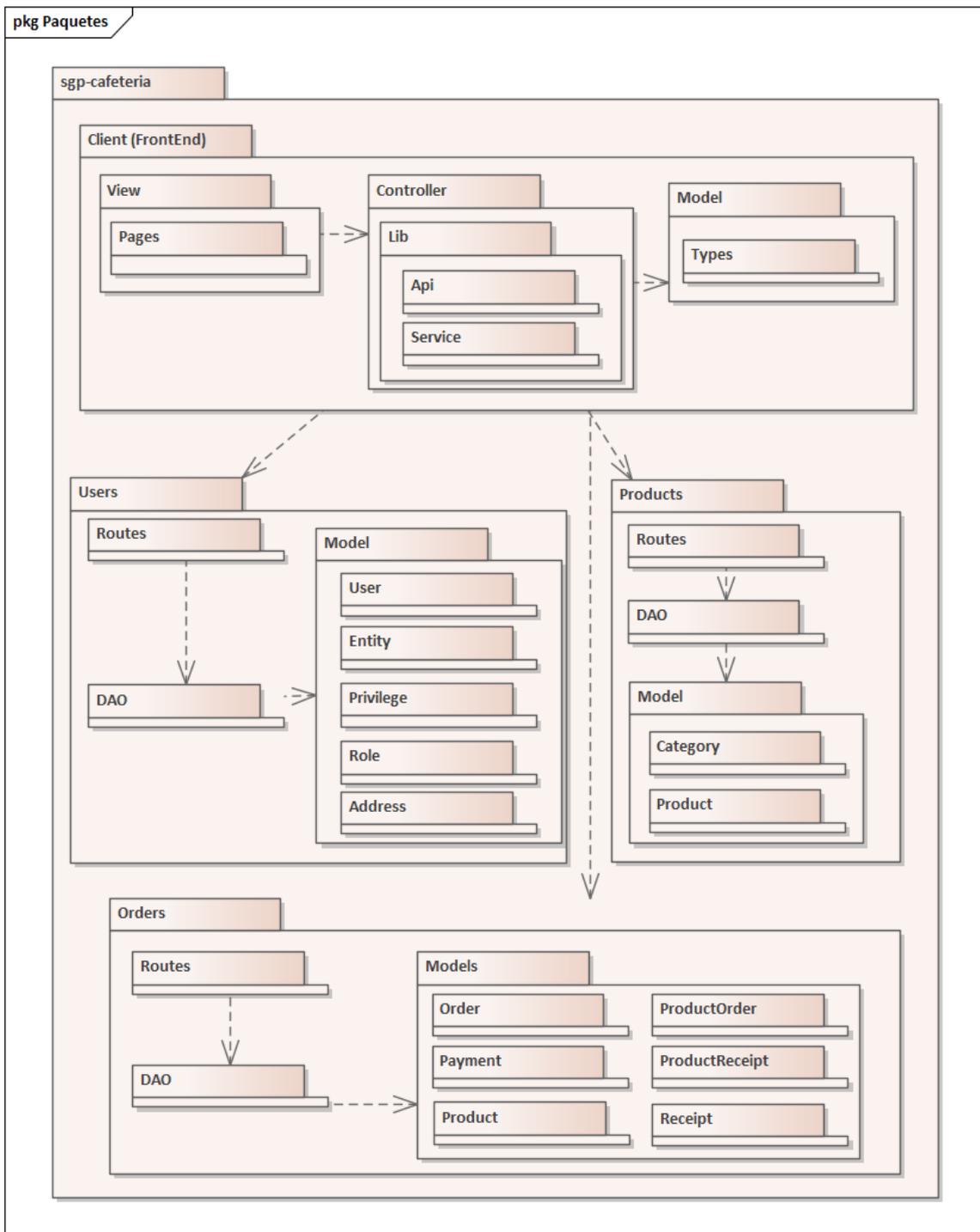


Figura 15. Diagrama de paquetes de microservicios

En la Figura 16. Modelo estático final del dominio se ve el modelo estático final del dominio o diagrama de clases basado en la representación del diagrama de conceptos (Ver Figura 1 y Figura 2), además incluye la identificación de atributos y operaciones.

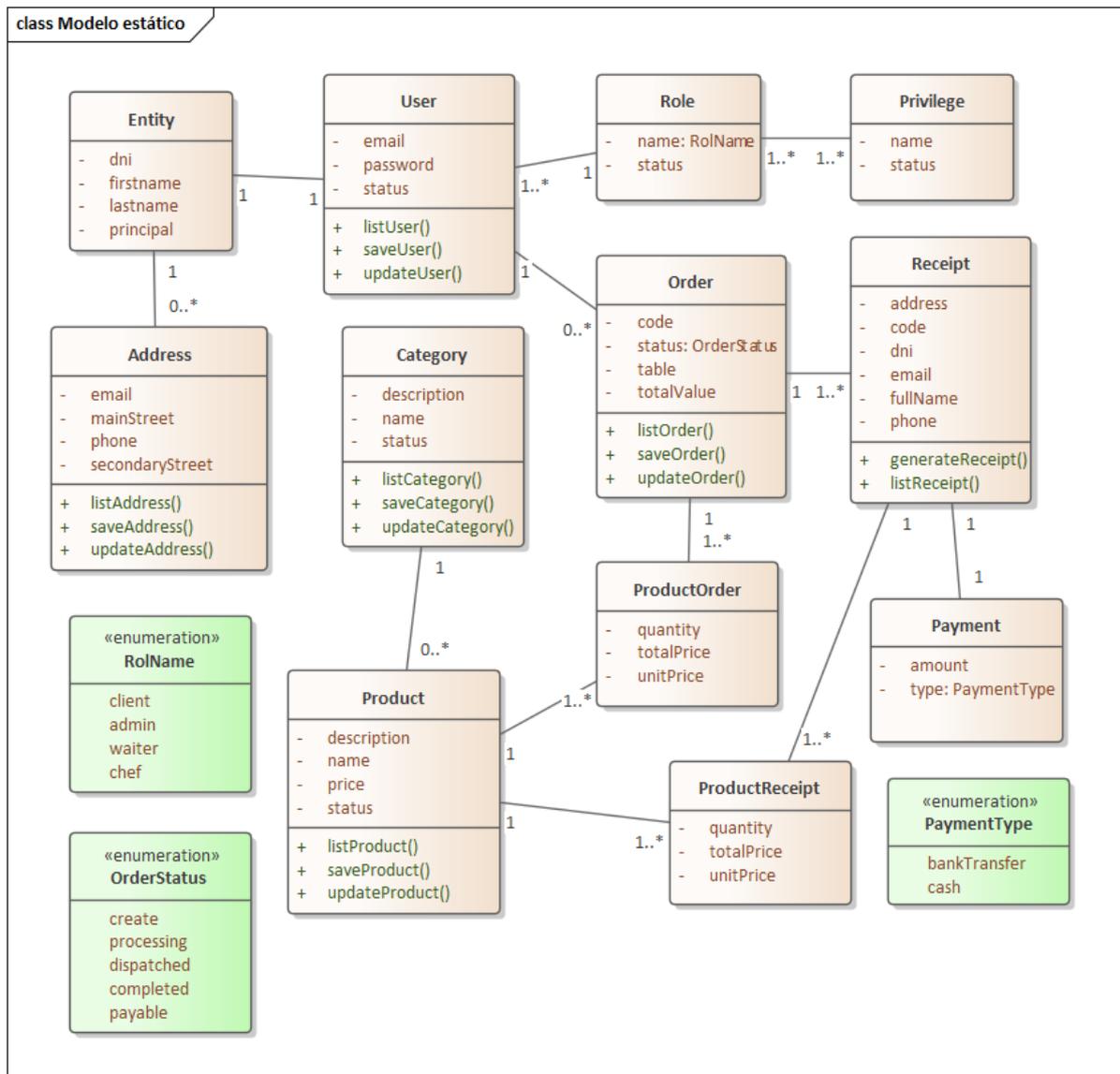


Figura 16. Modelo estático final del dominio

En la figura anterior se observa la integración y comunicación de todos los conceptos como si se tratara de un modelo tradicional monolítico. Las enumeraciones, conocidas como clases de tipo Enum, se han empleado con el propósito de establecer un conjunto discreto de valores. En este contexto, el nombre asignado a los roles se determina mediante la enumeración "RoleName".

El modelo estático correspondiente a cada servicio, se puede ver en las figuras a continuación:

Servicio de usuarios

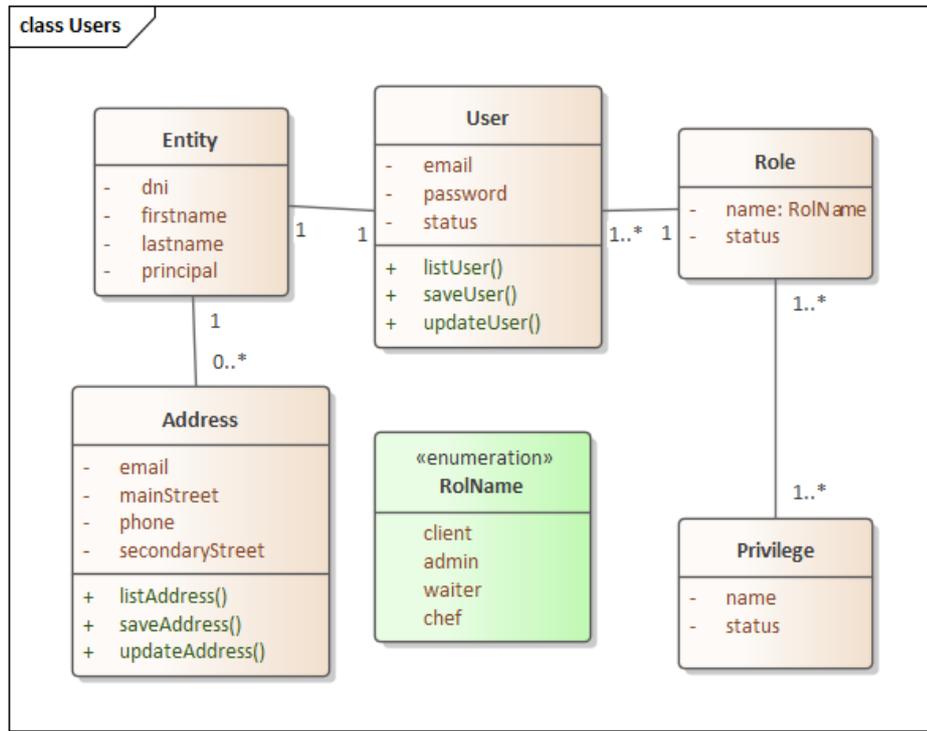


Figura 17. Modelo estático final de usuarios

Servicio de productos

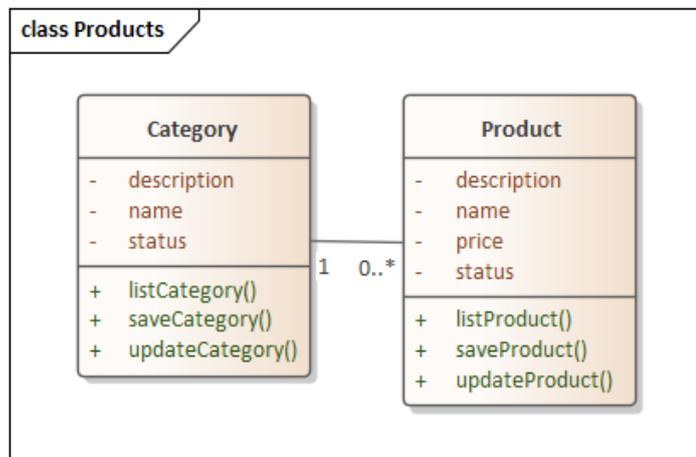


Figura 18. Modelo estático final de productos

Servicio de pedidos

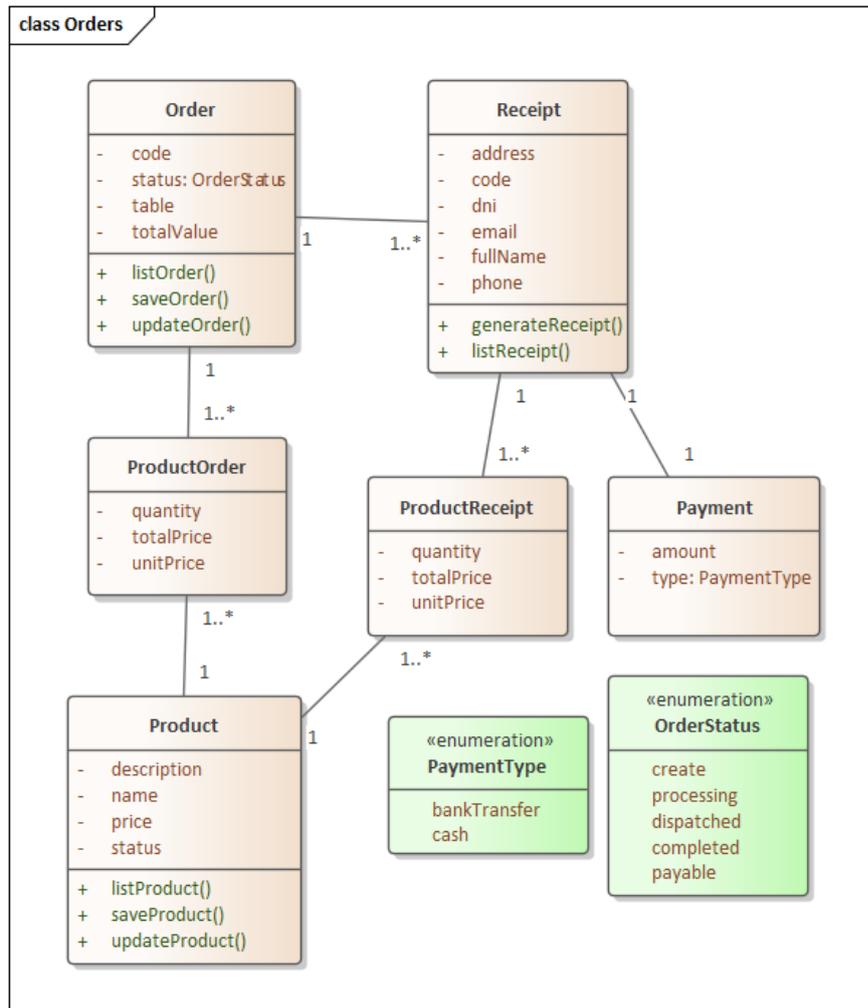


Figura 19. Modelo estático final de pedidos

En relación al modelo estático general de la solución (ver Figura 16. Modelo estático final del dominio) y a los diagramas de productos y pedidos (ver Figura 18. Modelo estático final de productos y Figura 19. Modelo estático final de pedidos), se observa que la clase Producto está presente en ambos. Este diseño evidencia la necesidad de establecer una comunicación efectiva entre estos dos servicios y la transmisión de datos entre ellos se gestiona a través de un bus de eventos.

6.1.2.4. Diseño de Arquitectura física

Las tecnologías con las que fueron construidas tanto los servicios como el consumo del API Rest en la aplicación web se puede ver presentada en la Figura 20, mediante un diagrama de componentes.

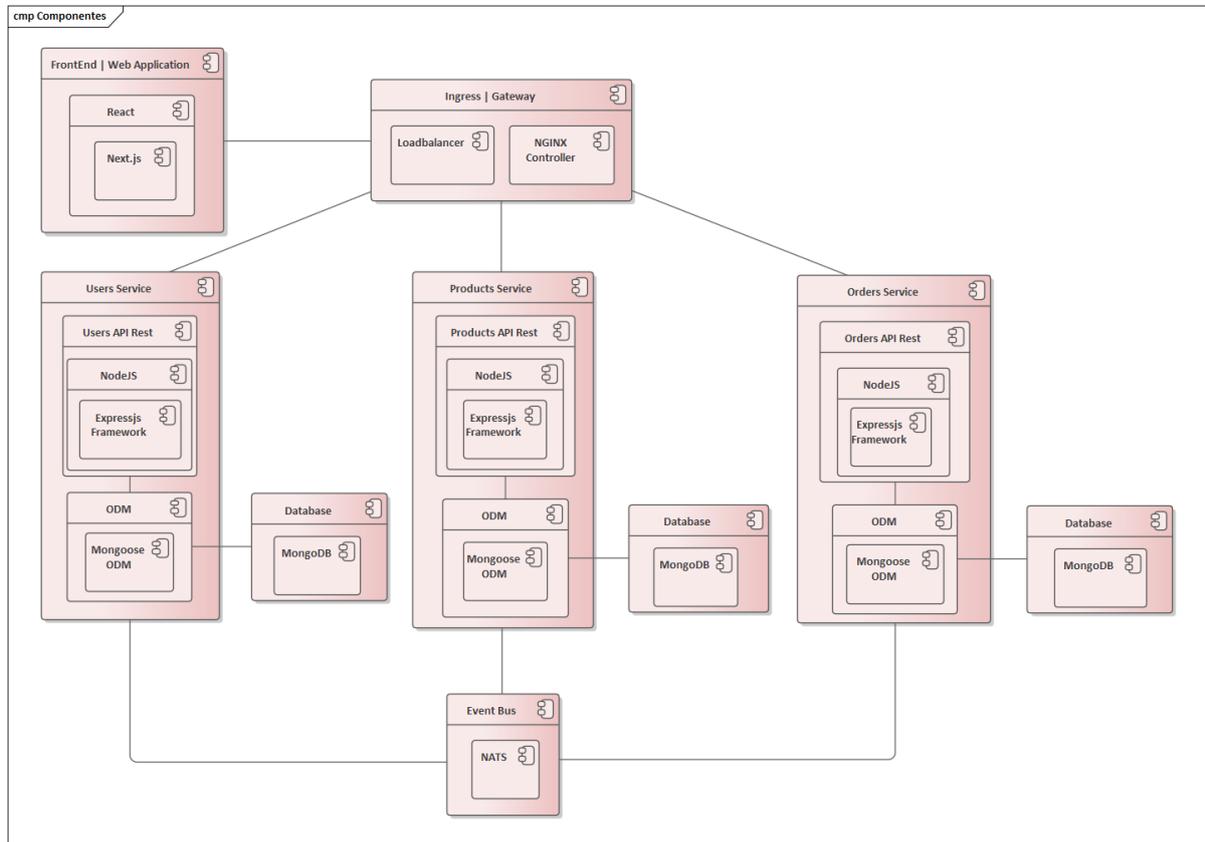


Figura 20. Diagrama de componentes de microservicios

Como se puede apreciar en la figura más atrás, el diseño de la comunicación entre servicios viene dado por un bus de eventos. El comportamiento de dicho bus se detalla en la Figura 21, donde se ilustra, a modo de ejemplo, la comunicación de eventos entre los servicios de Productos y Órdenes.

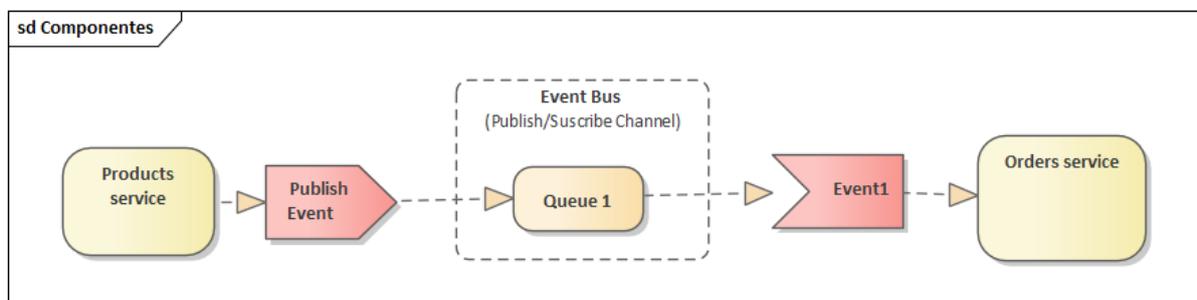


Figura 21. Representación gráfica del comportamiento del bus de eventos

En la Figura 22 se presenta el diagrama de despliegue de la aplicación en entornos de producción, la solución responde a un diseño arquitectónico de Microservicios, compuesto de tres aplicaciones dockerizadas, que son usuarios, productos y órdenes envueltas por un clúster Kubernetes. Adicional consta de un servicio de FrontEnd, también dockerizado, que es una aplicación web. En la Tabla 14 se describe a detalle el estilo arquitectónico de cada servicio, así como el objetivo bajo el cual fue diseñado:

Tabla 14. Servicios, objetivo y estilo arquitectónico

Servicio	Objetivo	Estilo arquitectónico	Plataforma de desarrollo
Usuarios	Gestionar las cuentas de usuario, privilegios y roles.	Capas	JavaScript usando tipado TypeScript, framework Nodejs y almacenamiento en MongoDB
Productos	Gestionar productos y categorías para presentar la información en un menú.	Capas	JavaScript usando tipado TypeScript, framework Nodejs y almacenamiento en MongoDB
Órdenes	Gestionar órdenes, así como el cobro por consumo y emisión de comprobantes de pago.	Capas	JavaScript usando tipado TypeScript, framework Nodejs y almacenamiento en MongoDB
FrontEnd	Consumir el API en una aplicación web para la interacción con los usuarios.	Capas	JavaScript con la librería React y el framework Next.js.

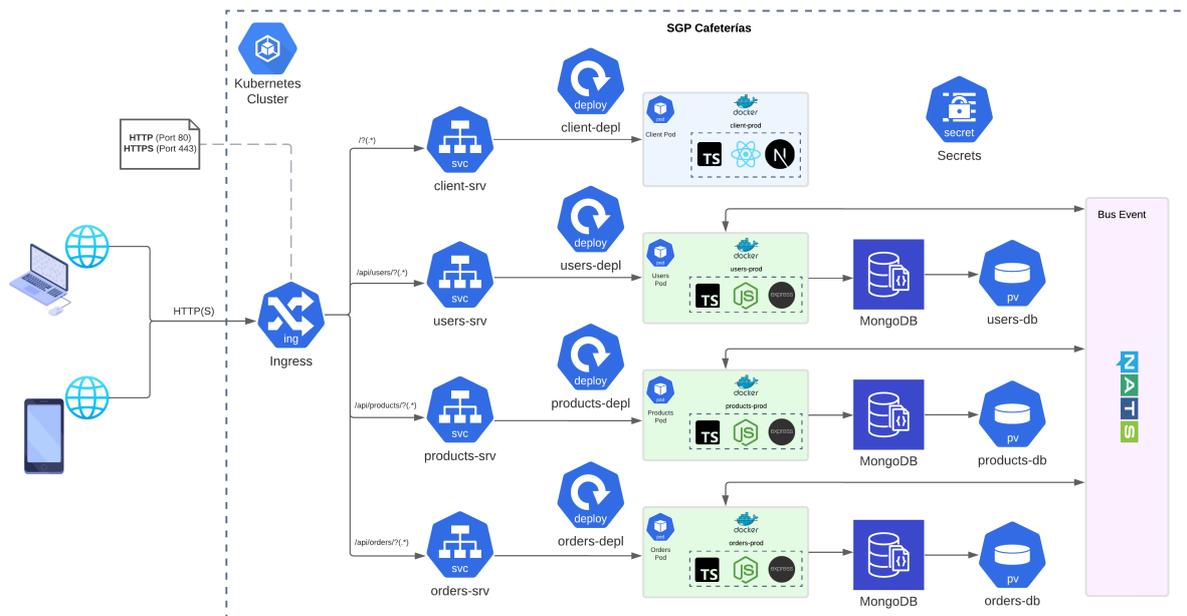


Figura 22. Diagrama de despliegue de microservicios

En la figura anterior, se visualiza la representación de un clúster Kubernetes que alberga cada uno de los microservicios. El acceso externo a estos servicios se gestiona mediante un recurso Ingress, que actúa como una puerta de enlace, verificando el tráfico entrante y, a través de un conjunto de reglas definidas, dirige las solicitudes hacia el servicio correspondiente.

Haciendo mención al concepto de servicio (Service), es importante destacar que este no es más que otro objeto de Kubernetes que permite que los Pods se expongan en la red interna del clúster, siendo accesibles mediante una dirección IP y un puerto específico. En este proyecto, se han definido Pods para los servicios de Usuarios, Productos, Órdenes y el Cliente (Aplicación Web). Cada Pod contiene un conjunto de contenedores Docker que comparten recursos.

Para gestionar de manera efectiva las actualizaciones que puedan experimentar estos Pods, se han establecido Deployments. Además, con el propósito de garantizar la persistencia de los datos, se han definido Persistent Volumes, los cuales están vinculados a cada Pod. Esta configuración asegura que los datos manejados en la base de datos MongoDB se almacenen en un espacio físico dentro del clúster.

6.2. Objetivo 2: Implementar la arquitectura de la solución informática mediante el uso de tecnologías JavaScript y la metodología ICONIX

A continuación, se presentan los resultados obtenidos de la fase de codificación de la solución informática. Las tecnologías empleadas han sido, como base, el lenguaje de programación JavaScript, Sistema de Gestión de Base de datos MongoDB. El desarrollo de esta fase se realizó según la planificación presente en la tabla:

Tabla 15. Planificación de desarrollo de servicios

Servicio	Casos de uso	Tiempo empleado (N° de semanas)
Configuración del espacio de trabajo		
Usuarios	CU01, CU03, CU04, CU05, CU06	1
Productos	CU02, CU14, CU15, CU16, CU17	1
Órdenes	CU14, CU15, CU16, CU17	1.5
Cliente (FrontEnd)	CU18, CU19	3

Conforme a la planificación expuesta en la Tabla 15, a continuación, se detallan las actividades llevadas a cabo.

6.2.1. Configuración de espacio de trabajo

El entorno de trabajo empleado, considerando que la implementación se basa en un clúster Kubernetes, está compuesto por los servicios de Usuarios, Productos, Órdenes y Cliente. Además, se incluye una carpeta dedicada a recursos compartidos entre los servicios denominada "shared", y un directorio, denominado "infra", destinado a la configuración de archivos de manifiesto útiles para el despliegue de diversos recursos de Kubernetes. Este arreglo se puede apreciar de manera más detallada en la Figura 23.

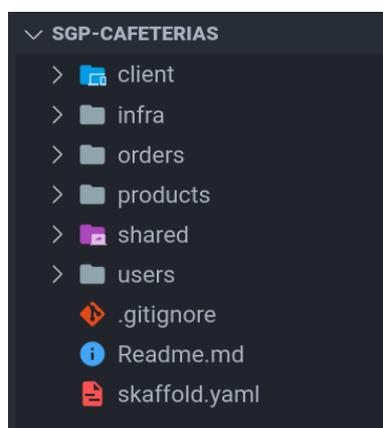


Figura 23. Espacio de trabajo general

Cada servicio incluye un archivo Dockerfile que detalla las instrucciones, presentadas como pasos, para ensamblar cada aplicación en una imagen Docker. Posteriormente, esta imagen se utilizará para su ejecución dentro del clúster. En la Figura 24 se presenta un ejemplo que ilustra las instrucciones establecidas en el archivo Dockerfile.



```
sgp-cafeterias - Dockerfile
1 FROM node:alpine
2 WORKDIR /app
3 COPY package.json .
4 RUN npm install
5 COPY . .
6 CMD ["npm", "run", "dev"]
```

Figura 24. Configuración de archivo Docker

La configuración de los distintos objetos del clúster Kubernetes, tales como Pods, Deployments, Servicios, Bus de Eventos y la Puerta de enlace (recursos definidos en el diagrama de despliegue de la Figura 22), está especificada en los archivos de manifiesto ubicados en la carpeta “infra”. Estos archivos están en formato YAML, como se puede apreciar en la Figura 25.

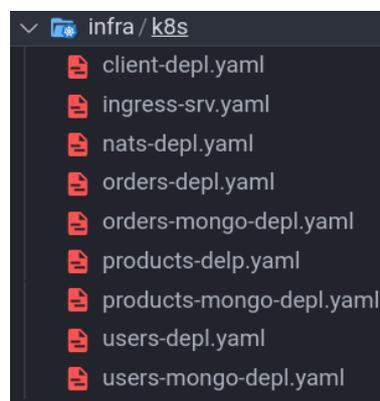


Figura 25. Espacio de trabajo de configuración del clúster Kubernetes

La Puerta de Enlace, o Ingress, tiene la responsabilidad de dirigir el tráfico proveniente de las peticiones externas de acuerdo con un conjunto de reglas definidas en un sistema basado en rutas. En la Figura 26, se han establecido estas reglas para garantizar el redireccionamiento adecuado de las solicitudes hacia cada uno de los servicios.

```
sgp-cafeterias - ingress-srv.yaml
14 http:
15   paths:
16     - path: /api/users/(?.*).
17       pathType: Prefix
18       backend:
19         service:
20           name: users-srv
21           port:
22             number: 3000
23     - path: /api/products/(?.*).
24       pathType: Prefix
25       backend:
26         service:
27           name: products-srv
28           port:
29             number: 3000
30     - path: /api/orders/(?.*).
31       pathType: Prefix
32       backend:
33         service:
34           name: orders-srv
35           port:
36             number: 3000
37     - path: /(?.*).
38       pathType: Prefix
39       backend:
40         service:
41           name: client-srv
42           port:
43             number: 3000
```

Figura 26. Configuración de puerta de enlace Ingress

En el contexto de los archivos de manifiesto destinados al despliegue de cada microservicio (ver Figura 27), es importante destacar que dichos documentos se componen de dos elementos fundamentales:

1. **Deployment:** permite especificar la creación de pods (cada pod ejecuta la imagen Docker construida por cada microservicio).
2. **Service:** permite la comunicación del pod con otros pods dentro del mismo clúster mediante una ip administrada por Kubernetes y un puerto establecido.

```
sgp-cafeaterias - client-depl.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: client-depl
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: client
10   template:
11     metadata:
12       labels:
13         app: client
14     spec:
15       containers:
16         - name: client
17           image: aniiencalada/client
18
19   ---
20   apiVersion: v1
21   kind: Service
22   metadata:
23     name: client-srv
24   spec:
25     selector:
26       app: client
27     ports:
28       - name: client
29         protocol: TCP
30         port: 3000
31         targetPort: 3000
32
```

Figura 27. Manifiestos Kubernetes de Deployment y Service

6.2.2. Servicios

La estructura de los servicios, es similar para los tres, dado esto, en las siguientes figuras se muestra como fue construido en servicio de Órdenes. En este contexto, la carpeta "models" alberga las clases que fueron concebidas en el diagrama de clases correspondiente, mientras que las operaciones se encuentran en la carpeta "routes". Con el propósito de facilitar la comunicación entre los diversos servicios, se implementó un bus de eventos en la carpeta "events". Adicionalmente, se observa que se han creado y/o configurado archivos adicionales en la carpeta "config" para gestionar funciones específicas, tales como el envío de correos electrónicos y otras operaciones relevantes.

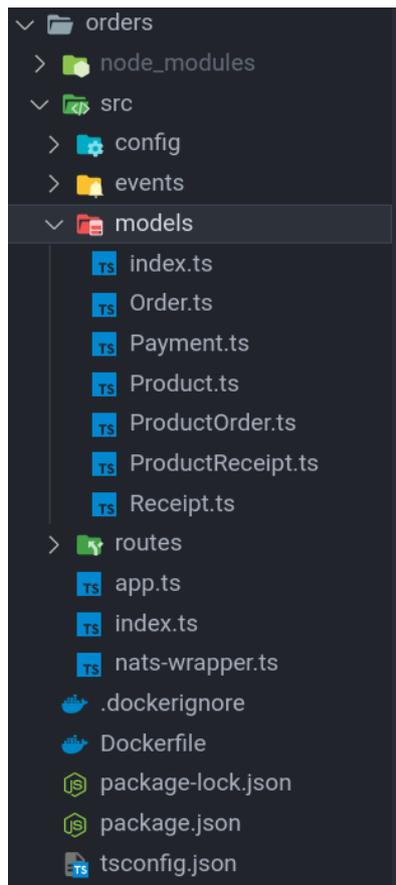


Figura 28. Espacio de trabajo de servicio órdenes

A continuación, en la Figura 29, se presenta de manera detallada la declaración y construcción de un modelo, junto con sus diversos atributos y las correspondientes validaciones asociadas.

```
sgp-cafeterias - Order.ts
1 import { OrderStatus, PaginatedModel } from "@cafetho/shared";
2 import { getModelForClass, prop, Ref } from "@typegoose/typegoose";
3
4 class Order extends PaginatedModel {
5   @prop({ required: true, unique: true })
6   public code!: string;
7
8   @prop({ required: true })
9   public totalValue!: number;
10
11  @prop({ default: OrderStatus.create, enum: Object.values(OrderStatus) })
12  public status?: OrderStatus;
13
14  @prop()
15  public table?: string;
16
17  @prop({ ref: "User", required: true })
18  public user!: Ref<"User">;
19 }
20
21 const OrderModel = getModelForClass(Order);
22
23 export { Order, OrderModel };
```

Figura 29. Modelo Orden en el servicio de órdenes

En la Figura 30, se muestra la definición de los endpoints para el servicio de órdenes. En el ejemplo específico presentado, se destaca la función encargada de listar todos los comprobantes de venta. Con el objetivo de facilitar tanto el mantenimiento como la escalabilidad de las operaciones, se optó por la escritura de archivos individuales para cada función diseñada en el diagrama de clases. Este enfoque modular contribuye a una gestión más eficiente y adaptable de las operaciones del servicio.

```
sgp-cafeterias - allReceipt.ts
10 const allReceiptRouter = Router();
11
12 allReceiptRouter.get(
13   "/receipts",
14   requireAuth,
15   authorization(Permissions["read:receipts"]),
16   async (req: Request, res: Response) => {
17     const receipt = await ReceiptModel.paginate(
18       ...new PaginateParameters(req).get()
19     );
20
21     res.json(receipt);
22   }
23 );
24
25 export { allReceiptRouter };
```

Figura 30. Controlador para obtener todas las órdenes en el servicio de órdenes

Para consolidar la definición de todas las rutas que integran el API del servicio, se ha implementado un enfoque centralizado mediante la creación de un solo archivo. Este archivo, como se ilustra en la Figura 31, exporta todas las rutas, proporcionando así una visión unificada y organizada de la estructura del servicio. Este enfoque centralizado simplifica la gestión y el mantenimiento del sistema al proporcionar una referencia única para todas las rutas del API.

```
sgp-caferterias - index.ts
12  const prefix = "/api";
13
14  const ordersURI = `${prefix}/orders`;
15
16  export const routes = (app: Express) => {
17    //Order Routes
18    app.use(ordersURI, allReceiptRouter);
19    app.use(ordersURI, allOrderRouter);
20    app.use(ordersURI, countProductReceiptRouter);
21    app.use(ordersURI, allProductOrderRouter);
22    app.use(ordersURI, oneOrderRouter);
23    app.use(ordersURI, oneReceiptRouter);
24    app.use(ordersURI, saveOrderRouter);
25    app.use(ordersURI, saveReceiptRouter);
26    app.use(ordersURI, updateOrderRouter);
27  };
28
```

Figura 31. Rutas del servicio de órdenes

En coherencia con lo expuesto en secciones previas, se implementó un bus de eventos para facilitar la gestión de la comunicación entre el servicio de productos y pedidos, como se ejemplifica en este caso. Los archivos pertinentes destinados a este propósito se presentan en la Figura 32, destacando la estructura utilizada para este componente fundamental en la arquitectura de microservicios.

```
sgp-caferterias - Events
2  └─ events
3    └─ index.ts
4    └─ listeners
5        └─ index.ts
6            └─ product-created-listener.ts
7            └─ product-updated-listener.ts
8            └─ queue-group-name.ts
9    └─ publishers
10        └─ index.ts
11        └─ receipt-created-publisher.ts
```

Figura 32. Directorio del bus de eventos

El directorio en cuestión alberga una subcarpeta identificada como "listeners", la cual está destinada a gestionar los escuchadores del bus de eventos. Estos archivos, representados visualmente en la Figura 33, tienen la responsabilidad de monitorear posibles modificaciones en otros servicios. En el escenario descrito, el servicio de pedidos actúa como

oyente, verificando si se ha creado un nuevo producto. En caso afirmativo, se desencadena una funcionalidad específica; en este ejemplo, el servicio de pedidos genera una copia del producto recién creado en su propia base de datos.

```
sgp-cafeterias - product-created-listener.ts

6 export class ProductCreatedListener extends Listener<ProductCreatedEvent> {
7   readonly subject: Subjects.ProductCreated = Subjects.ProductCreated;
8   queueGroupName = queueGroupName;
9   async onMessage(data: ProductCreatedEvent["data"], msg: Message) {
10    const product = new ProductModel({ ...data, _id: data.id });
11    await product.save();
12    msg.ack();
13  }
14 }
```

Figura 33. Escuchador de cambios en el registro de productos

Al igual que se tiene la capacidad de estar alerta a eventos mediante los escuchadores, se posibilita la acción contraria: la publicación de eventos. Como se ilustra en la Figura 34, se observa un componente publicador encargado de transmitir la información asociada a un nuevo registro de producto hacia aquellos servicios que se hallan suscritos a dicho evento

```
sgp-cafeterias - product-created-publisher.ts

1 import { ProductCreatedEvent, Publisher, Subjects } from "@cafetho/shared";
2
3 export class ProductCreatedPublisher extends Publisher<ProductCreatedEvent> {
4   readonly subject: Subjects.ProductCreated = Subjects.ProductCreated;
5 }
6
```

Figura 34. Publicador de cambios en el registro de producto

6.2.3. Servicio FrontEnd

La configuración del cliente o FrontEnd fue diseñada considerando componentes, páginas y funciones de llamadas al API Rest. Una visión panorámica de su estructura se presenta en la Figura 35.

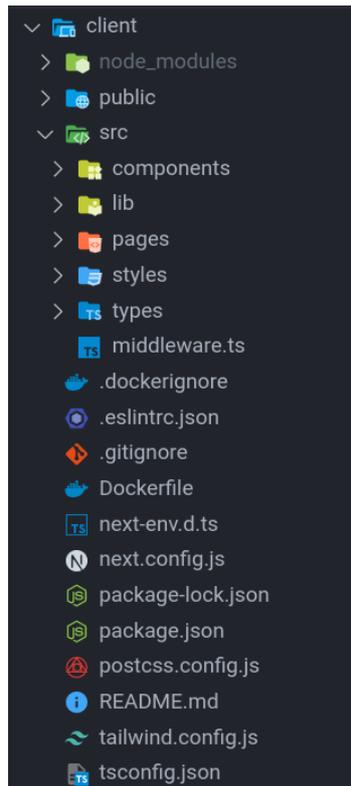


Figura 35. Espacio de trabajo de FrontEnd

Al explorar los directorios y archivos más pertinentes, se observa en la Figura 36 la presencia de un archivo denominado "middleware". Este archivo desempeña la función de verificar la existencia de un usuario autenticado en el sistema, asegurando de este modo determinadas rutas en el sitio web que requieren una sesión de usuario.

```
sgp-cafeterias - middleware.ts

5  export function middleware(req: NextRequest) {
6    const session = req.cookies.get("session");
7
8    if (!session) {
9      const url: NextURL = req.nextUrl.clone();
10     const searchParams = url.search;
11     url.search = "";
12     url.searchParams.set("returnUrl", `${url.pathname}${searchParams}`);
13     url.pathname = `/ingresar`;
14     return NextResponse.redirect(url);
15   }
16
17   return NextResponse.next();
18 }
19
20 export const config = {
21   matcher: ["/admin/:path*"],
22 };
```

Figura 36. Middleware de autenticación

Un directorio crucial es el designado como "components" (ver Figura 37), el cual hace alusión a elementos dispuestos como piezas individuales. Estos elementos contienen tanto la lógica como el diseño necesario para la construcción de las interfaces gráficas del sitio web.

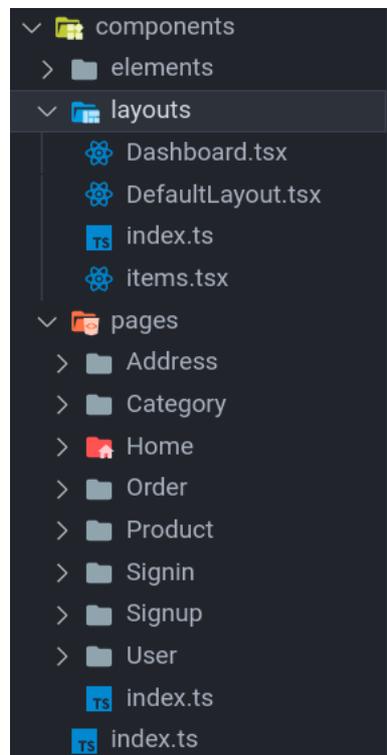


Figura 37. Espacio de trabajo de componentes contenedores de páginas

De igual manera, se encuentran presentes los "Layouts" o plantillas que encapsulan la estructura y diseño fundamental de una página web. En el caso ilustrado en la Figura 38, se muestra cómo se ha desarrollado un "Layout" o componente contenedor predeterminado para las interfaces gráficas de la aplicación.

```
sgp-cafeaterias - DefaultLayout.tsx
1 import { PropsWithChildren } from "react";
2
3 export const DefaultLayout = (props: PropsWithChildren) => {
4   return (
5     <div className="h-full">
6       <div className="min-h-full flex items-center justify-center py-24 px-4 sm:px-6 lg:px-8">
7         <div className="max-w-3xl w-full space-y-8">
8           <div className="hero">
9             <div className="hero-content flex-col lg:flex-row-reverse">
10              {props.children}
11            </div>
12          </div>
13        </div>
14      </div>
15    </div>
16  );
17 };
18
```

Figura 38. Componente contenedor por defecto de la aplicación

El consumo del API Rest se segmentó, asignando las peticiones GET a la carpeta "api" y las peticiones POST y PUT al directorio "services", con el propósito de optimizar la gestión de los recursos utilizados para enviar solicitudes a los servicios. La configuración de estos archivos se manejó conforme se ilustra en la Figura 39.

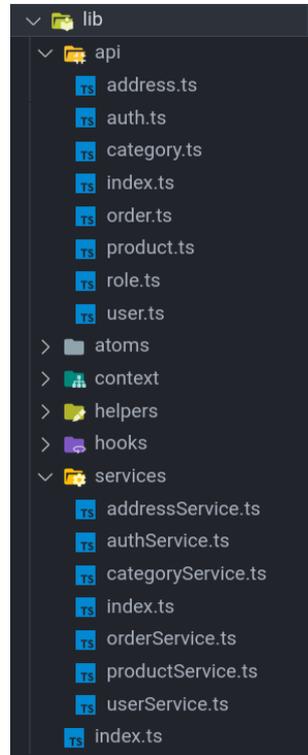


Figura 39. Consumo de API Rest

En las dos figuras siguientes se ve el consumo del API. En la Figura 40 se puede visualizar un ejemplar de peticiones GET, mientras que en la Figura 41 se puede observar un ejemplo para las peticiones de tipo POST y PUT.

```
sgp-cafeterias - address.ts
5  export const useAddresses = (entity: string) => {
6    const dataPagination = useDataPagination<EntityModel.Response>();
7    const request = getPaginationRequest("/api/users/entity", {
8      dataPagination,
9      initialParams: { user: entity },
10     searchKeys: ["dni", "firstname", "lastname"],
11   });
12
13   const addresses =
14     useSWRRequest<PaginationModel.Response<EntityModel.Response>>(request);
15   return { ...addresses, ...dataPagination };
16 };
```

Figura 40. Peticiones de tipo lectura (GET)

```
sgp-cafeterias - orderService.ts

1 import { httpClient } from "@lib/helpers";
2 import { OrderModel } from "@types";
3
4 export const OrderService = {
5   save: (values: OrderModel.NewValues) => {
6     return httpClient.post<OrderModel.Response>("/api/orders", values);
7   },
8   update: (values: OrderModel.UpdateValues, id: string) => {
9     return httpClient.put(`/api/orders/${id}`, values);
10  },
11 };
12
```

Figura 41. Peticiones de tipo escritura (POST y PUT)

La Figura 42 muestra las vistas que fueron construidas para la aplicación web. Es importante mencionar que los nombres de los archivos, son los nombres de las rutas, por lo que no fue necesario un archivo para la configuración de éstas.



Figura 42. Espacio de trabajo de páginas

A continuación, se presenta en la Figura 43 la implementación de un componente asociado a una página específica del sitio web. Estas páginas hacen uso de los componentes previamente creados, así como del diseño base proporcionado por el layout destinado a la construcción de la página en cuestión. De este modo, los componentes ubicados en el directorio "pages" contienen la mínima lógica necesaria, ya que esta se encuentra implementada en cada uno de los componentes que los componen.

```
sgp-cafeaterias - mi-orden.tsx
5
6 const MyOrder: NextPageWithLayout = (props) => {
7   const pageTitle = `Mi orden | ${props.siteName}`;
8
9   return (
10    <>
11      <Head>
12        <title>{pageTitle}</title>
13      </Head>
14
15      <Toolbar title="Mi orden" leading="Detalles de orden" />
16
17      <div className="flex flex-col lg:flex-row shadow-md my-10">
18        <ProductList />
19        <OrderSummary />
20      </div>
21    </>
22  );
23 };
24
25 MyOrder.getLayout = (page: React.ReactElement) => <Dashboard>{page}</Dashboard>;
26
27 export default MyOrder;
```

Figura 43. Página Cambiar clave

Adicionalmente a los servicios y vistas desarrolladas, se ha creado un proyecto denominado "shared", diseñado como una librería para compartir recursos específicos entre los servicios, tales como eventos o tipos de dato. Este proyecto, creado con Node.js, alberga archivos y directorios que son compartidos entre los distintos servicios. La estructura del mismo proyecto se muestra en la Figura 44.

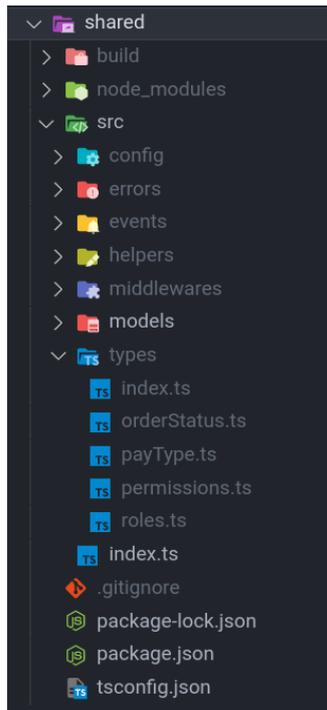


Figura 44. Espacio de trabajo del proyecto de intercambio de recursos

6.2.4. Realizar pruebas de funcionamiento del software

Las pruebas que se realizaron fueron unitarias y de integración. Se hizo uso de Jest para automatizar los casos de prueba, siendo el factor a considerar para las pruebas unitarias funciones simples y atómicas, encargadas de operaciones concretas como la validación de cédulas o generación de códigos. Las pruebas de integración, fueron del comportamiento de los endpoints al ejecutarse y la correlación existente entre los mismos.

La estructura de los archivos de prueba, siguen el formato que se ve en la Figura 45.

```
sgp-cafeterias - signin.test.ts
11 it("Recibir un error cuando se proporciona una contraseña incorrecta", async () => {
12   await global.insertRoles();
13
14   await request(app)
15     .post("/api/users/signup")
16     .send({
17     email: "test@test.com",
18     password: "password123",
19     firstname: "Maria",
20     lastname: "Encalada",
21     dni: "0707031662",
22   })
23   .expect(201);
24
25   await request(app)
26     .post("/api/users/signin")
27     .send({
28     email: "test@test.com",
29     password: "passwordwrong",
30   })
31   .expect(400);
32 });
```

Figura 45. Estructura de archivos de prueba

Seguidamente, se presentan los resultados de las pruebas unitarias y de integración de los siguientes servicios: usuarios (ver Figura 46), productos (ver Figura 47) y pedidos (ver Figura 48). En todas las figuras se puede observar que se han superado satisfactoriamente los tests definidos.

```
PASS src/routes/___test___/current-user.test.ts (10.644 s)
PASS src/routes/___test___/logout.test.ts (10.673 s)
PASS src/routes/___test___/signup.test.ts (10.73 s)
PASS src/routes/___test___/signin.test.ts (10.824 s)

Test Suites: 4 passed, 4 total
Tests:      8 passed, 8 total
Snapshots:  0 total
Time:       11.162 s, estimated 13 s
Ran all test suites.
```

Figura 46. Resultado de pruebas unitarias/integración del servicio de usuarios

```
PASS src/routes/___test___/category.test.ts
PASS src/routes/___test___/product.test.ts (11.347 s)

Test Suites: 2 passed, 2 total
Tests:      12 passed, 12 total
Snapshots:  0 total
Time:       11.596 s, estimated 13 s
```

Figura 47. Resultado de pruebas unitarias/integración del servicio de productos

```
PASS src/routes/___test___/order.test.ts (6.402 s)
PASS src/routes/___test___/receipt.test.ts (6.43 s)

Test Suites: 2 passed, 2 total
Tests:      5 passed, 5 total
Snapshots:  0 total
Time:       6.687 s, estimated 8 s
Ran all test suites.
```

Figura 48. Resultado de pruebas unitarias/integración del servicio de pedidos

6.3. Objetivo 3: Comparar los procesos de toma de pedidos tradicional y web en un ambiente de pruebas simulado dentro de la FEIRNNR para determinar los beneficios de la actualización

Durante la elaboración de este TT y dada la gradual reanudación de actividades presenciales, no se contaba con una cafetería universitaria en la FEIRNNR. Sin embargo, la cafetería de la carrera de turismo, con características similares, se involucró en la fase inicial del proyecto y se consideró para la fase 3, ya que su comportamiento es extrapolable a ambas.

Después de implementar los módulos, se realizó una evaluación en escenarios simulados para comparar el desempeño del sistema y las ventajas del proceso automatizado (web) frente al tradicional con la finalidad de encontrar la forma más viable para la gestión de pedidos en cafeterías. Cabe mencionar que la automatización es parcial y su objetivo busca disminuir la interacción presencial en la gestión de pedidos de cafeterías en Loja. Además, todo el proceso se basó en las etapas propuestas por la metodología “Implantación de un Sistema ERP en una PyME” del autor Pablo González [1].

6.3.1. Planificación del caso de prueba para la simulación manual versus web

En esta sección se describe el plan de pruebas que se diseñó para evaluar el sistema automatizado y compararlo con el proceso manual. El objetivo de las pruebas es medir el tiempo requerido en las actividades y la satisfacción de los usuarios al realizar pedidos de cafetería mediante la plataforma web o el método tradicional. Para ello, se definió el comité del proyecto y el cronograma para las actividades a realizar.

6.3.1.1. Consolidación del comité del proyecto

En esta subsección se presenta la conformación del comité del proyecto, que es el grupo de personas encargadas de planificar, ejecutar y supervisar las pruebas del sistema automatizado y el proceso manual. El comité está integrado por el director del TT, el autor del TT, y los usuarios que se encargaran de evaluar. A continuación, en la Tabla 16 se describen las funciones y responsabilidades de cada uno de ellos.

Tabla 16. Comité del proyecto

Autor	Cargo	Rol
Director de TT	Individual	Supervisar el trabajo técnico realizado
Autor del TT	Individual	Desplegar y configurar la herramienta desarrollada
Gerente-Administrador de Cafetería	Individual	Comprobar el correcto funcionamiento de la solución desarrollada y emitir comentarios
Clientes	Grupal	Comprobar el correcto funcionamiento de la solución desarrollada y emitir comentarios

6.3.1.2. Establecimiento del cronograma

En esta subsección se muestra el cronograma de las actividades que se realizaron para llevar a cabo las pruebas del sistema automatizado y el proceso manual. A continuación, en la Tabla 17 se detallan las actividades, los responsables y los tiempos estimados de cada fase.

Tabla 17. Cronograma de actividades

Actividades	Actores	Semana 1					Semana 2					Semana 3					Semana 4					
		L	M	M	J	V	L	M	M	J	V	L	M	M	J	V	L	M	M	J	V	
Documentación de manual de instalación de la aplicación web.	Director TT y Autor de TT	■	■	■																		
Documentación de manual de usuario de la aplicación web.	Director TT y Autor de TT			■	■	■																
Carga de Datos y Configuraciones	Director TT y Autor de TT						■	■														
Identificación de los escenarios de evaluación.	Director TT y Autor de TT						■	■														
Creación de los escenarios de prueba.	Director TT y Autor de TT								■	■	■											
Pruebas de los escenarios en la forma tradicional (manual).	Autor de TT, Gerente-Administrador de Cafetería											■	■	■	■	■						
Pruebas de los escenarios en la forma automatizada.	Autor de TT, Gerente-Administrador de Cafetería											■	■	■	■	■						
Elaboración y aplicación de una encuesta de satisfacción.	Autor de TT y Clientes															■	■	■	■			
Aplicación de Acta de Validación de la aplicación web.	Autor de TT y Gerente-Administrador de Cafetería																			■	■	

6.3.2. Ejecución de la simulación

En esta sección se describe el proceso de ejecución de la simulación, que consiste en la gestión de los pedidos de cafetería por parte de los usuarios mediante la plataforma web o el método tradicional, según el escenario asignado. El objetivo de la simulación es recolectar los datos necesarios para medir el tiempo y la satisfacción de los usuarios. A continuación, se explican los pasos que se siguieron para llevar a cabo la simulación.

6.3.2.1. Carga de Datos y Configuraciones

El primer paso consiste en realizar las configuraciones de las tecnologías y herramientas que se emplearon para el funcionamiento de la solución informática. Estas configuraciones incluyen aspectos como la instalación de los programas necesarios, la creación de las bases de datos, la definición de los permisos y las conexiones entre los componentes de la aplicación. Para facilitar este proceso, se elaboró un manual de instalación y configuración que detalla los pasos a seguir, el mismo que se puede visualizar en **Anexo 4. Manual de instalación del sistema informático.**

6.3.2.2. Pruebas

La siguiente actividad implica llevar a cabo las pruebas de satisfacción del uso de la aplicación con un grupo compuesto por estudiantes y miembros de comunidades tecnológicas, quienes representan a los usuarios potenciales del sistema automatizado. Para evaluar la satisfacción, se empleó una encuesta basada en la escala de Likert, un método que permite evaluar actitudes y opiniones mediante afirmaciones con opciones de respuesta graduadas [37]. Las preguntas de esta encuesta fueron seleccionadas con el propósito de cuantificar el cumplimiento de los requerimientos no funcionales establecidos en las primeras etapas de este Trabajo de Titulación, abordando aspectos como la experiencia del usuario, eficiencia y usabilidad de la solución informática. En la Tabla 18 se presenta el conjunto de preguntas utilizado para la encuesta.

Tabla 18. Banco de preguntas de la encuesta de satisfacción

Pregunta	Respuesta				
	1	2	3	4	5
¿La aplicación es fácil de usar?	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
¿Considera que la manera de presentar el menú y categorías es adecuada?	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
Al realizar un pedido de un plato/bebida ¿la aplicación agiliza el proceso?	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

¿Cómo calificaría el proceso de realizar un pedido?	Muy difícil	Difícil	Neutral	Fácil	Muy Fácil
El proceso de búsqueda de productos en la aplicación, para usted, representó una dificultad de:	Muy difícil	Difícil	Neutral	Fácil	Muy Fácil
¿Qué tan probable es que usted vuelva a utilizar la aplicación?	Nada probable	Poco probable	Neutral	Probable	Muy probable
¿Qué tan probable es que recomiende la aplicación a otras personas?	Nada probable	Poco probable	Neutral	Probable	Muy probable
¿Qué le pareció la experiencia de usar la aplicación?	Nada agradable	Poco agradable	Neutral	Agradable	Muy agradable
¿En qué podría mejorar la presente aplicación? Por favor, Indique su recomendación.	Recomendación (Opcional)				

A continuación, se adjuntan los anexos referentes a la encuesta de satisfacción que se efectuó a los usuarios. El **Anexo 5. Modelo de Encuesta de Satisfacción** muestra el formato final de la encuesta que se empleó, con las preguntas y las alternativas de respuesta. El **Anexo 6. Resultados de la Encuesta de Satisfacción** expone los resultados de la encuesta y los porcentajes de cada alternativa. Estos anexos permiten evaluar la impresión y el criterio de los usuarios sobre el sistema automatizado.

Asimismo, para comparar los procesos de toma de pedidos tradicional y web, se definieron escenarios de evaluación que representan las situaciones más comunes que se presentan en las cafeterías. En la Tabla 19 se describen los escenarios que se identificaron para cada proceso.

Tabla 19. Escenarios de prueba para la comparación del proceso tradicional vs web

Número	Escenario de prueba
1	Registro de producto nuevo (y actualización de la carta/menú)
2	Toma del pedido (incluyendo vista de menú, toma de orden, paso a cocina y entrega del pedido en la mesa)
3	Cobro por consumo (identificación de consumo y cobro de cuentas separadas por múltiples consumos en una misma mesa)

Los escenarios de evaluación mencionados se utilizaron principalmente para recolectar la información sobre la organización de actividades y el tiempo empleado en cada proceso de toma de pedidos. Esta información permite comparar el desempeño y la eficiencia de los procesos tradicional y web. En el **Anexo 7. Acta de validación de pruebas del sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless)** se muestra el resultado de la ejecución de los escenarios, así como el acta de validación de pruebas que certifica su veracidad.

6.3.2.3. Documentación

En este apartado se describe la documentación que se elaboró para el sistema automatizado. Se trata del manual de usuario, que es un documento que explica el funcionamiento y el uso del sistema, desde la perspectiva de cada uno de los roles que conforman la solución informática. El manual de usuario se encuentra en el **Anexo 8. Manual de usuario del sistema informático.**

6.3.3. Establecer un análisis cuantitativo de los resultados

En esta sección se presentan los resultados del análisis cuantitativo de los datos recolectados en la simulación. Estos datos corresponden a la encuesta de satisfacción y a los escenarios de evaluación que se aplicaron usando un muestreo por conveniencia tanto a los usuarios de las cafeterías como a los locales que brindan este tipo de servicio para comparar los procesos de toma de pedidos tradicional y web. El análisis cuantitativo permite medir el tiempo y la satisfacción de los usuarios en cada proceso y establecer las diferencias y las ventajas de cada uno.

En primer lugar, respecto a los resultados de la encuesta de satisfacción, la Tabla 20 muestra el porcentaje de aceptación de cada una de las preguntas que se hicieron a los usuarios. Para calcular este porcentaje se consideraron las escalas 3, 4 y 5, que representan una valoración positiva según la escala de Likert. De esta forma, se obtuvo que, de los 48 usuarios encuestados, el 97.14% aceptó la solución informática, lo que indica que el sistema informático cumple con su objetivo.

Tabla 20. Promedio de aceptación de la Encuesta de Satisfacción

Pregunta	1	2	3	4	5	Promedio de Aceptación
¿La aplicación es fácil de usar?	2.08%	0.00%	6.25%	16.67%	75.00%	97.92%
¿Considera que la manera de presentar el menú y categorías es adecuada?	2.08%	4.17%	10.42%	22.92%	60.42%	93.75%
Al realizar un pedido de un plato/bebida ¿la aplicación agiliza el proceso?	2.08%	2.08%	6.25%	18.75%	70.83%	95.83%
¿Cómo calificaría el proceso de realizar un pedido?	0.00%	2.08%	6.25%	27.08%	64.58%	97.92%
El proceso de búsqueda de productos en la aplicación, para usted, representó una dificultad de:	0.00%	4.17%	8.33%	27.08%	60.42%	95.83%
¿Qué tan probable es que usted vuelva a utilizar la aplicación?	0.00%	2.08%	6.25%	20.83%	70.83%	97.92%
¿Qué tan probable es que recomiende la aplicación a otras personas?	0.00%	0.00%	8.33%	22.92%	68.75%	100.00%
¿Qué le pareció la experiencia de usar la aplicación?	0.00%	2.08%	4.17%	22.92%	70.83%	97.92%
Promedio Final de Aceptación						97.14%

A continuación, se realiza el análisis de los resultados obtenidos al aplicar los escenarios de evaluación para comparar los procesos de toma de pedidos tradicional y web. Los escenarios de evaluación son situaciones simuladas que reflejan las condiciones reales del entorno y permiten medir el desempeño y la eficiencia de los procesos. Se llevó a cabo esta evaluación en tres cafeterías, cuyos resultados se muestran en las figuras siguientes:



Figura 49. Resumen del tiempo empleado en el registro y actualización de menú en el método tradicional frente al proceso web.

En la Figura 49, se ve la diferencia entre la actualización del menú en cafeterías que tienen sus cartas impresas frente a la actualización mediante el sistema web.

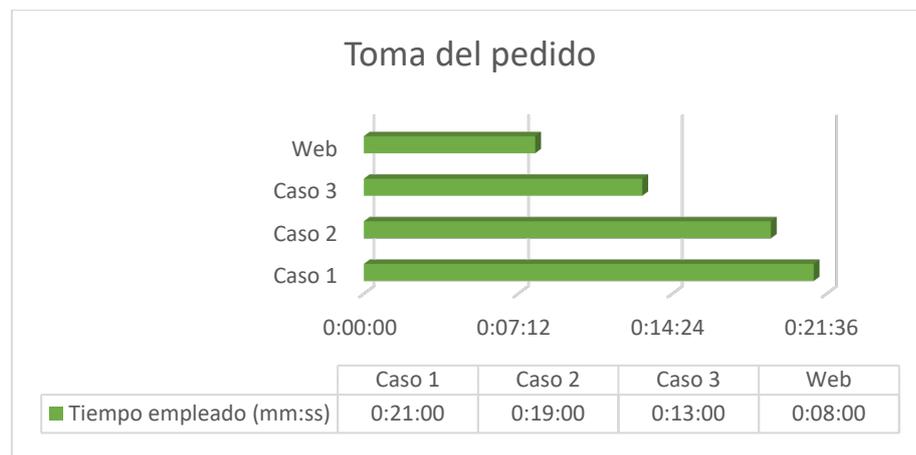


Figura 50. Resumen del tiempo empleado en la toma de pedidos en el método tradicional frente al proceso web.

En la Figura 50, se presenta el tiempo que tarda un cliente en recibir su pedido en mesa desde el momento que llega al local. Esto incluye el tiempo de preparación. El pedido que sirvió de referencia estuvo compuesto por 3 items diferentes del menú.

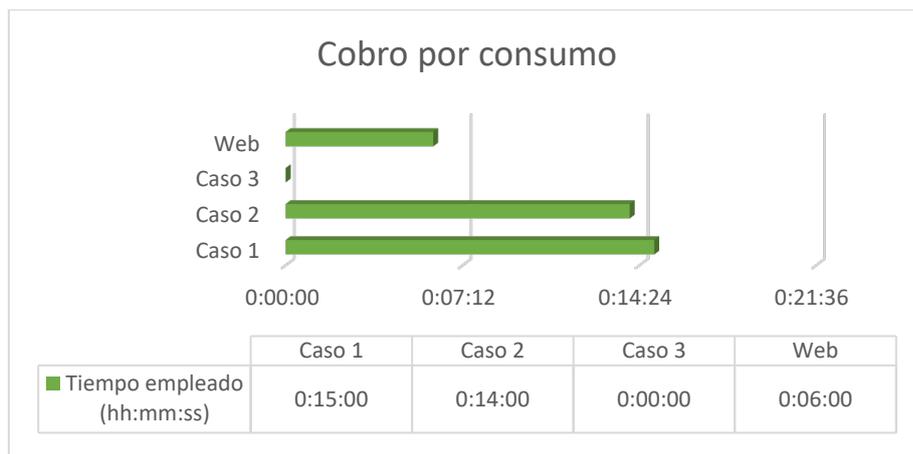


Figura 51. Resumen del tiempo empleado en el cobro por consumo en grupo o individual en el método tradicional frente al proceso web.

En la Figura 51, se muestra el tiempo necesario para cobrar y registrar el cobro por consumo. Para Caso 3 se observa que el tiempo empleado es de 0 segundos debido a que el cobro consta en el escenario 2.

Luego de comparar 3 casos en 3 diferentes escenarios, se puede evidenciar en los gráficos anteriores la diferencia de tiempo que existe entre los métodos tradicionales frente a los resultados con la solución informática para la gestión de pedidos en cafeterías. Esto supone un ahorro de tiempo para el personal y clientes al reducir pasos en sus flujos, favorece la mayor disponibilidad para otras actividades que incrementan el valor competitivo de los locales y facilita la actualización de la oferta en pocos minutos.

En promedio, los resultados se pueden presentar de la siguiente forma:



Figura 52. Promedio del tiempo empleado en el registro y actualización de menú en el método tradicional frente al proceso web

El Figura 52 se observa una diferencia promedio de tiempo de 42 minutos en el proceso de registrar un producto.

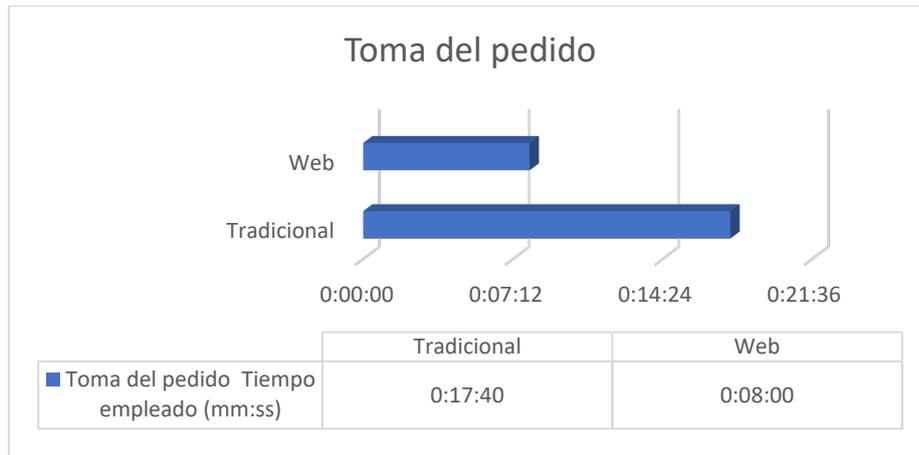


Figura 53. Promedio del tiempo empleado en la toma de pedidos en el método tradicional frente al proceso web.

Por su parte, el proceso de toma de pedidos evidencia una diferencia promedio de tiempo de 9 minutos y 40 segundos, tal como se puede observar en la Figura 53.



Figura 54. Promedio del tiempo empleado en el cobro por consumo en grupo o individual en el método tradicional frente al proceso web.

Para finalizar, se tiene una diferencia promedio de 8 minutos y 30 segundos en el proceso de cobro por consumo, tal como se evidencia en la Figura 54.

7. Discusión

En este proyecto de Trabajo de Titulación denominado “Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitecturas distribuidas (microservicios y/o serverless)” se busca implementar una solución informática que responda a las nuevas formas de interacción y a nuevos modelos de atención en cafeterías.

7.1. **Objetivo 1: Construir el modelo de arquitectura física y lógica en base a los requerimientos del usuario utilizando el Modelamiento de desarrollo de software dirigido por ICONIX.**

El uso de la metodología ICONIX ha sido favorable para el desarrollo del presente proyecto debido a que favorece la construcción de la documentación y su comportamiento iterativo e incremental respondió a la necesidad de hacer cambios a medida que se desarrollaban las actividades, permitiendo hacer correcciones y/o mejoras, (hacer referencia a autores que den peso al enunciado).

El diseño del sistema basado en la arquitectura microservicios fue una elección que facilitó la construcción del mismo, constando de 4 imágenes Docker que se integran con el uso de Kubernetes por lo que no es necesario una arquitectura sin servidor. A diferencia de Serverless, arquitectura sin servidor que llama funciones en la nube únicamente cuando son necesarias, microservicios respondió a las condiciones para el despliegue de la aplicación, requerimientos y presupuesto. Esta es una solución escalable, mantenible y que facilita que los cambios/mejoras se realicen en un servicio sin afectar o alterar a otros, además, se pueden integrar nuevos servicios a medida de las necesidades.

Para el diseño de la arquitectura, se partió de un análisis de casos de cafeterías en la ciudad de Loja. Se tomaron 4 cafeterías del catastro municipal de cafeterías y 1 cafetería a cargo de la carrera de Turismo de la Universidad Nacional de Loja, debido a que era la única que estaba funcionando en aquel entonces. La cuál ayudó a determinar flujos normales y los procesos que siguen el personal y clientes encontrando que no existe una definición estándar en la gestión pedidos de dichas cafeterías. También se pudieron identificar inconformidades con la idea de menús digitalizados por el miedo a la migración en temas de usabilidad y de puntos de quiebre aún por encima de los costos que puede implicar la infraestructura tecnológica.

La solución se ajusta para ser usada como un estándar en los servicios de cafeterías bares y restaurantes que disponen de cartas fijas, entrega de comprobantes de venta y un nivel de consumo medio-alto.

7.2. Objetivo 2: Implementar la arquitectura de la solución informática mediante el uso de tecnologías JavaScript y la metodología ICONIX.

A nivel de tecnologías, la elección fue adecuada considerando el tiempo que se tenía para el desarrollo del proyecto, se confió en tecnologías conocidas y con las que se había trabajado previamente reduciendo en tiempo de aprendizaje. En el diseño de las interfaces, al tratarse de un diseño ligero y minimalista la utilización de una librería de componentes o marco de trabajo de CSS, tal como se define DaisyUI agilizó la construcción de vistas y el uso de Next.js redujo el tiempo invertido para la construcción de páginas y rutas.

Sin dejar de mencionar que la utilización del orquestador Kubernetes ofreció el espacio ideal donde se desarrolló el proyecto. Se construyeron imágenes distintas de Docker, aun cuando era posible no hacer esto, con el objetivo de separar totalmente conceptos y más bien, a posterior, manejar dentro de la misma imagen otros recursos que puedan ser vitales para el servicio como la configuración de un volumen persistente de almacenamiento.

Como resultado del objetivo 2, no solo se construyeron 3 servicios y las interfaces desde donde sean ejecutadas las peticiones, sino que también se buscó dar una alternativa que promueva una migración hacia el uso de tecnologías informáticas en las cafeterías tal como lo hacen otros locales a niveles internacionales como Eatsa o Ihop. Pero, a nivel particular, aplicar esta solución dentro de las instalaciones universitarias como una solución ligera que no pierda de vista la necesidad de la interacción humana.

Desde un inicio, la población más sensible a la utilización del sistema han sido personas mayores que prefieren la atención directa con camareros por temor al uso de tecnologías y personas activistas de la seguridad informática y privacidad de datos. Dada esa situación, al diseñar el sistema fue necesario considerar que no siempre el cliente será quien haga el pedido, sino que el rol de mesero debe tener la opción habilitada.

7.3. Objetivo 3: Realizar pruebas de funcionamiento del software.

Una vez aplicadas las pruebas de funcionamiento y dada la naturaleza del sistema, se evidenció una reducción entre la interacción mesero-cliente, siendo de ayuda a mantener el distanciamiento social propiciado por la pandemia de COVID-19 y/o por otras enfermedades de transmisión aérea. La solución informática responde a la necesidad de cafeterías de presentar sus platos y bebidas de forma simple y amigable, así como de ser un medio por el cual los clientes y meseros realicen sus pedidos, reduciendo el tiempo empleado en este proceso como:

- a) La toma de pedidos pasando de alrededor de 20 minutos en cafeterías de la ciudad y de 13 minutos en cafeterías universitarias a 6 minutos en la aplicación web.

- b) El registro/actualización de productos y menú en el modelo tradicional tardan más de una hora y como mínimo 13 minutos frente a 8 minutos en la solución informática.
- c) El cobro por consumo que tarda alrededor de 15 minutos en locales de la ciudad con respecto a los 6 minutos que se los realiza en la aplicación web.

Para el último caso, en la cafetería de universitaria no representa tiempo empleado dado que en su gestión no incluye la entrega de comprobantes por consumo, no acepta pagos por transferencias y no permite que el pago sea realizado luego de consumido el producto, sino que debe ser cancelado el valor del pedido al momento de hacerlo.

Para los clientes de las cafeterías representa una mejora en la agilidad de la atención, teniendo un grado de aceptación del 97.14% entre audiencias de diversas edades y condiciones socioeconómicas.

En base a este análisis se refleja que la solución informática desarrollada a través de arquitectura de microservicios y/o serverless en el presente TT facilita a los negocios relacionados con las cafeterías y/o restaurantes en la gestión de sus menús, toma de pedidos y cobros, a través de la reducción de los tiempos empleados a la atención del cliente y además, se mejora la organización y comunicación de sus actividades en sus procesos; con lo cual se da contestación satisfactoriamente a la pregunta de investigación de presente TT: “¿Cómo gestionar el proceso de toma de pedidos en los locales físicos de Cafeterías?”.

8. Conclusiones

Una vez terminado el trabajo de titulación, se concluye lo siguiente:

- El sistema para la toma de pedidos en cafeterías utilizando arquitectura de microservicios ofrece una alternativa estandarizada o común que puede ser usada por cualquier cafetería dentro y fuera de la Universidad Nacional de Loja. Además, también puede ser usada por restaurantes y bares en la gestión de sus pedidos favoreciendo el distanciamiento social y administración adecuada de su negocio.
- La utilización de la metodología ICONIX favorece la documentación de proyectos a través de 3 etapas dedicadas al análisis y diseño mediante modelo de requerimientos, caso de uso, secuencia y arquitectura, así como la construcción, corrección y mejora iterativa e incremental ofreciendo flexibilidad al cambio y entregas continuas.
- Una arquitectura basada en microservicios facilita la construcción de sistemas informáticos viendo la solución en pequeñas partes independientes que pueden comunicarse entre sí y facilitando el mantenimiento a través de la actualización de módulos o integración con nuevos servicios. Así mismo facilita la asignación de recursos computacionales bajo demanda y necesidad.
- Las pruebas de software con usuarios reales, aunque en escenarios simulados, es una retroalimentación para el sistema que ofrece perspectivas más diversas del funcionamiento que se espera, así mismo sirve para identificar el grado de cumplimiento que tiene la solución con las necesidades identificadas.
- La implementación de la solución informática en cafeterías o restaurantes representa una reducción en el tiempo empleado para la gestión de sus procesos, además, facilita la atención a los clientes facultándoles de la posibilidad de contar con un menú interactivo, realizar sus pedidos y ver el estado de los mismos.

9. Recomendaciones

Una vez finalizado el Trabajo de Titulación se recomienda lo siguiente:

1. Implementar esta solución dentro de las cafeterías universitarias, empezando con un caso piloto para evaluar la disposición de los usuarios con la misma.
2. Evaluar la necesidad de crear una versión móvil del presente proyecto considerando que el modelo actual es una aplicación web responsiva.
3. Mantener el uso de tecnologías como Next.js que faciliten gestión de rutas y construcción de aplicaciones web reduciendo tiempo de trabajo. Así como el uso de tecnologías de administración de estilos como TailwindCSS.
4. La presente aplicación se puede implementar en restaurantes debido a que los procesos que gestiona esta aplicación se realizan de forma estándar.

9.1. Trabajos Futuros

Se recomiendan los siguientes trabajos futuros:

- Integración del sistema de toma de pedidos en cafeterías con un módulo de facturación electrónica para que en lugar de comprobantes de venta se pueda generar facturas con valor tributario.
- Agregar como método de inicio de sesión el servicio basado en cuentas sociales como Facebook o Google para mayor facilidad y agilidad.
- Implementar el módulo de pedidos a domicilio con comunicación a servicio de delivery para ampliar el rango de atención a los clientes.
- Integrar una pasarela de pago como un método más para el cobro por consumo. Evaluando el costo por transacción y seguridad de las mismas.

10. Bibliografía

- [1] P. González Fernández, “Implantación de un SISTEMA ERP en una PyME”, 2015.
- [2] M. Lowe, R. Byron, G. Ferry, y M. Garcia, “Food for Thought: Frequent Interracial Dining Experiences as a Predictor of Students’ Racial Climate Perceptions”, *J Higher Educ*, vol. 84, núm. 4, pp. 569–600, jul. 2013, doi: 10.1080/00221546.2013.11777302.
- [3] P. Brewer y A. Sebby, “The effect of online restaurant menus on consumers’ purchase intentions during the COVID-19 pandemic”, *Int J Hosp Manag*, vol. 94, p. 102777, 2021, doi: 10.1016/j.ijhm.2020.102777.
- [4] M. Jeong, K. Kim, F. Ma, y R. DiPietro, “Key factors driving customers’ restaurant dining behavior during the COVID-19 pandemic”, *International Journal of Contemporary Hospitality Management*, vol. 34, núm. 2, pp. 836–858, ene. 2022, doi: 10.1108/IJCHM-07-2021-0831/FULL/HTML.
- [5] S. Saeed, N. Jhanjhi, M. Naqvi, y M. Humayun, “Analysis of software development methodologies”, *International Journal of Computing and Digital Systems*, vol. 8, núm. 5, pp. 445–460, 2019, doi: 10.12785/IJCDS/080502.
- [6] A. El-Abbassy y M. El-Zeweidy, “ICONIX Approach to MVC: Applying Robustness Analysis on the Model-View-Controller Architecture”, *Journal of the ACS*, vol. 7, 2013.
- [7] D. Rosenberg, M. Stephens, y M. Apress, “Agile development with ICONIX process”, *Springer*, 2005, Consultado: el 14 de agosto de 2023. [En línea]. Disponible en: <https://link.springer.com/content/pdf/10.1007/978-1-4302-0009-3.pdf>
- [8] B. Thakare, B. Bhokse, y L. Thakare, “Deriving Best Practices from Development Methodology Base (Part 2)”, Consultado: el 14 de agosto de 2023. [En línea]. Disponible en: www.ijert.org
- [9] T. Rozman, R. Horvat, y I. Rozman, “Using ICONIX unified object modelling approach for software tools development”, 2000, Consultado: el 14 de agosto de 2023. [En línea]. Disponible en: <https://www.researchgate.net/publication/280521255>
- [10] “Agile Planning”, *Agile Development with ICONIX Process*, pp. 173–188, nov. 2005, doi: 10.1007/978-1-4302-0009-3_9.
- [11] A. Ranjan, A. Sinha, y R. Battewad, “JavaScript for modern web development: building a web application using HTML, CSS, and JavaScript”, p. 414, 2020, Consultado: el 14 de agosto de 2023. [En línea]. Disponible en: https://books.google.com/books/about/JavaScript_for_Modern_Web_Development.html?id=b2bdDwAAQBAJ
- [12] Stack Overflow, “Stack Overflow Developer Survey 2021”. Consultado: el 12 de mayo de 2022. [En línea]. Disponible en: <https://insights.stackoverflow.com/survey/2021#most-popular-technologies-language-prof>
- [13] M. Vorobyov, “JavaScript - Reviews, Pros & Cons | Companies using JavaScript”. Consultado: el 12 de mayo de 2022. [En línea]. Disponible en: <https://stackshare.io/javascript>
- [14] K. Sun y S. Ryu, “Analysis of JavaScript programs: Challenges and research trends”, *ACM Comput Surv*, vol. 50, núm. 4, ago. 2017, doi: 10.1145/3106741.

- [15] A. Nandaa, *Beginning API Development with Node.js: Build Highly Scalable, Developer-Friendly APIs for the Modern Web with JavaScript and Node.js*. Packt Publishing Ltd, 2018.
- [16] M. Valarezo Pardo, J. Honores Tapia, A. Gómez Moreno, y L. Vincés Sánchez, “Comparación de tendencias tecnológicas en aplicaciones web”, 2018.
- [17] S. Hoque, *Full-Stack React projects: build modern web applications using the MERN stack*. 2020. Consultado: el 14 de agosto de 2023. [En línea]. Disponible en: https://books.google.com/books/about/Full_Stack_React_Projects.html?id=097dDwAAQBAJ
- [18] B. Chinthanet *et al.*, “What makes a good Node.js package? Investigating Users, Contributors, and Runnability”, *ACM Trans. Softw. Eng. Methodol.* XX, p. 20, jun. 2021, doi: 10.1145/1122445.1122456.
- [19] A. Mardan, “Using Express.js to Create Node.js Web Apps”, *Practical Node.js*, pp. 51–87, 2018, doi: 10.1007/978-1-4842-3039-8_2.
- [20] A. Mardan, “Publishing Node.js Modules and Contributing to Open Source”, *Practical Node.js*, pp. 261–267, 2014, doi: 10.1007/978-1-4302-6596-2_12.
- [21] A. Karnauch, “Developer Reputation Estimator: Increasing the Transparency of Developer Contributions in Open Source Software”, *Masters Theses*, may 2020, Consultado: el 14 de agosto de 2023. [En línea]. Disponible en: https://trace.tennessee.edu/utk_gradthes/5615
- [22] I. Kainu, “Optimization in React.js: Methods, Tools, and Techniques to Improve Performance of Modern Web Applications”, may 2022, Consultado: el 14 de agosto de 2023. [En línea]. Disponible en: <https://trepo.tuni.fi/handle/10024/140258>
- [23] M. Fariz, S. Lazuardy, y D. Anggraini, “Modern Front End Web Architectures with React.Js and Next.Js”, *International Research Journal of Advanced Engineering and Science*, vol. 7, núm. 1, pp. 132–141, 2022.
- [24] W. Kithulwatta, W. Wickramaarachchi, K. Jayasena, B. Kumara, y R. Rathnayaka, “Adoption of Docker Containers as an Infrastructure for Deploying Software Applications: A Review”, pp. 247–259, 2022, doi: 10.1007/978-981-16-5559-3_21.
- [25] A. Potdar, D. Narayan, S. Kengond, y M. Mulla, “Performance Evaluation of Docker Container and Virtual Machine”, *Procedia Comput Sci*, vol. 171, pp. 1419–1428, ene. 2020, doi: 10.1016/J.PROCS.2020.04.152.
- [26] S. Buchanan, J. Rangama, y N. Bellavance, “Introducing Azure Kubernetes Service”, *Introducing Azure Kubernetes Service*, 2020, doi: 10.1007/978-1-4842-5519-3.
- [27] C. Miyachi, “The Rise of Kubernetes”, *2021 Cloud Continuum*, pp. 1–5, nov. 2021, doi: 10.1109/CLOUDCONTINUUM54760.2021.00002.
- [28] B. Johansson, M. Ragberger, T. Nolte, y A. V. Papadopoulos, “Kubernetes Orchestration of High Availability Distributed Control Systems”, *Proceedings of the IEEE International Conference on Industrial Technology*, vol. 2022-August, 2022, doi: 10.1109/ICIT48603.2022.10002757.
- [29] S. Hashemipour y M. Ali, “Amazon web services (aws) – an overview of the on-demand cloud computing platform”, *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 332 LNICST, pp. 40–47, 2020, doi: 10.1007/978-3-030-60036-5_3/COVER.

- [30] S. Ifrah, "Getting Started with Containers on Amazon AWS", *Deploy Containers on AWS*, pp. 1–40, 2019, doi: 10.1007/978-1-4842-5101-0_1.
- [31] S. Ifrah, "Deploy Containerized Applications with Azure Kubernetes Service (AKS)", *Getting Started with Containers in Azure*, pp. 97–128, 2020, doi: 10.1007/978-1-4842-5753-1_4.
- [32] H. Chawla y H. Kathuria, "Azure Kubernetes Service", *Building Microservices Applications on Microsoft Azure*, pp. 151–177, 2019, doi: 10.1007/978-1-4842-4828-7_5.
- [33] P. Ransara, A. Dayananda, P. Seewali, y D. Gunarathna, "The Library Automation through the Cloud Computing server: Based on Digital Ocean Cloud server", 2021, Consultado: el 14 de agosto de 2023. [En línea]. Disponible en: <http://repository.kln.ac.lk/handle/123456789/24464>
- [34] V. Pikkuhookana y J. Soini, "Software development using cloud services", 2023, Consultado: el 14 de agosto de 2023. [En línea]. Disponible en: <http://www.theseus.fi/handle/10024/797159>
- [35] R. Tasnim, A. Mim, S. Mim, y I. Jabiullah, "A Comparative Study On Three Selective Cloud Providers", *International Journal on Cybernetics & Informatics*, vol. 11, núm. 4, pp. 167–178, ago. 2022, doi: 10.5121/ijci.2022.110413.
- [36] Menulux, "Menulux Digital Tablet Menu and Restaurant POS Systems". Consultado: el 14 de agosto de 2023. [En línea]. Disponible en: <https://www.menulux.com/en/about>
- [37] A. Joshi, S. Kale, S. Chandel, y D. Pal, "Likert Scale: Explored and Explained", *Br J Appl Sci Technol*, vol. BJASt, núm. 4, p. 157, 2015, doi: 10.9734/BJAST/2015/14975.

11. Anexos

Anexo 1. Entrevistas para la obtención de requerimientos

La selección de cafeterías para la muestra se tomó en base al predio municipal concorde abril de 2022. El predio municipal es el siguiente:

Nombre	Dirección
Alebrije	Lourdes 01-03 y Av. 24 De Mayo
Artesanale	10 de Agosto 14-80 y Bolivar
Artesanale Suc. 1	18 de Noviembre y Gobernacion de Mainas
Artesanale Suc. 3	Prolongacion 24 de Mayo y Av. Zoilo Rodriguez
Bistro Aleman Lecka	24 de Mayo 207-39 Azuay y Miguel Riofrio
Cafe Bourbon Ol Roastery	Av. Orillas Del Zamora 101-18 y Virgilio Abarca
Cafe de Aquiles	Bolivar 13-13 Lourdes
Cafe de Matilde	Sucre y José Antonio Eguiguren
Cafecito Lojano El	18 de Noviembre 219-28 y Cariamanga
Cafecito Lojano Suc. El	Olmedo E/ 10 de Agosto y Jose A. Eguiguren
Cafeteria Indera	18 de Noviembre 201-32 10 de Agosto
Cafeteria y Heladeria Disert Coffec	Bolivar Sn 10 de Agosto
Chocosweet	Av. Orillas Del Zamora y Prol. 24 de Mayo (Boulevard Del Rio)
Chocosweet Suc.	24 de Mayo 201-31 y Jose Antonio Eguiguren
Crepella	Juan José Peña y Colón
Crepes & Coffee	Zoilo Rodriguez y Antisana
Emporio Lojano Desde 2012 Ricemacgo	10 de Agosto 14-30 E/ Sucre y Bolívar
Francis	Celica 152-18 E/ Av. Universitaria y 18 de Nov.
Gina'S Cafe de la Casa Suc.	Simon Bolivar 201-19 201-19 Jose Antonio Eguiguren
Ginas Cafe de la Casa	Bolivar 07-77 10 de Agosto
Gollerias	24 de Mayo 213-15 Entre Lourdes y Leopoldo Palacios
Hijole Empanadas	Pio Jaramillo Alvarado Sn Venezuela
Kaweh Coffee Shop	Mercadillo 158-14 Bolivar
La Fresca Juice Bar	Bolivar Sn Mercadillo
La Mojigata	24 de Mayo E/10 de Agosto y Rocafuerte
Layseca's	Sucre S/N Diego Vaca de Vega
Linburguer	10 de Agosto 06-08 y Juan J Pena
Moka Coffee House	Rocafuerte 13-57 E/ Bolívar y Bernardo Valdivieso

Montbel	Bolivar y 201-25 10 de Agosto
Natural Yogurt	Bolívar y Diego Vaca de Vega
Neptali Cafe	Av. Orillas Del Zamora 05-66 y Prol. 24 de Mayo
Notas de Cafe	Bolivar y Rocafuerte (Portal de Santo Domingo)
Omm Siri	Ramon Pinto 11-119 y Jose Picoita
Philadelphia 98	24 de Mayo Sn Miguel Riofrio
Ricuras de Sal y Dulce	Olmedo 277-21 Juan Jose Peña
Shawarma	24 de Mayo Sn Rocafuerte
Tamal Lojano	18 de Noviembre 05-12 E Imbabura
Timothy's	Diego Vaca de Vega y Eterna Juventud
Topoli	Bolivar 13-78 y Miguel Riofrio
Villonaco	Vía Antigua A Catamayo (Central Eólica Villonaco)
Waffle Planet	24 de Mayo y Rocafuerte

Acta de Entrevista					
Código:	E-CCL-04		Versión:	1	
Fecha:	Mayo 20, 2022	Hora Inicio:	17h00	Hora Fin:	18h00
Lugar:	Ciudad de Loja Rincón de Borgoña Restaurant y Casa de Banquetes				
Entrevistador:	Maria Encalada Córdova				
Entrevistada/o	Luis Alvear				

DESCRIPCIÓN

<p>Introducción</p> <p>La presente entrevista tiene el objetivo de conocer de forma general en funcionamiento de cafeterías en la ciudad de Loja y los nuevos modelos adoptados con la llegada de la pandemia del COVID-19</p>
<p>Desarrollo</p> <p>¿Cómo inicia su interacción con los visitantes? Cuando llegan los clientes se diferencian entre 2 tipos de visitantes:</p>

Clientes con reservación: Para ellos, dado que hay una reservación previa, se les da la bienvenida y se comienza a preparar el pedido, esto no suele tardar mucho ya que cuando hay reservaciones, parte de los alimentos necesarios se pre-cosen o se preparan de forma parcial para reducir los tiempos de espera.

Clientes sin reservación: Se les acerca la carta para que puedan revisar los platos y bebidas que hay disponibles.

¿Cuánto tiempo en promedio dedica a la atención de los clientes antes de tener la orden definida?

El tiempo de atención suele estar entre 10 y 20 minutos. Una de las características que se ha tratado de destacar dentro del local, es la importancia del tiempo de los clientes.

¿Qué medidas de bioseguridad se implementaron en su local a partir de la pandemia?

Al inicio de la pandemia se cerró el local, tal como lo dispuso el COE nacional y local, se mantuvo así hasta inicios del 2021. Durante el confinamiento se trató de mitigar el impacto de perder su fuente de ingresos, sin embargo, con la reapertura el trabajo que se realizó es mucho mejor gestionado.

¿Cuánto varía su menú de un día a otro?

Los platos del menú no están preparados, siempre se cuecen cuando hay demanda, por lo que siempre están disponibles. En ocasiones especiales, se agregan productos especiales con el fin de ofrecer un mejor servicio, estos son promocionados a través de redes sociales (Página en facebook: Rincón de Borgoña Restaurant y Casa de Banquetes), pero en modo normal, se mantiene el menú todos los días. En las fechas especiales se activan promociones, como por ejemplo en el día de la madre.

¿Cómo cree que puede mejorar sus servicios considerando la situación de pandemia que vivimos?

Respetando las normas de bioseguridad, respondiendo a las recomendaciones del COE nacional y local. Adicionalmente, se puede mejorar con una reducción del contacto entre personal y clientes, aunque las restricciones son menores, la prioridad siempre es la seguridad sanitaria e integral de colaboradores y visitantes.

En ocasiones, al momento de cancelar por consumo, se tiene a varios clientes esperando por el mismo fin, en estos y todos los casos, se trata de agilizar la atención y reducir los tiempos de espera.

Observaciones (Opcional)

Acta de Entrevista					
Código:	E-CCL-04		Versión:	1	
Fecha:	Mayo 20, 2022	Hora Inicio:	16h00	Hora Fin:	17h00
Lugar:	Ciudad de Loja Cafetería La Llorona				
Entrevistador:	Maria Encalada Córdova				
Entrevistada/o					

DESCRIPCIÓN

<p>Introducción</p> <p>La presente entrevista tiene el objetivo de conocer de forma general en funcionamiento de cafeterías en la ciudad de Loja y los nuevos modelos adoptados con la llegada de la pandemia del COVID-19</p>
<p>Desarrollo</p> <p>¿Cómo inicia su interacción con los visitantes? Cuando llegan los clientes alguien del equipo de colaboradores se acerca, comparte la carta de la casa y se retira con el fin de dar tiempo a revisar y decidir.</p> <p>¿Cuánto tiempo en promedio dedica a la atención de los clientes antes de tener la orden definida? El tiempo de atención suele estar entre 10 minutos. En algunas ocasiones se hacen recomendaciones y se dan indicaciones ante cualquier duda que puedan tener los clientes.</p> <p>¿Qué medidas de bioseguridad se implementaron en su local a partir de la pandemia? El local se abrió en el 2021, quizá en una etapa de transición entre lo que fue la oleada de casos COVID-19 y la reactivación económica cobijada por la esperanza de reducción de contagios. Las medidas que se tomaron fueron y siguen siendo las recomendadas por el COE nacional y local, adicionalmente, se implementan protocolos de seguridad internos con el fin de asegurar tanto a colaboradores como a visitantes.</p> <p>¿Cuánto varía su menú de un día a otro? Los platos del menú no están preparados, siempre se cuecen cuando hay demanda, por lo que siempre están disponibles. En ocasiones especiales, se agregan productos especiales con el fin de ofrecer un mejor servicio, estos son promocionados a través de redes sociales (Página en facebook: Cafeteria La Llorona), pero en modo normal, se mantiene el menú todos los días.</p> <p>¿Cómo cree que puede mejorar sus servicios considerando la situación de pandemia</p>

que vivimos?

Respetando las normas de bioseguridad, respondiendo a las recomendaciones del COE nacional y local. Adicionalmente, se puede mejorar con una reducción del contacto entre personal y clientes, aunque las restricciones son menores, la prioridad siempre es la seguridad sanitaria e integral de colaboradores y visitantes.

En ocasiones, al momento de cancelar por consumo, se tiene a varios clientes esperando por el mismo fin, en estos, se trata de mantener la calma del equipo de apoyo y, sobre todo, la buena atención al cliente.

Observaciones (Opcional)**Acta de Entrevista**

Código:	E-CCL-02		Versión:	1	
Fecha:	Mayo 16, 2022	Hora Inicio:	15h00	Hora Fin:	16h00
Lugar:	Ciudad de Loja Cafetería Emporio				
Entrevistador:	Maria Encalada Córdova				
Entrevistada/o	Rita Macas				

DESCRIPCIÓN**Introducción**

La presente entrevista tiene el objetivo de conocer de forma general en funcionamiento de cafeterías en la ciudad de Loja y los nuevos modelos adoptados con la llegada de la pandemia del COVID-19

Desarrollo**¿Cómo inicia su interacción con los visitantes?**

Cuando llegan los clientes se diferencia entre 2 tipos de visitantes:

Clientes con reservación: Para ellos, dado que hay una reservación previa, se les da la bienvenida y se comienza a preparar el pedido, esto no suele tardar mucho ya que cuando hay reservaciones, parte de los alimentos necesarios se pre-cosen o se preparan de forma parcial para reducir los tiempos de espera.

Clientes sin reservación: Se les acerca la carta para que puedan revisar los platos y bebidas que hay disponibles.

¿Cuánto tiempo en promedio dedica a la atención de los clientes antes de tener la orden definida?

El tiempo de atención suele estar entre 10 y 20 minutos. Se hace la asignación de mesas, se resuelven dudas en caso de existir y se da un tiempo prudente para que el cliente pueda ver la carta y decidir que desea. Si fuera reservación, se tendría una mesa asignada y se procedería a preparar los alimentos que constan en la reservación.

¿Qué medidas de bioseguridad se implementaron en su local a partir de la pandemia?

Las medidas dispuestas por el COE nacional y local, dando vital importancia al cuidado del equipo y visitantes.

¿Cuánto varía su menú de un día a otro?

Los platos del menú no están preparados, siempre se cuecen cuando hay demanda, por lo que siempre están disponibles. En ocasiones especiales, se agregan productos especiales con el fin de ofrecer un mejor servicio, estos son promocionados a través de redes sociales (Página en facebook: [Emporio Lojano](#)), pero en modo normal, se mantiene el menú todos los días.

¿Cómo cree que puede mejorar sus servicios considerando la situación de pandemia que vivimos?

Respetando las normas de bioseguridad, respondiendo a las recomendaciones del COE nacional y local. Adicionalmente, se puede mejorar con una reducción del contacto entre personal y clientes, aunque las restricciones son menores, la prioridad siempre es la seguridad sanitaria e integral de colaboradores y visitantes.

Observaciones (Opcional)

Acta de Entrevista					
Código:	E-CCL-03		Versión:	1	
Fecha:	Mayo 18, 2022	Hora Inicio:	10h00	Hora Fin:	11h00
Lugar:	Ciudad de Loja Cafetería Universitaria de Carrera de Turismo - Universidad Nacional de Loja				
Entrevistador:	Maria Encalada Córdova				
Entrevistada/o	Diana Valdiviezo				

DESCRIPCIÓN

Introducción

La presente entrevista tiene el objetivo de conocer de forma general en funcionamiento de cafeterías en la ciudad de Loja y los nuevos modelos adoptados con la llegada de la pandemia del COVID-19

Desarrollo

¿Cómo inicia su interacción con los visitantes?

Cuando llegan los clientes alguien del equipo de colaboradores se acerca, comparte la carta de la casa y se retira con el fin de dar tiempo a revisar y decidir.

¿Cuánto tiempo en promedio dedica a la atención de los clientes antes de tener la orden definida?

El tiempo de atención suele estar entre 10 minutos. En algunas ocasiones se hacen recomendaciones y se dan indicaciones ante cualquier duda que puedan tener los clientes.

¿Qué medidas de bioseguridad se implementaron en su local a partir de la pandemia?

El local se abrió en el 2021, quizá en una etapa de transición entre lo que fue la oleada de casos COVID-19 y la reactivación económica cobijada por la esperanza de reducción de contagios. Las medidas que se tomaron fueron y siguen siendo las recomendadas por el COE nacional y local, adicionalmente, se implementan protocolos de seguridad internos con el fin de asegurar tanto a colaboradores como a visitantes.

¿Cuánto varía su menú de un día a otro?

Los platos del menú cafetería, se mantienen todos los días, los productos que se sirven en almuerzos cambian de un día a otro. Estos platos se repiten en diferentes días de la semana, pero de un día a otro, siempre se trata de servir platos distintos.

¿Cómo cree que puede mejorar sus servicios considerando la situación de pandemia que vivimos?

Respetando las normas de bioseguridad, respondiendo a las recomendaciones del COE nacional y local. Adicionalmente, se puede mejorar con una reducción del contacto entre personal y clientes, aunque las restricciones son menores, la prioridad siempre es la seguridad sanitaria e integral de colaboradores y visitantes.

En ocasiones, al momento de cancelar por consumo, se tiene a varios clientes esperando por el mismo fin, en estos, se trata de mantener la calma del equipo de apoyo y, sobre todo, la buena atención al cliente, pero hay momentos de mayor actividad.

Observaciones (Opcional)

Anexo 2. Menús de Cafeterías

Emporio Lojano

Delicioso, seguro y especial



Café de especialidad

Café especial de origen emporio lojano

8oz

Somos apasionados por el café, lo producimos y servimos

1.75 \$

Café especial de origen emporio lojano

12oz

Disfruta en grande, el tradicional Jarro de café de especialidad

2.00 \$

Latte macchiato

Leche cremada con un toque de café de especialidad

2.00 \$

Ice latte macchiato

Leche cremada con un toque de café de especialidad

3.50 \$

Café espresso

Disfruta le esencia del café lojano

2.25 \$

Cappuccino especial emporio

Base de café de especialidad, leche cremada y un toque de canela

2.75 \$

Mocaccino especial emporio

Base de café de especialidad, chocolate fino de aroma y leche cremada

3.25 \$

Chocolate

Chocolate fino de aroma en leche con un toque de canela

2.00 \$

Té/Agua aromática

Infusión de selección de hierbas lojanas secas a su gusto: horchata, hierba luisa, toronjil, canela, menta, etc

1.25 \$



Aromática e infusiones

Nuestra pasión es servirle, siéntase libre de pedirnos cualquier variación en sus bebidas

Horchata natural caliente

tradicional bebida lojana con selección de hierbas frescas y un toque de limón

1.75 \$

Emporio Lojano

Delicioso, seguro y especial



Jugos, batidos y granizados

Jugo de fruta natural

Con o sin azúcar. Elije entre tomate de árbol, mora, fresa o la fruta de temporada 1.75 \$

Batido de fruta natural

Con o sin azúcar. Tu fruta favorita batida con leche 2.25 \$

Granizado de café

Único del emporio con café de especialidad 2.75 \$

Granizado de fruta

Tu fruta favorita granizada 2.75 \$

Granizado de horchata

Granizado de nuestra exquisita horchata de hierbas frescas 2.75 \$



Especiales

Affogatto matilde

Doble bola de helado de vainilla + espresso + bombón de chocolate 4.75 \$

Frappuccino

Helado de vainilla + espresso + leche + jarabe de chocolate 3.75 \$



Inolvidable

Café con ron

3.75 \$

Café amareto

3.75 \$

Café turbo (Whisky)

4.00 \$

Cappuccino amareto

4.00 \$

Mojito de café de especialidad

5.00 \$

Cerveza personal

2.50 \$

Michelada especial de la casa

3.50 \$

Copa de vino especial de la casa

3.75 \$

Botella de vino especial de la casa

18.00 \$

Nuestra pasión es servirle,
siéntase libre de pedirnos
cualquier variación en sus
bebidas

Emporio Lojano

Delicioso, seguro y especial



Desayunos

Desayuno catalina

Huevos revueltos con jamón y tocino, tostadas, jugo del día y café especial negro o con leche 3.75 \$

Desayuno especial emporio lojano

Huevos revueltos, jugo del día, tamal o humita y café especial negro o con leche 3.75 \$



Emporio

Empapizza original

Exquisita empanada gigante rellena de full queso, jamón, salami y un toque de oregano 3.50 \$

Empanada full queso ó queso-cebolla

Famosas del emporio. Full queso o queso cebolla. 1.25 \$

Bolón salvaje (original del emporio)

Bolón gigante de chicharrón, con huevo frito y tocino 3.00 \$

Tigrillo

Majado de verde con queso, chicharrón y huevo frito 3.00 \$

Chifle Bencho

Único del emporio. Chifle de guineo revueltos con huevo y queso. Tocino montado. 3.50 \$

EmpanaZA de verde

Empanada gigante de verde full relleno de pollo. 2.75 \$

Mote con chicharrón

3.00 \$

Mote o Yuca pilla (revuelto huevo y queso) 3.00 \$

Nuestra pasión es servirle,
siéntase libre de pedirnos
cualquier variación en sus
comidas

Emporio Lojano

Delicioso, seguro y especial



Lojano

Tamal lojano

Exquisita tradición lojana a base de maíz relleno de pollo 1.50 \$

Humita lojana

Exquisita tradición lojana a base de maíz tierno amasado y rellena de queso 1.50 \$

Cecina + mote pillo

La medida justa para un entre día tradicional lojano. 5.50 \$

Cecina + yuca pilla

La medida justa para un entre día tradicional lojano. 5.50 \$

Nano cecina

Cecina lojana en la porción justa para disfrutarla. 3.50 \$



Vegetariano

Empapizza vegetariana

Exquisita empanada gigante rellena de full queso, cebolla, pimiento, aceitunas y champiñones con un toque de oregano 5.50 \$

Ensalada emporio

Exquisita combinación de lechuga, tomate fresco, zanahoria, aceitunas, queso en cubitos y nueces. Vinagreta de la casa. 5.00 \$

Nuestra pasión es servirle,
siéntase libre de pedirnos
cualquier variación en sus
bebidas

Emporio Lojano

Delicioso, seguro y especial



Postres

Tiramisu	3.00 \$
Torta moja de chocolate	3.00 \$
Miel con queso	2.50 \$



Para tu crush

Super papa	4.50 \$
Papas a la francesa recién frita con salchicha vienesa especial ó chorizo especial ó filete de pollo y ensalada	
Super papa EMPO con cecina	6.00 \$
Papas a la francesa recién frita con lomo de cerdo a la plancha y ensalada	
Alitas BBQ	7.00 \$
Papas a la francesa recién frita con alitas BBQ (6 porciones)	
Sanduche de Embutidos	4.50 \$
Único del emporio. Jamón de pierna, queso, salami, jugosos champiñones y aceitunas. Incluye papa frita a la francesa	

Nuestra pasión es servirle,
siéntase libre de pedirnos
cualquier variación en sus
bebidas

Emporio Lojano

Delicioso, seguro y especial

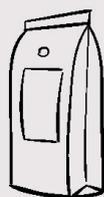


Eventos

Tu mejor momento con nosotros, contamos con salas de celebraciones y cenas exclusivas para ti, tu familia y amigos.

Desde 10 \$

Disfruta tranquilo hasta sin prisas a gusto con tu familia y amigos.



Tienda

Comparte con tu familia nuestro café de especialidad origen papalango.

La finca JoseLita es uno de los últimos rezagos de bosque nativo de Pindal al pie del Cerro Misama, aquí el café crece entre árboles frutales y maderables que aportan aroma y dulzor.

En papalango, el café recarga su espíritu en invierno y da sus mejores frutos en Agosto de cada año, luego de un largo proceso de cultivo tradicional a mano, sin afectaciones al medio ambiente.

Desde 7.5 \$

Tal y como siempre se ha hecho, el café se tiende al sol y de él absorbe energía y sabor sin iguales.

Recien tostado y molido.

Nuestra pasión es servirte.
Nuestras salas son exclusivas,
espaciosas, cómodas y elegantes.

Artesanale

Tradición y calidad para su deleite



0980778413



07 2570050



0998747403



Artesanale loja



Típicos



Bolón de chicharón.....	\$ 1.00
Bolón de queso.....	\$ 1.00
Empanadas de verde.....	\$ 1.35
Empanadas chilenas.....	\$ 1.25
Empanadas de queso.....	\$ 0.70
Tamales.....	\$ 1.00
Humitas.....	\$ 1.00
Quimbolitos.....	\$ 1.00

Combos

Combo tigrillo..... \$ 4.50
+ cecina + huevo + ensalada + café.

Combo bolón \$ 2.90
+ b. de chicharrón + b. queso + huevo + ensalada + café.

Agrandando tu combo

Queso y/o huevo..... \$ 0.50

Chicharrón..... \$ 1.00



Artesanale loja

Tradición y calidad para su deleite

Artesanale



0980778413

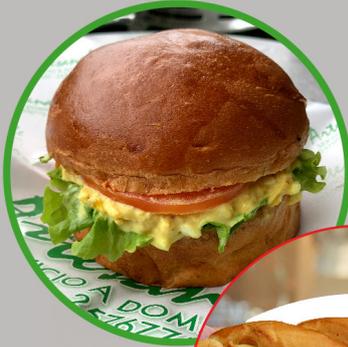
072570050

Sanduches

Entero

Medio

De pollo.....	\$ 2.15	
Parrillero.....	\$ 4.60	
<i>Carne, huevo, tomate y lechuga.</i>		
De jamón y queso.....	\$ 4.60	\$ 2.65
Artesanale	\$ 6.60	\$ 3.50
<i>Queso, jamón, pernil, salami, lechuga y tomate.</i>		
Steak sandwich.....	\$ 4.60.....	
<i>Carne, tocino, cebolla, lechuga y queso.</i>		



Prensados

Queso.....	\$ 1.30
Mixto.....	\$ 1.45
Pollo.....	\$ 1.80

Pasta y Lasagana

Lasagna de pollo.....	\$ 4.99
Pasta a la carbonara.....	\$ 4.99
<i>spaghetti, fettuccine, fusili, penne.</i>	
Pasta a la bolognesa.....	\$ 4.99
Pasta de pollo y champiñones...	\$ 4.99



Artesanale loja

Tradición y calidad para su deleite



Ensalada de frutas

Bandeja de frutas	\$ 2.80
Frutas + queso.....	\$ 4.20
Frutas + crema y helado.....	\$ 4.20
Frutas + granola y yogurt.....	\$ 3.40
Aderezo adicional	\$ 0.50

Gelato

Copa rossa.....	\$ 3.90
Tutta fruta.....	\$ 3.90
Moka chocolate.....	\$ 3.90
Banana Split.....	\$ 3.90
Copa o cono simple.....	\$ 1.10
Copa o cono doble.....	\$ 2.10
Copa o cono triple.....	\$ 3.10
Milkshake (18 sabores a elección).....	\$ 3.10





Postres



- Selva negra..... \$ 1.90
- Tres leches..... \$ 1.90
- Tres leches al maracuya..... \$ 2.00
- Tres chocolates..... \$ 1.90
- Chesse cake..... \$ 1.90
- Mousse mix chocolate..... \$ 1.90

Dulces

- Brazo gitano..... \$ 1.90
- Artesanale..... \$ 2.00
- Fresas con crema..... \$ 2.15
- Strawberry..... \$ 2.25
- Donuts..... \$ 0.90
- Brownies al gelato..... \$ 2.50



Artesanale Pizza



0980778413

072570050

Clasicas

	PERSONAL	PEQUEÑA	MEDIANA	GRANDE	EXTRA GRANDE
Margarita (salsa de tomate y queso mozzarella).....	\$ 2.00	\$ 4.25	\$ 7.70	\$ 11.40	\$ 16.20
Jamón.....	\$ 2.25	\$ 4.80	\$ 8.65	\$ 12.80	\$ 18.25
Pepperoni.....	\$ 2.50	\$ 5.35	\$ 9.60	\$ 14.20	\$ 20.25
Meatlovers (pepperoni, tocino y carne molida).....	\$ 2.75	\$ 5.85	\$ 10.60	\$ 15.60	\$ 22.25
Artesanale (Chorizo, jamón, salami, tocino, pollo, carne a la parrilla, champiñones y aceituna)....	\$ 3.00	\$ 6.40	\$ 11.50	\$ 17.00	\$ 24.30
Pollo.....	\$ 2.50	\$ 5.35	\$ 9.60	\$ 14.20	\$ 20.25
Parrillera (carne a la parrilla, chorizo, tocino salsa BBQ).....	\$ 2.50	\$ 5.35	\$ 9.60	\$ 14.20	\$ 20.25
Hawaiana (jamón, piña y duraznos).....	\$ 2.25	\$ 4.80	\$ 8.65	\$ 12.80	\$ 18.25
Champiñones.....	\$ 2.50	\$ 5.35	\$ 9.60	\$ 14.20	\$ 20.25
4 Estaciones (pollo, champiñones, jamón, chorizo y salami)....	\$ 3.25	\$ 6.95	\$ 12.50	\$ 18.50	\$ 26.30

Por ingrediente

1 Ingrediente.....	\$ 2.25	\$ 4.80	\$ 8.65	\$ 12.80	\$ 18.25
2 Ingredientes.....	\$ 2.50	\$ 5.35	\$ 9.60	\$ 14.20	\$ 20.25
3 Ingredientes.....	\$ 2.75	\$ 5.85	\$ 10.60	\$ 15.60	\$ 22.25

Por la compra de tu pizza lleva

Pan de ajo.....	\$ 1.00
Pan de ajo + gaseosa o té.....	\$ 1.65
Ensalada.....	\$ 1.00
Ensalada + gaseosa.....	\$ 1.65
Pan de ajo + ensalada + gaseosa o té.....	\$ 2.50

f Artesanale loja

Tradición y calidad para su deleite

Artesanale



0980778413



07 2570050

ENSALADAS

Primavera..... \$ 1.65

Lechuga, tomate, cebollas, queso parmesano, aceitunas y vinagreta.

Cesar..... \$ 4.50

Lechugas, pan tostado, huevo cocido, pollo, aceitunas, queso parmesano y salsa.

Pollo..... \$ 4.50

Lechugas, pollo, champiñones, queso parmesano y salsa.



CREPES

• Nutella crepe..... \$ 4.30

Nutella con banana y frutilla

• Frutichoc..... \$ 4.30

Helado de fresa, chocolate con frutillas y salsa dulce.

• Festival de frutas..... \$ 4.30

Helado de limon, cereza y frutas de la estación.

• Banana nuts..... \$ 4.30

Helado de manjar, almendras, frutos secos y bananos.

Crepes de sal

• Pollo y champiñones..... \$ 4.50

• Parrillero..... \$ 4.50

Carne a la parrilla, tocino y salsa BBQ.

• Crepizza..... \$ 4.50

Jamon, chorizo, carne, salami, tocino, queso y salsa de pizza



Artesanale loja

Tradición y calidad para su deleite

Artesanale

Bebidas



0980778413

072570050



Bebidas frias

- Tentación de café..... \$ 3.90
Helado de vainilla, espresso, licor y chantilly.
- Frozen cappuccino..... \$ 2.95
- Ice moka..... \$ 3.50
Helado de chocolate, espresso y crema batida.
- Te helado..... \$ 1.15
- Granizado de fresa o mora . \$ 1.50
- Malteada de chocolate..... \$ 2.15
- Jugos de fruta..... \$ 1.75
- Batidos..... \$ 2.10
- Zumo de fresa..... \$ 2.50
- Gaseosas..... \$ 0.70
- Coca cola light de 500 cc..... 1.10
- Agua embotellada..... \$ 0.65

Bebidas calientes

- Cubano (Café con ron)..... \$ 1.50
- Irishcoffe (Café Whisky)..... \$ 3.50
- Cappuccino..... \$ 1.70
- Mokaccino \$ 1.80
- Tinto..... \$ 0.70
- Café latte..... \$ 1.25
- Espresso..... \$ 1.25
- Chocolate..... \$ 1.80
- Café amore..... \$ 2.30
Espresso con crema y leche condensada
- Té, Horchatas
y Aguas aromaticas..... \$ 0.70
- Adicional Crema. Amareto,
Brandy, Vainilla \$ 0.50



Tradición y calidad para su deleite

Anexo 3. Modelado de la arquitectura lógica

Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless)

Modelado de la arquitectura lógica

Glosario de términos

El sistema debe ser capaz de reconocer los siguientes conceptos:

- **Ciente** el sistema debe permitir navegar por el menú, detalles de los platos y bebidas y realizar pedidos
- **Cocinero** el sistema debe presentar la lista de pedidos que los clientes realizan, así como procesar el estado de pendiente a en proceso y terminado
- **Mesero** para este perfil, el sistema muestra el detalle del consumo del cliente a través de un código de identificación, así como el total a pagar, además, es quien registra platos y/o bebidas en el menú y gestiona las categorías.
- **Administrador**, a este usuario, el sistema debe permitir gestionar perfiles de usuario cocina y caja, los datos a registrar son: nombre, apellido, cédula, correo electrónico y clave.
- **Menú**, el sistema debe permitir almacenar el listado de platos y bebidas disponibles ordenado por categorías.
- **Platos y/o bebidas**, detalles que debe almacenar el sistema, como denominación, descripción (opcional) y precio.
- **Pedido**, el sistema debe permitir la gestión de pedidos realizados por el cliente o por meseros, este está compuesto por información del cliente (nombres, cédula, celular o puede elegir con detalles por defecto), información del pedido (platos y/o bebidas pedidas, cantidad y precio unitario) y número de mesa.

Requerimientos funcionales

El sistema permitirá a todos los usuarios:

Código de requerimiento	Descripción	Categoría
RF01	Iniciar sesión con sus credenciales de cuenta: usuario y clave.	Visible
RF02	Actualizar su clave de usuario.	Visible
RF03	Cerrar la sesión de su cuenta.	Inferido
RF04	Actualizar información personal de su propia cuenta de usuario	Visible

El sistema permitirá al administrador:

Código de requerimiento	Descripción	Categoría
RF05	Registrar cuentas de usuario. Los datos a registrar son: nombre, apellido, cédula, correo electrónico y clave.	Visible
RF06	Registrar roles a las cuentas de usuario, estos serán: Administrador, Cocinero, Mesero y Cliente.	Visible
RF07	Actualizar los roles de usuario.	Visible
RF08	Buscar cuentas de usuario por el correo electrónico o cédula.	Inferido
RF09	Dar de baja cuentas de usuario	Visible

El sistema permitirá al cliente:

Código de requerimiento	Descripción	Categoría
RF10	Registrar su cuenta de usuario. Los datos obligatorios a registrar son: nombre, apellido, correo electrónico y clave, mientras que opcionales: cédula, dirección y número de celular.	Visible
RF11	Visualizar el menú disponible.	Visible
RF12	Elegir un plato del menú y ver los detalles del producto, estos son: denominación, precio y descripción (opcional).	Visible
RF13	Seleccionar platos y/o bebidas del menú	Visible
RF14	Quitar productos de su pedido, antes de que este sea confirmado.	Visible
RF15	Y al mesero generar un pedido con los platos seleccionados y el número de mesa.	Visible
RF16	Ver el detalle de su pedido, así como el valor a pagar por el consumo.	Visible
RF17	Dar por terminado su consumo, cuando esto haya pasado, el sistema debe notificar al cajero para que haga el cobro.	Inferido

El sistema permitirá al Cocinero:

Código de requerimiento	Descripción	Categoría
RF18	Visualizar los pedidos pendientes de preparar, realizados por los clientes	Visible
RF19	Actualizar el estado de los pedidos a despachado cuando este ha sido terminado	Inferido

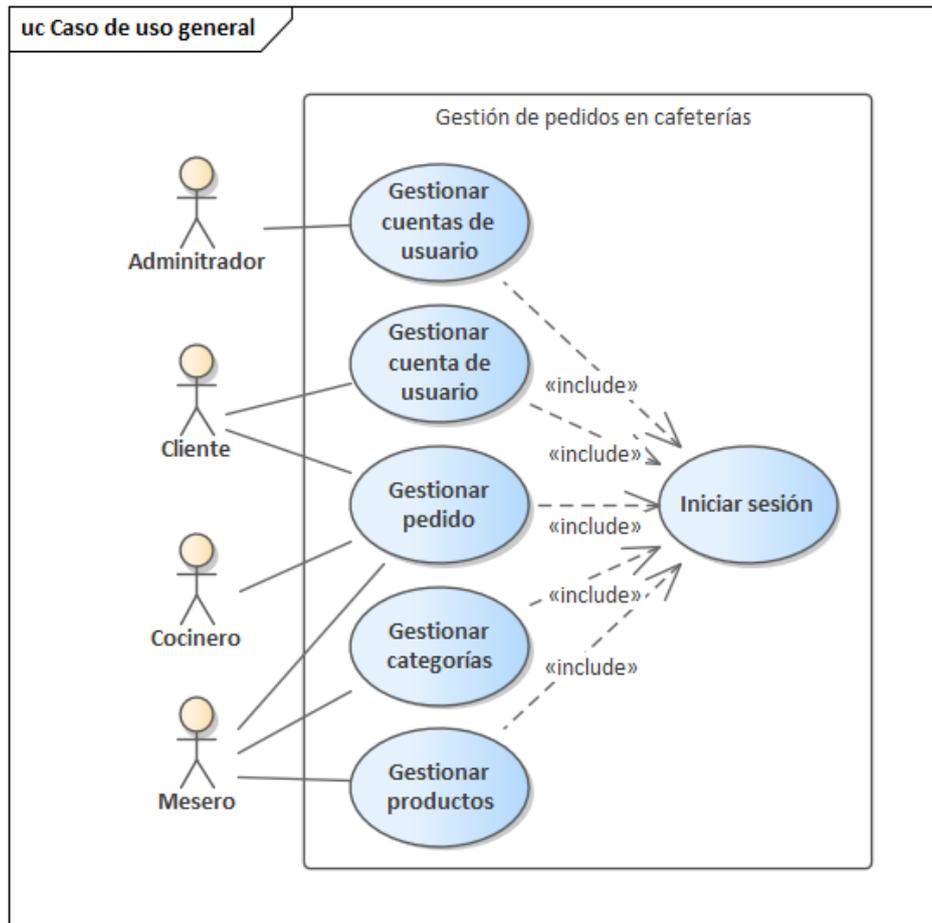
El sistema permitirá al mesero:

Código de requerimiento	Descripción	Categoría
RF20	Registrar categorías al menú, los datos a registrar son: denominación	Visible
RF21	Actualizar la denominación de las categorías del menú	Inferido
RF22	Buscar categorías por denominación.	Inferido
RF23	Dar de baja categorías	Visible
RF24	Registrar platos o bebidas en el menú, los datos a guardar son: denominación, descripción (opcional) y precio.	Visible
RF25	Actualizar la información de los platos y/o bebidas.	Visible
RF26	Buscar platos y/o bebidas por denominación.	Inferido
RF27	Dar de baja platos y/o bebidas	Visible
RF28	Visualizar el detalle los pedidos que han sido despachados a través del número de mesa o nombre	Visible
RF29	Calcular el valor a pagar, con impuestos, de pedidos en estado despachado	oculto
RF30	Generar un comprobante de pago en base al detalle de pedidos despachados	Visible

Requerimientos no funcionales

Código de requerimiento	Descripción	Categoría
RNF01	Cifrar las claves de cuentas de usuario	Seguridad
RNF02	Los permisos de acceso al sistema podrán ser actualizados únicamente por el usuario administrador	Seguridad
RNF03	Amigable y fácil de usar	Interfaz Gráfica de usuario
RNF04	El sistema será desarrollado para ser usado desde navegadores web de ordenadores o dispositivos móviles	Plataforma de desarrollo
RNF05	El sistema debe estar disponible durante las 24 horas del día de todos los días de la semana	Disponibilidad
RNF06	El tiempo de respuesta máximo esperado es de 2 segundos	Eficiencia
RNF07	Los registros modificados en la base de datos deben ser actualizados para todos los usuarios que accedan al sistema en por lo menos 2 segundos	Eficiencia
RNF08	El diseño del sistema debe ser de tipo Responsive y se adapte a todo tamaño de pantalla	Usabilidad

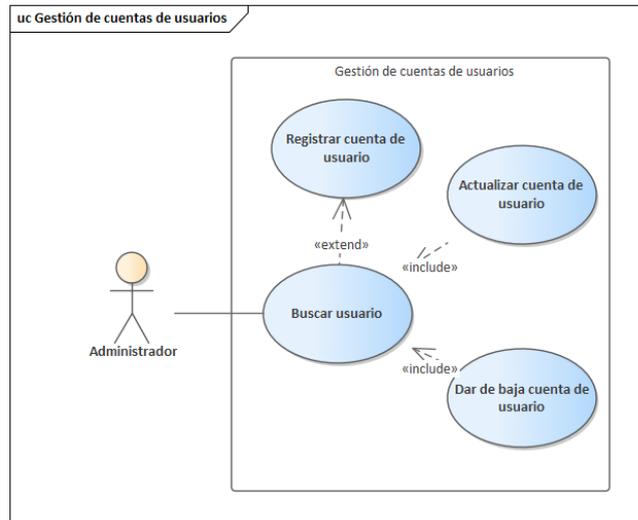
Modelado de casos de uso



Identificador	CU01
Nombre	Iniciar sesión
Actores	Todos los usuarios del sistema.
Descripción	Autenticación del usuario en el sistema.
Precondiciones	Tener cuenta registrada en el sistema. Estar en la página "Iniciar sesión".
Flujo Principal	El usuario ingresa sus credenciales de acceso (correo y clave). El usuario pulsa el botón "Iniciar sesión". El sistema verifica la información ingresada en el formulario. El sistema autentica al usuario y redirige a la página principal con acceso a las funciones autorizadas a su rol. El caso de uso finaliza.
Flujo Alternativo	Campos obligatorios vacíos 4.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario. Credenciales de acceso no válidas 5.1. El sistema muestra un mensaje indicando que las credenciales no son correctas.

Servicio de usuarios

Los diagramas de caso de uso detallados o explotados se presentan a continuación. El caso de uso Gestionar cuentas de usuarios, se ven las acciones que realizaría un administrador.



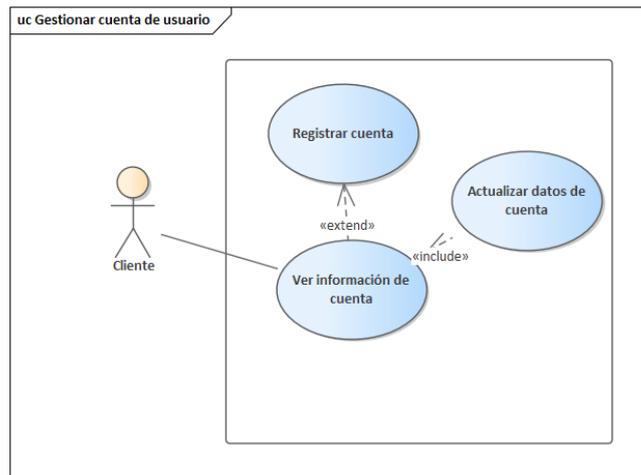
Identificador	CU02
Nombre	Registrar cuentas de usuario
Actor	Administrador
Descripción	Registro de nuevas cuentas de usuario
Precondición	Iniciar sesión Encontrarse en la página Listar usuarios
Flujo normal	<p>El administrador pulsa el botón “Registrar usuario”.</p> <p>El sistema muestra un formulario registrar usuario.</p> <p>El administrador ingresa la información del usuario (nombre, apellido, dirección de correo electrónico)</p> <p>El administrador asigna un rol</p> <p>El administrador pulsa el botón “Guardar”.</p> <p>El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.</p> <p>El sistema verifica que la dirección de correo electrónico no esté registrada.</p> <p>El sistema guarda los cambios y muestra un mensaje indicando que el usuario se ha registrado con éxito.</p>
Flujo alterno	<p>Campos obligatorios vacíos</p> <p>6.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.</p> <p>Validación de cédula incorrecta</p> <p>6.1. El sistema muestra un mensaje de alerta indicando que el número de cédula de la entidad es incorrecto cuando no es de tipo pasarte</p> <p>Formato de correo incorrecto</p> <p>6.1. El sistema muestra un mensaje indicando que la dirección de correo electrónico no cumple el formato.</p> <p>Registro existente</p> <p>7.1. El sistema muestra un mensaje indicando que hay un usuario registrado con la dirección de correo electrónico ingresada.</p>

Identificador	CU03
Nombre	Actualizar cuenta de usuario
Actor	Administrador
Descripción	Actualizar datos de cuenta de usuario
Precondición	Iniciar sesión Encontrarse en la página Listar usuarios Buscar usuario
Flujo normal	El administrador pulsa el botón “Actualizar”, junto a la información del usuario. El sistema muestra un formulario con la información guardada previamente del usuario y los campos permitidos con permisos de cambio. El administrador modifica la información. El administrador pulsa el botón “Guardar”. El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. El sistema guarda los cambios y muestra un mensaje indicando que el usuario se ha actualizado con éxito.
Flujo alternativo	Campos obligatorios vacíos 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.

Identificador	CU04
Nombre	Dar de baja cuenta de usuario
Actor	Administrador
Descripción	Inhabilitar cuenta de usuario
Precondición	Iniciar sesión Encontrarse en la página Listar usuarios Buscar usuario
Flujo normal	El administrador pulsa el botón “Dar de baja”, junto a la información del usuario. El sistema muestra un mensaje de confirmación de la acción. El administrador pulsa el botón “Guardar”. El sistema da de baja el usuario y muestra un mensaje indicando que la operación se ha realizado con éxito.
Flujo alternativo	Cancelar operación 3.1. El usuario presiona el botón “Cancelar” 3.2. El sistema cancela la operación.

Identificador	CU05
Nombre	Buscar usuario
Actor	Administrador
Descripción	Buscar cuenta de usuario por correo electrónico
Precondición	Iniciar sesión Encontrarse en la página Listar usuarios
Flujo normal	El administrador ingresa la dirección de correo electrónico en el campo ubicado en la parte superior de la lista de usuarios. El administrador presiona el botón “Buscar” El sistema busca el registro de usuario y muestra.
Flujo alternativo	Usuario no encontrado 3.1. El sistema muestra un mensaje indicando que el usuario con el correo ingresado no ha sido encontrado.

Los casos de uso descritos a continuación responden a la acción que realiza el cliente de gestión de su propia cuenta, mientras que lo descrito anteriormente, son las acciones del administrador con la gestión de cuentas de todos los roles.



Identificador	CU06
Nombre	Registrar cuenta de usuario
Actor	Cliente
Descripción	Registro de cuenta de usuario con rol cliente
Precondición	Encontrarse en la página Registrar cuenta
Flujo normal	<p>El cliente ingresa la información del usuario (nombre, apellido, dirección de correo electrónico).</p> <p>El cliente pulsa el botón “Guardar”.</p> <p>El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.</p> <p>El sistema verifica que la dirección de correo electrónico no esté registrada.</p> <p>El sistema guarda los cambios y muestra un mensaje indicando que el usuario se ha registrado con éxito.</p>
Flujo alterno	<p>Campos obligatorios vacíos</p> <p>4.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.</p> <p>Validación de cédula incorrecta</p> <p>4.1. El sistema muestra un mensaje de alerta indicando que el número de cédula de la entidad es incorrecto cuando no es de tipo pasarte</p> <p>Formato de correo incorrecto</p> <p>4.1. El sistema muestra un mensaje indicando que la dirección de correo electrónico no cumple el formato.</p> <p>Registro existente</p> <p>5.1. El sistema muestra un mensaje indicando que hay un usuario registrado con la dirección de correo electrónico ingresada.</p>

Identificador	CU07
Nombre	Actualizar cuenta de usuario
Actor	Cliente
Descripción	Actualizar datos de cuenta de usuario rol cliente
Precondición	<p>Iniciar sesión</p> <p>Encontrarse en la página Ver información de cuenta</p>
Flujo normal	<p>El cliente pulsa el botón “Actualizar”.</p> <p>El sistema muestra un formulario con la información guardada previamente del usuario y los campos permitidos con permisos de cambio.</p> <p>El cliente modifica la información.</p> <p>El cliente pulsa el botón “Guardar”.</p> <p>El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.</p>

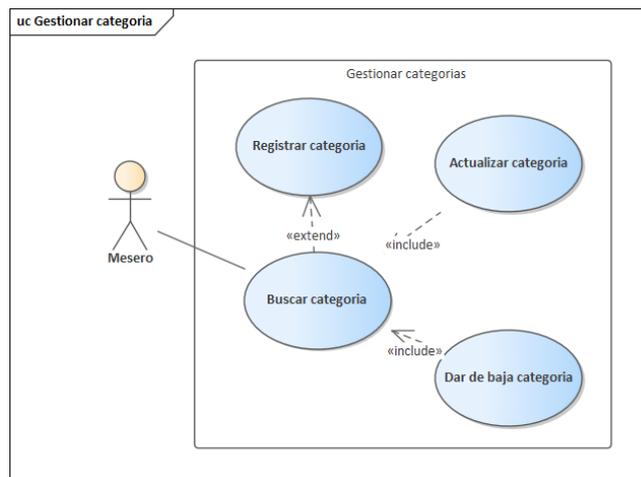
	El sistema guarda los cambios y muestra un mensaje indicando que el usuario se ha actualizado con éxito.
Flujo alterno	Campos obligatorios vacíos 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.

Tabla 21. Descripción de caso de uso - Ver información de cuenta de usuario

Identificador	CU08
Nombre	Ver información de cuenta
Actor	Cliente
Descripción	Ver la información de la cuenta de usuario
Precondición	Iniciar sesión
Flujo normal	El cliente pulsa el botón "Ver perfil" o avatar de usuario. El sistema busca el usuario y muestra la información registrada por el usuario.
Flujo alterno	

Servicio de Productos

Los diagramas de casos de uso explotados correspondientes al servicio de Productos son los siguientes:



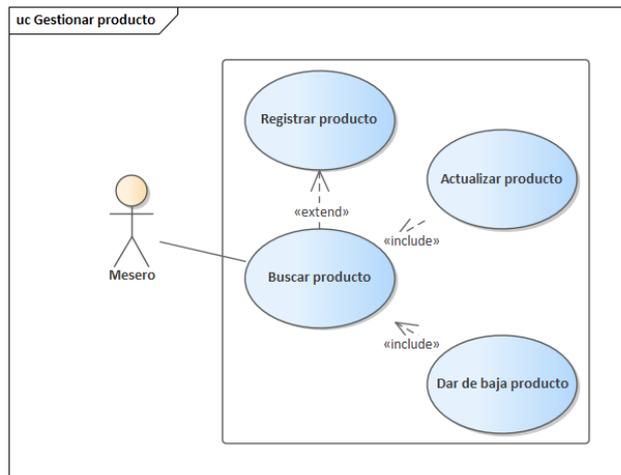
Identificador	CU09
Nombre	Registrar categoría
Actor	Mesero
Descripción	Registro de nuevas categorías
Precondición	Iniciar sesión Encontrarse en la página Listar categorías
Flujo normal	El mesero pulsa el botón "Registrar categoría". El sistema muestra un formulario registrar categoría. El mesero ingresa la información de la categoría (nombre y descripción) El mesero pulsa el botón "Guardar". El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. El sistema guarda la categoría muestra un mensaje indicando que el registro ha sido exitoso.

Flujo alternativo	Campos obligatorios vacíos 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.
--------------------------	--

Identificador	CU10
Nombre	Actualizar categoría
Actor	Mesero
Descripción	Actualizar datos de categoría
Precondición	Iniciar sesión Encontrarse en la página Listar categorías Buscar categoría
Flujo normal	El mesero pulsa el botón “Actualizar”, junto a la información de la categoría. El sistema muestra un formulario con la información guardada previamente de la categoría y los campos permitidos con permisos de cambio. El mesero modifica la información. El mesero pulsa el botón “Guardar”. El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. El sistema actualiza la categoría y muestra un mensaje indicando que la operación se ha realizado con éxito.
Flujo alternativo	Campos obligatorios vacíos 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.

Identificador	CU11
Nombre	Dar de baja categoría
Actor	Mesero
Descripción	Inhabilitar categoría
Precondición	Iniciar sesión Encontrarse en la página Listar categorías Buscar categoría
Flujo normal	El mesero pulsa el botón “Dar de baja”, junto a la información de la categoría. El sistema muestra un mensaje de confirmación de la acción. El mesero pulsa el botón “Guardar”. El sistema da de baja la categoría y muestra un mensaje indicando que la operación se ha realizado con éxito.
Flujo alternativo	Cancelar operación 3.1. El usuario presiona el botón “Cancelar” 3.2. El sistema cancela la operación.

Identificador	CU12
Nombre	Buscar categoría
Actor	Mesero
Descripción	Buscar categoría por denominación
Precondición	Iniciar sesión Encontrarse en la página Listar categorías
Flujo normal	El mesero ingresa el nombre de la categoría en el campo ubicado en la parte superior de la lista de categorías. El mesero presiona el botón “Buscar” El sistema busca la categoría y muestra.
Flujo alternativo	Categoría no encontrada 3.1. El sistema muestra un mensaje indicando que la categoría con el nombre ingresado no ha sido encontrada.



Identificador	CU13
Nombre	Registrar producto
Actor	Mesero
Descripción	Registro de nuevos productos
Precondición	Iniciar sesión Encontrarse en la página Listar productos
Flujo normal	El mesero pulsa el botón “Registrar producto”. El sistema muestra un formulario de registro de producto. El mesero ingresa la información (nombre, descripción y precio). El mesero pulsa el botón “Guardar”. El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. El sistema guarda el producto y muestra un mensaje indicando que el registro ha sido exitoso.
Flujo alterno	Campos obligatorios vacíos 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.

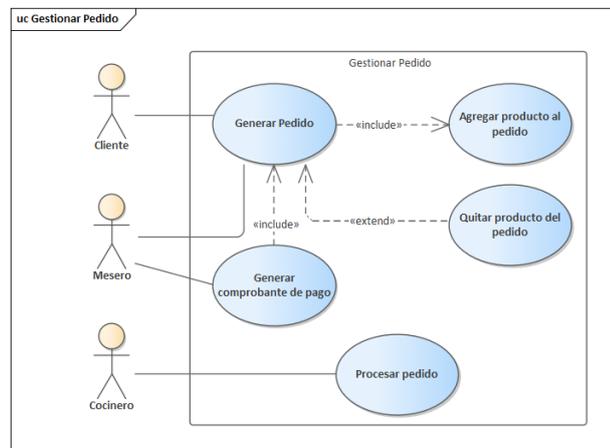
Identificador	CU14
Nombre	Actualizar producto
Actor	Mesero
Descripción	Actualizar datos de producto
Precondición	Iniciar sesión Encontrarse en la página Listar productos Buscar producto
Flujo normal	El mesero pulsa el botón “Actualizar”, junto a la información del producto. El sistema muestra un formulario con la información guardada previamente del producto y los campos permitidos con permisos de cambio. El mesero modifica la información. El mesero pulsa el botón “Guardar”. El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. El sistema actualiza el producto muestra un mensaje indicando que la operación se ha realizado con éxito.
Flujo alterno	Campos obligatorios vacíos 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.

Identificador	CU15
Nombre	Dar de baja producto
Actor	Mesero
Descripción	Inhabilitar producto
Precondición	Iniciar sesión Encontrarse en la página Listar productos Buscar producto
Flujo normal	El mesero pulsa el botón “Dar de baja”, junto a la información del producto. El sistema muestra un mensaje de confirmación de la acción. El mesero pulsa el botón “Guardar”. El sistema da de baja el producto y muestra un mensaje indicando que la operación se ha realizado con éxito.
Flujo alternativo	Cancelar operación 3.1. El usuario presiona el botón “Cancelar” 3.2. El sistema cancela la operación.

Identificador	CU16
Nombre	Buscar producto
Actor	Mesero
Descripción	Buscar producto por denominación
Precondición	Iniciar sesión Encontrarse en la página Listar productos
Flujo normal	El mesero ingresa el nombre del producto en el campo ubicado en la parte superior de la lista de productos. El mesero presiona el botón “Buscar” El sistema busca el producto y muestra el registro.
Flujo alternativo	Producto no encontrado 3.1. El sistema muestra un mensaje indicando que el producto con el nombre ingresado no ha sido encontrado.

Servicio de pedidos

Este servicio cuenta de una acción general que involucra a varios actores. Las acciones y la interacción entre usuarios se presentan en los próximos recursos.



Identificador	CU17
Nombre	Agregar productos al pedido
Actor	Mesero o cliente
Descripción	Agregar productos a la lista de un pedido
Precondición	Haberse cumplido el caso de uso Iniciar sesión Encontrarse en la página pública del menú
Flujo normal	El usuario pulsa el botón "Agregar a pedido". El sistema muestra la opción de elegir la cantidad de productos. El usuario presiona el botón "Agregar" El sistema agrega a la lista del pedido y muestra un mensaje indicando que la operación se realizó con éxito.
Flujo alterno	

Identificador	CU18
Nombre	Quitar producto al pedido
Actor	Mesero o cliente
Descripción	Quitar productos a la lista de un pedido
Precondición	Haberse cumplido el caso de uso Iniciar sesión Encontrarse en la lista de productos del pedido Haber agregar productos al pedido
Flujo normal	El usuario pulsa el botón "Quitar", ubicado junto al producto que desea retirar. El sistema remueve el producto de la lista El sistema muestra la lista actualizada.
Flujo alterno	

Identificador	CU19
Nombre	Generar pedido
Actor	Mesero o cliente
Descripción	Confirmar pedido
Precondición	Haberse cumplido el caso de uso Iniciar sesión Tener productos agregados
Flujo normal	El usuario pulsa el botón "Generar pedido". El sistema cambia el estado del pedido a "Pendiente" El sistema envía la orden a cocina. El sistema muestra el detalle del pedido al cliente.
Flujo alterno	Cancelar pedido El usuario pulsa el botón "Cancelar pedido" El sistema borra la lista del pedido y muestra un mensaje de pedido cancelado.

Identificador	CU20
Nombre	Procesar pedido
Actor	Cocinero
Descripción	Procesar el estado del pedido
Precondición	Haberse cumplido el caso de uso Iniciar sesión Encontrarse en la página Listar pedidos
Flujo normal	El cocinero pulsa el botón "Procesar pedido", ubicado junto a la información del pedido. El sistema muestra un mensaje de confirmación de la acción al cocinero. El sistema muestra para el usuario que realizó el pedido, sea cliente o mesero, el estado del pedido. El cocinero, una vez terminado de preparar el pedido, confirma el pedido como despachado.

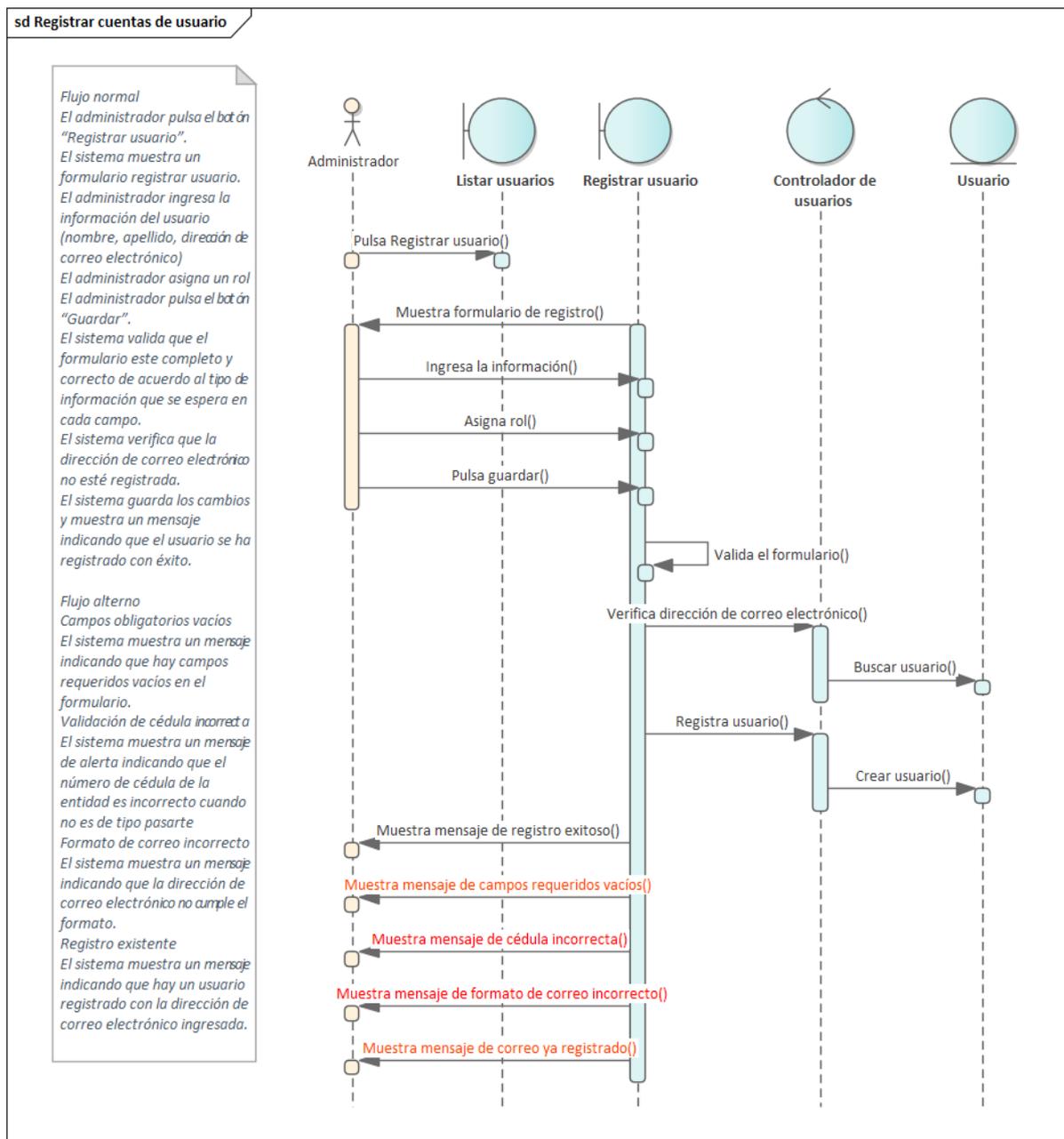
	El sistema notifica a los meseros sobre el pedido listo para entregar a mesa.
Flujo alterno	

Identificador	CU21
Nombre	Generar comprobante de pago
Actor	Mesero
Descripción	El mesero cobra por los productos consumidos
Precondición	Haberse cumplido el caso de uso Iniciar sesión Tener un pedido con estado procesado.
Flujo normal	<p>El mesero pulsa el botón "Generar comprobante de pago".</p> <p>El sistema muestra un formulario para registrar datos del cliente, del pago y el detalle de consumo.</p> <p>El mesero completa el formulario.</p> <p>El mesero, una vez realizado y verificado el pago, pulsa el botón "Guardar".</p> <p>El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.</p> <p>El sistema guarda el comprobante de pago y el pago.</p> <p>El sistema envía el comprobante de pago vía correo electrónico.</p> <p>El sistema actualiza el estado del pedido a "Terminado" y muestra un mensaje indicando que la operación se ha realizado con éxito.</p>
Flujo alterno	<p>Consumidor final</p> <p>2.1. El cliente puede decidir que en su comprobante no salgan sus datos, sino información por defecto.</p> <p>7.1. El sistema no envía comprobante de pago.</p> <p>Datos diferentes a los del registro de usuario (de existir)</p> <p>3.1. El cliente solicita al mesero que se ingresen otros datos para el comprobante de pago.</p> <p>3.2. El mesero registra los datos indicados por el cliente.</p> <p>Campos obligatorios vacíos</p> <p>6.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.</p>

Modelado de interacción mediante diagrama de secuencia

Una vez descrito el comportamiento de los actores con relación a los casos de uso, se presentan diagramas de secuencia que detallan la interacción entre las diferentes capas del sistema. Cada diagrama, está compuesto por el flujo normal que debería seguir, y en texto de color rojo, se presenta el flujo alterno. La documentación de los diagramas, está dividida por los servicios previamente mencionados.

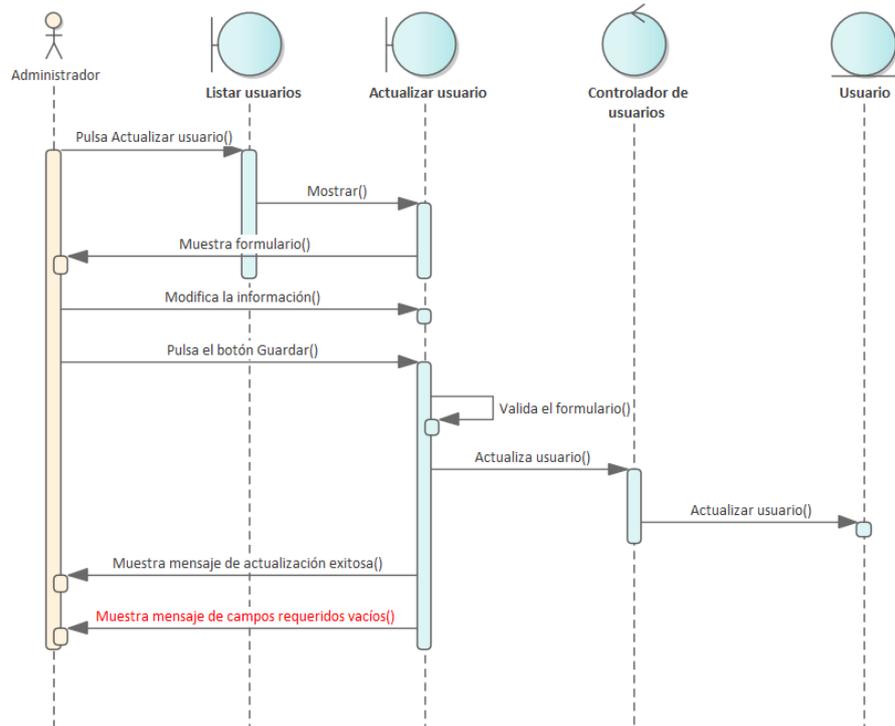
Servicio de usuarios



sd Actualizar cuentas de usuario

Flujo normal
 El administrador pulsa el botón "Actualizar", junto a la información del usuario. El sistema muestra un formulario con la información guardada previamente del usuario y los campos permitidos con permisos de cambio. El administrador modifica la información. El administrador pulsa el botón "Guardar". El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. El sistema guarda los cambios y muestra un mensaje indicando que el usuario se ha actualizado con éxito.

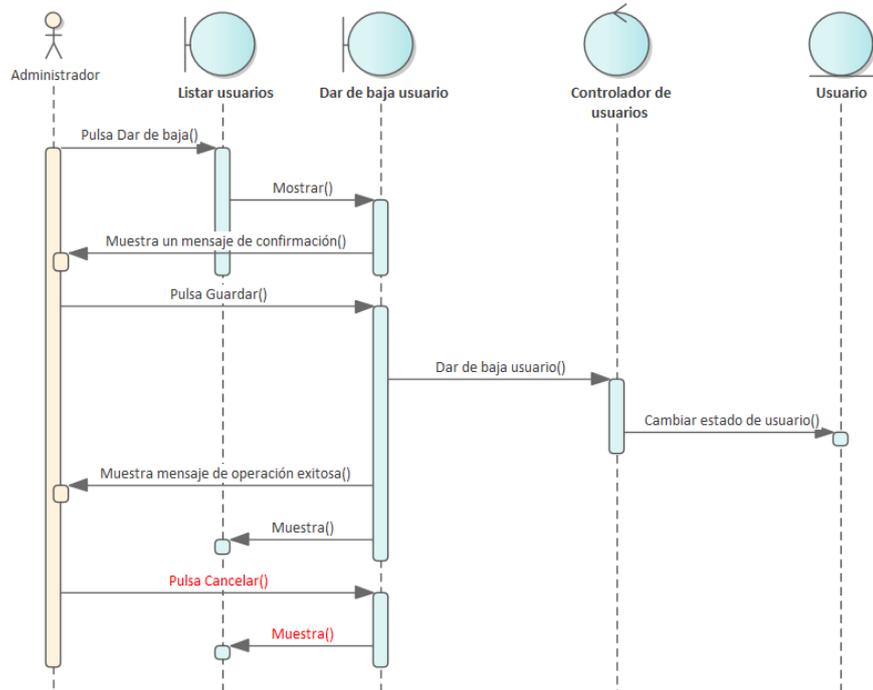
Flujo alterno
 Campos obligatorios vacíos
 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.



sd Dar de baja cuenta de usuario

Flujo normal
 El administrador pulsa el botón "Dar de baja", junto a la información del usuario. El sistema muestra un mensaje de confirmación de la acción. El administrador pulsa el botón "Guardar". El sistema da de baja y muestra un mensaje indicando que el usuario se ha sido dado de baja con éxito.

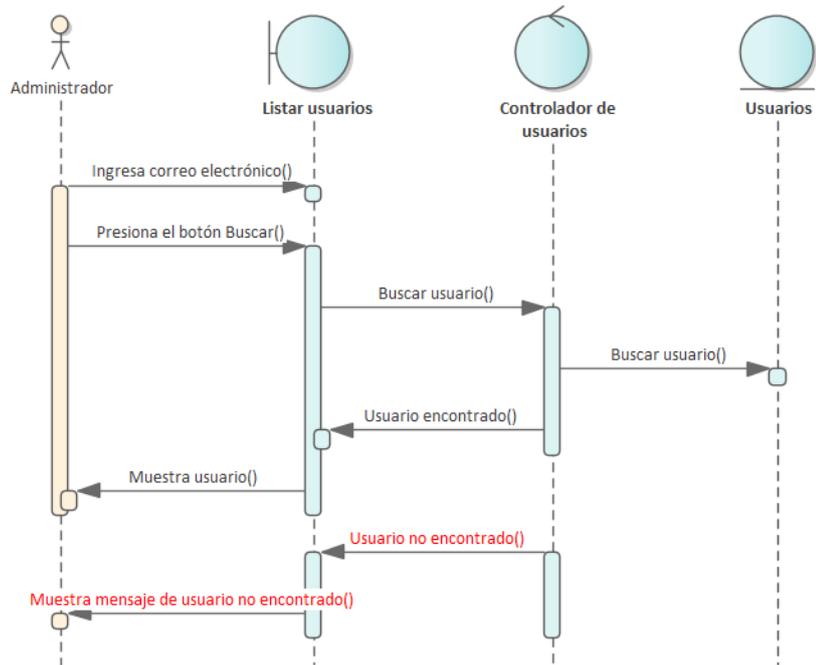
Flujo alterno
 Cancelar operación
 3.1. El usuario presiona el botón "Cancelar"
 3.2. El sistema cancela la operación.



sd Buscar usuario

Flujo normal
 El administrador ingresa el correo en el campo ubicado en la parte superior de la lista de usuarios.
 El administrador presiona el botón "Buscar"
 El sistema busca el registro de usuario y muestra.

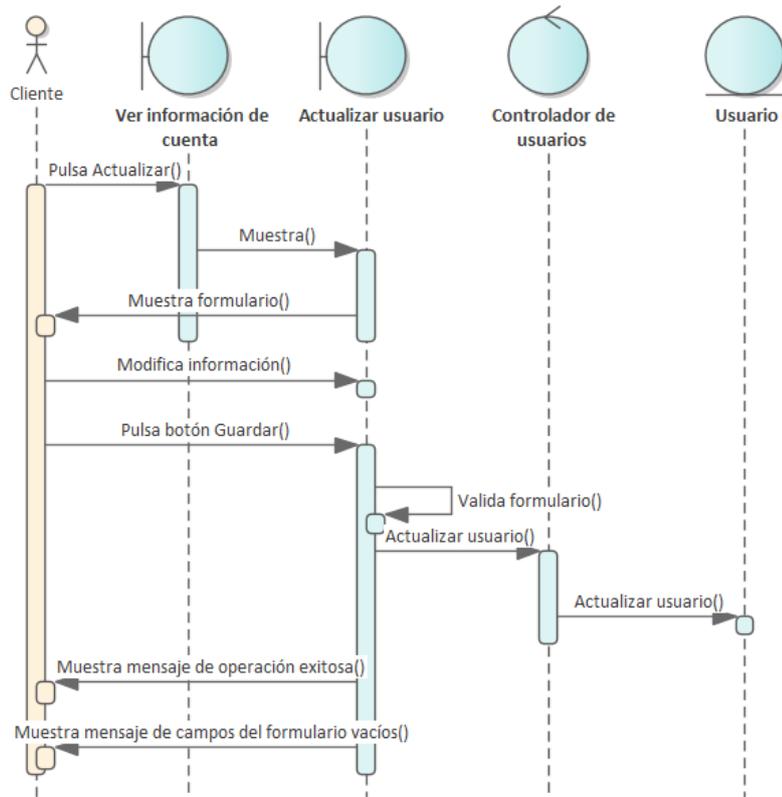
Flujo alterno
 Usuario no encontrado
 3.1. El sistema muestra un mensaje indicando que el usuario con el correo ingresado no ha sido encontrado.



sd Actualizar cuenta cliente

Flujo normal
 El cliente pulsa el botón "Actualizar".
 El sistema muestra un formulario con la información guardada previamente del usuario y los campos permitidos con permisos de cambio.
 El cliente modifica la información.
 El cliente pulsa el botón "Guardar".
 El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.
 El sistema guarda los cambios y muestra un mensaje indicando que el usuario se ha actualizado con éxito.

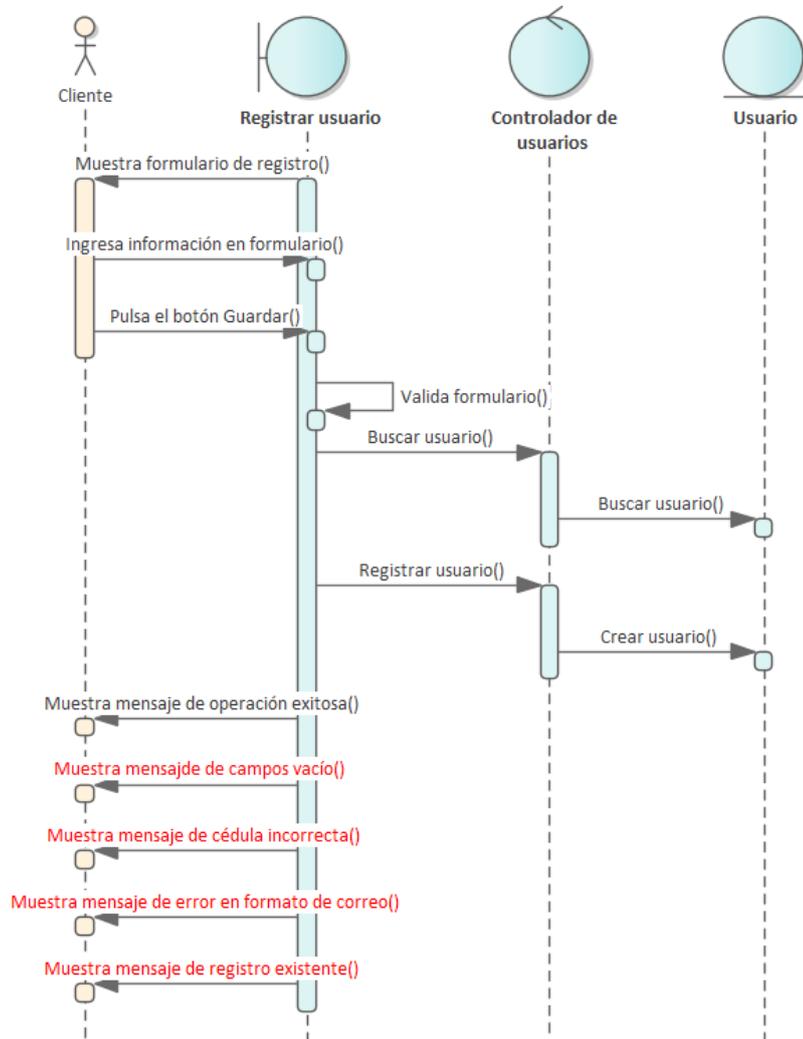
Flujo alterno
 Campos obligatorios vacíos
 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.



sd Registrar cuenta cliente

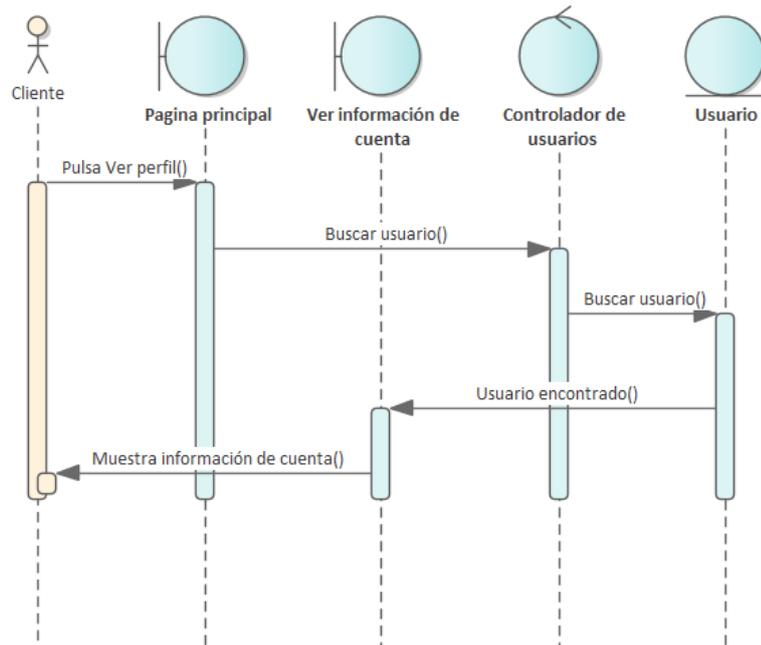
Flujo normal
 El sistema muestra un formulario registrar usuario.
 El cliente ingresa la información del usuario (nombre, apellido, dirección de correo electrónico).
 El cliente pulsa el botón "Guardar".
 El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.
 El sistema verifica que la dirección de correo electrónico no esté registrada.
 El sistema guarda los cambios y muestra un mensaje indicando que el usuario se ha registrado con éxito.

Flujo alterno
 Campos obligatorios vacíos
 4.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.
 Validación de cédula incorrecta
 4.1. El sistema muestra un mensaje de alerta indicando que el número de cédula de la entidad es incorrecto cuando no es de tipo pasarte
 Formato de correo incorrecto
 4.1. El sistema muestra un mensaje indicando que la dirección de correo electrónico no cumple el formato.
 Registro existente
 5.1. El sistema muestra un mensaje indicando que hay un usuario registrado con la dirección de correo electrónico ingresada.



sd Ver información de cuenta

Flujo normal
 El cliente pulsa el botón "Ver perfil" o avatar de usuario.
 El sistema busca el usuario y muestra la información registrada por el usuario.

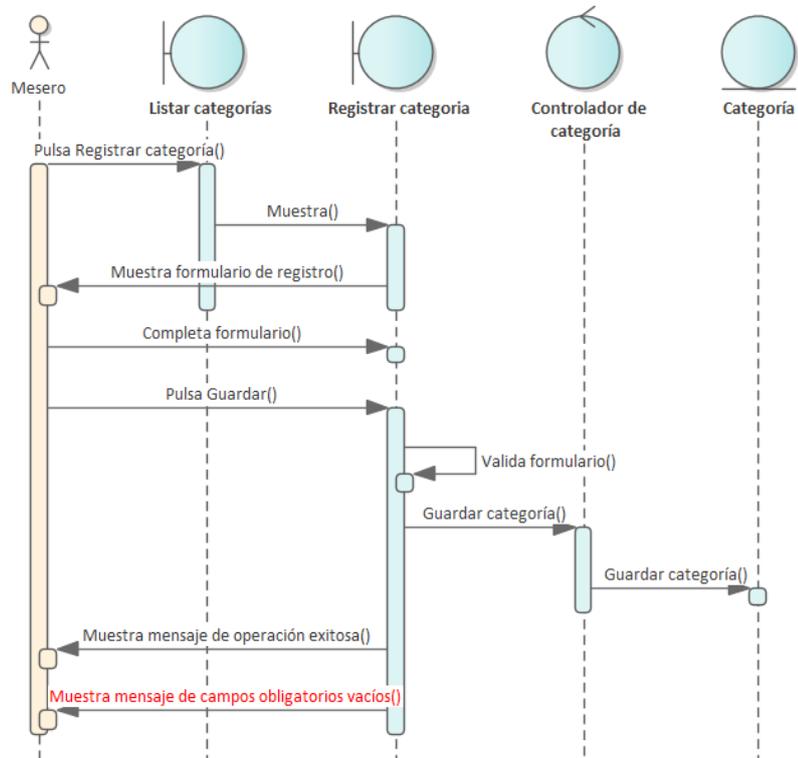


Servicio de productos

sd Registrar categoría

Flujo normal
 El mesero pulsa el botón "Registrar categoría".
 El sistema muestra un formulario registrar categoría.
 El mesero ingresa la información de la categoría (nombre y descripción)
 El mesero pulsa el botón "Guardar".
 El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.
 El sistema guarda la categoría muestra un mensaje indicando que el registro ha sido exitoso.

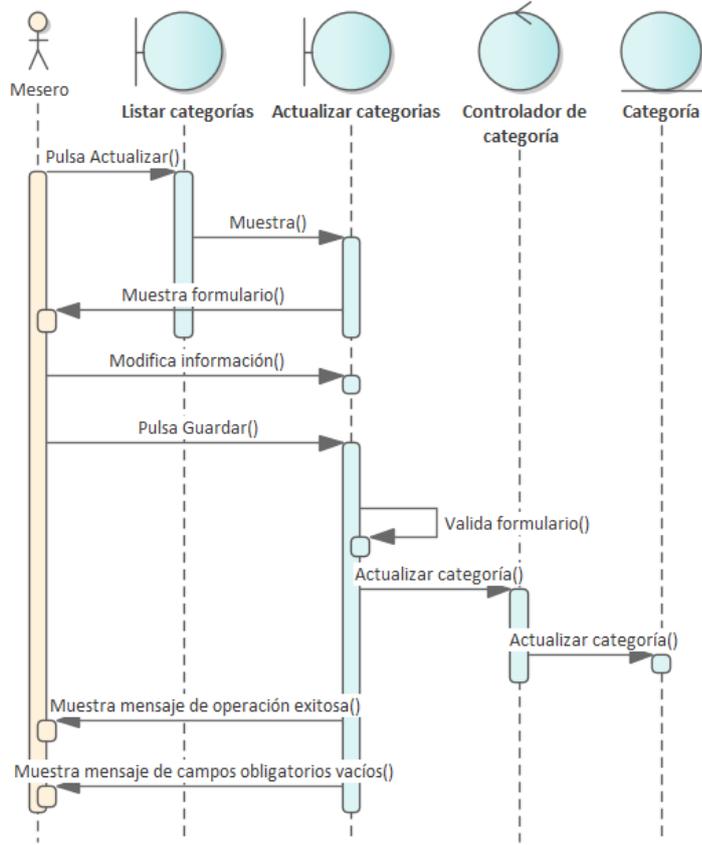
Flujo alterno
 Campos obligatorios vacíos 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.



sd Actualizar categoría

Flujo normal
 El mesero pulsa el botón "Actualizar", junto a la información de la categoría.
 El sistema muestra un formulario con la información guardada previamente de la categoría y los campos permitidos con permisos de cambio.
 El mesero modifica la información.
 El mesero pulsa el botón "Guardar".
 El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo.
 El sistema actualiza la categoría y muestra un mensaje indicando que la operación se ha realizado con éxito.

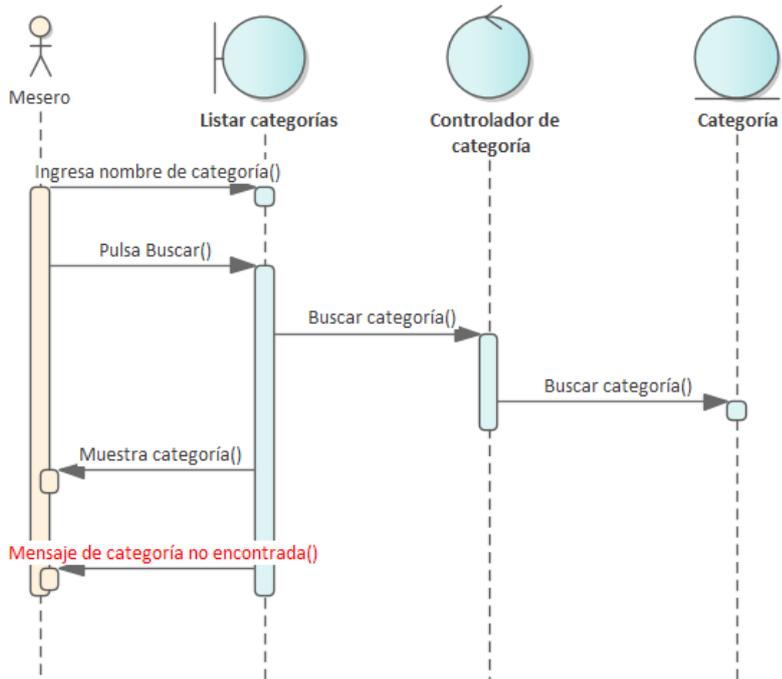
Flujo alterno
 Campos obligatorios vacíos
 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.



sd Buscar categoría

Flujo normal
 El mesero ingresa el nombre de la categoría en el campo ubicado en la parte superior de la lista de categorías.
 El mesero presiona el botón "Buscar"
 El sistema busca la categoría y muestra.

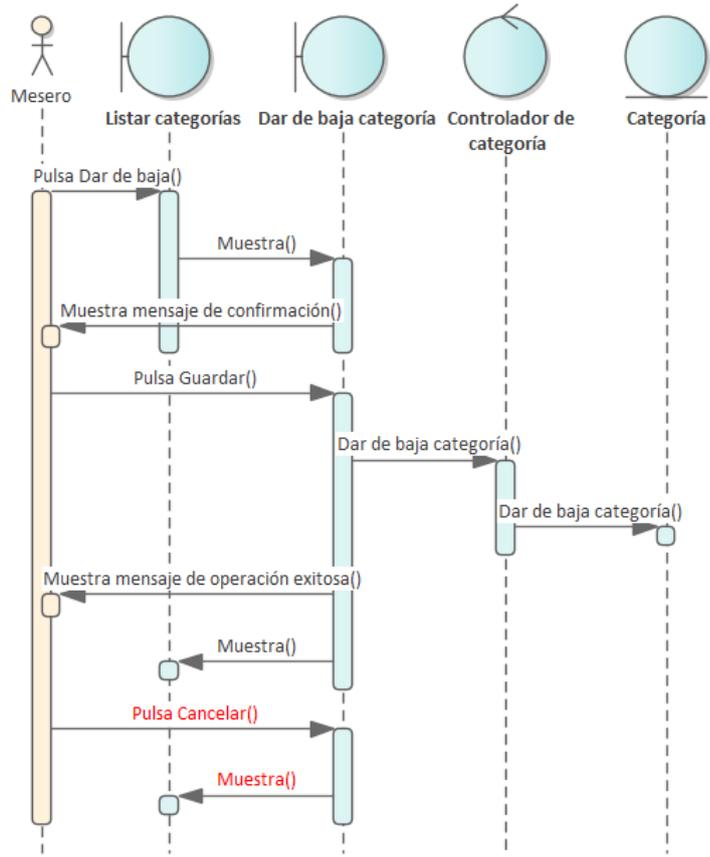
Flujo alterno
 Categoría no encontrada
 3.1. El sistema muestra un mensaje indicando que la categoría con el nombre ingresado no ha sido encontrada.



sd Dar de baja categoría

Flujo normal
 El mesero pulsa el botón "Dar de baja", junto a la información de la categoría.
 El sistema muestra un mensaje de confirmación de la acción.
 El mesero pulsa el botón "Guardar".
 El sistema da de baja la categoría y muestra un mensaje indicando que la operación se ha realizado con éxito.

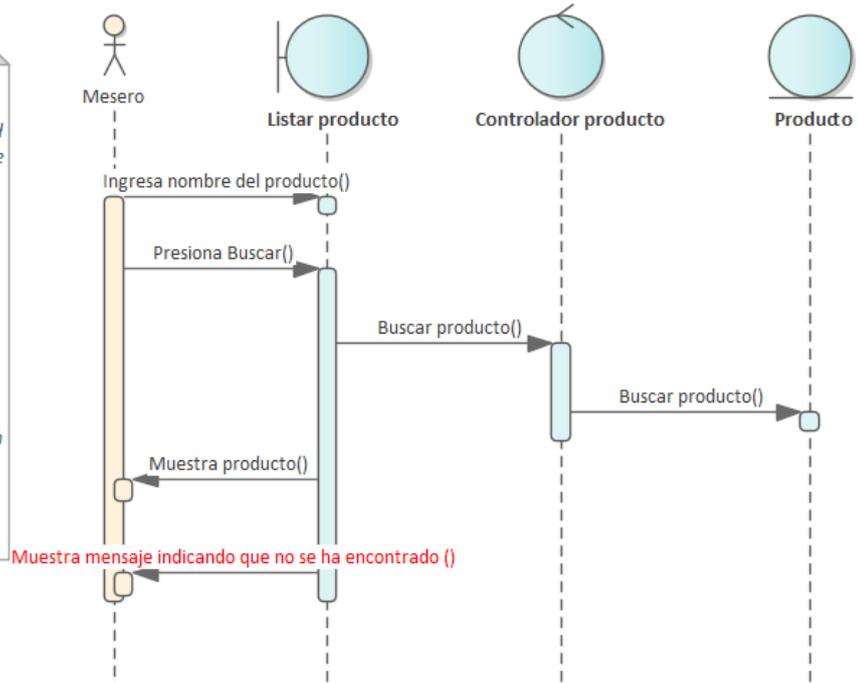
Flujo alterno
 Cancelar operación
 3.1. El usuario presiona el botón "Cancelar"
 3.2. El sistema cancela la operación.



sd Buscar producto

Flujo normal
 El mesero ingresa el nombre del producto en el campo ubicado en la parte superior de la lista de productos.
 El mesero presiona el botón "Buscar"
 El sistema busca el producto y muestra el registro.

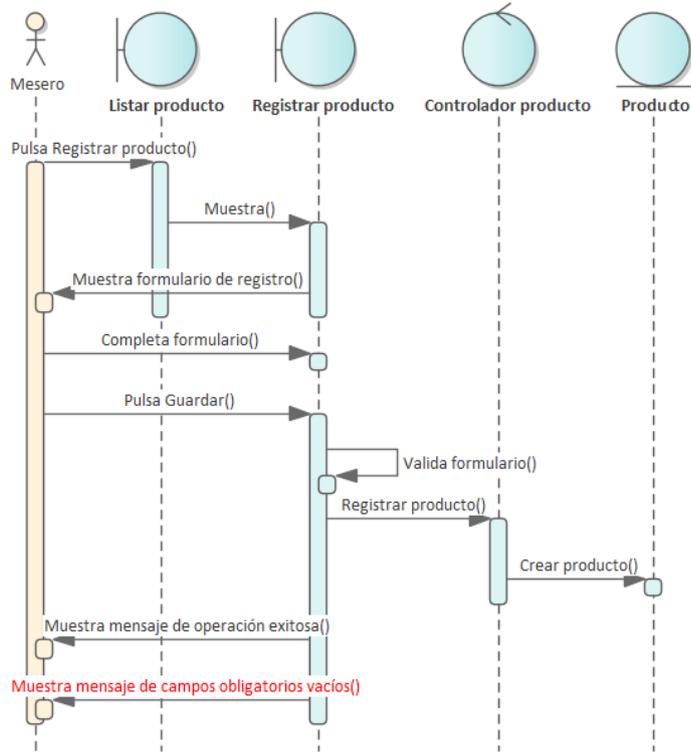
Flujo alterno
 Producto no encontrado
 3.1. El sistema muestra un mensaje indicando que el producto con el nombre ingresado no ha sido encontrado.



sd Registrar producto

Flujo normal
 El mesero pulsa el botón "Registrar producto".
 El sistema muestra un formulario de registro de producto.
 El mesero ingresa la información (nombre, descripción y precio).
 El mesero pulsa el botón "Guardar".
 El sistema valida que el formulario este completo y correcto de acuerdo a tipo de información que se espera en cada campo.
 El sistema guarda el producto y muestra un mensaje indicando que el registro ha sido exitoso.

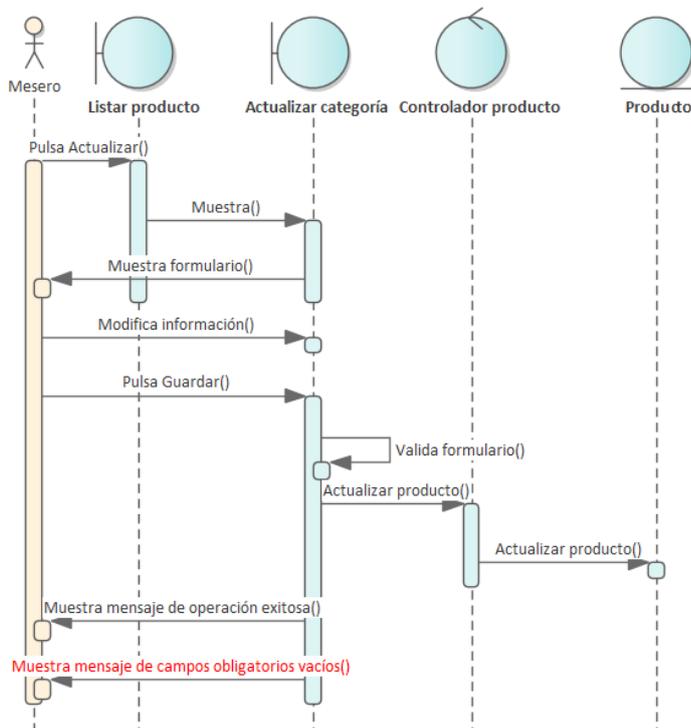
Flujo alterno
 Campos obligatorios vacíos 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.



sd Actualizar producto

Flujo normal
 El mesero pulsa el botón "Actualizar", junto a la información del producto.
 El sistema muestra un formulario con la información guardada previamente del producto y los campos permitidos con permisos de cambio.
 El mesero modifica la información.
 El mesero pulsa el botón "Guardar".
 El sistema valida que el formulario este completo y correcto de acuerdo a tipo de información que se espera en cada campo.
 El sistema actualiza el producto muestra un mensaje indicando que la operación se ha realizado con éxito.

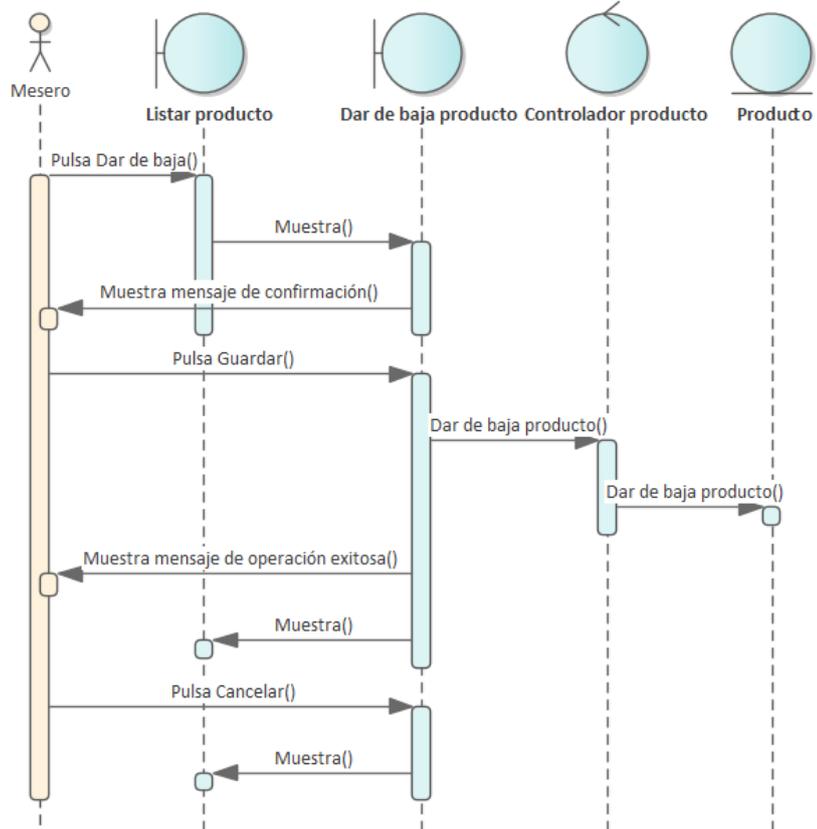
Flujo alterno
 Campos obligatorios vacíos 5.1. El sistema muestra un mensaje indicando que hay campos requeridos vacíos en el formulario.



sd Dar de baja producto

Flujo normal
 El mesero pulsa el botón "Dar de baja", junto a la información del producto.
 El sistema muestra un mensaje de confirmación de la acción.
 El mesero pulsa el botón "Guardar".
 El sistema da de baja el producto y muestra un mensaje indicando que la operación se ha realizado con éxito.

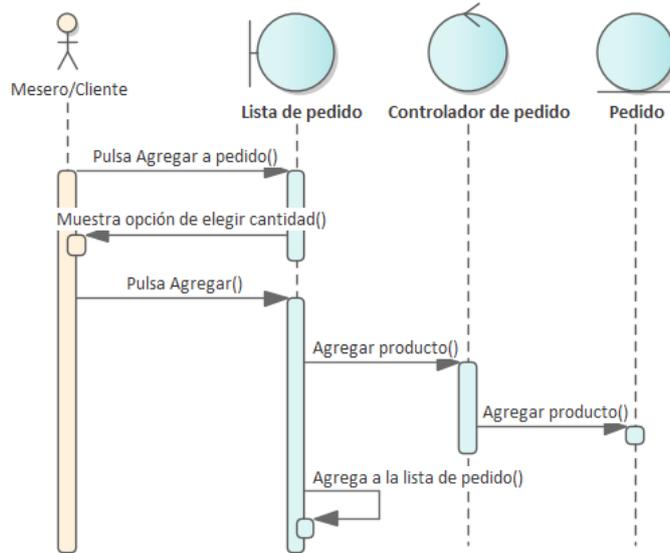
Flujo alterno
 Cancelar operación
 3.1. El usuario presiona el botón "Cancelar"
 3.2. El sistema cancela la operación.



Servicio de pedidos

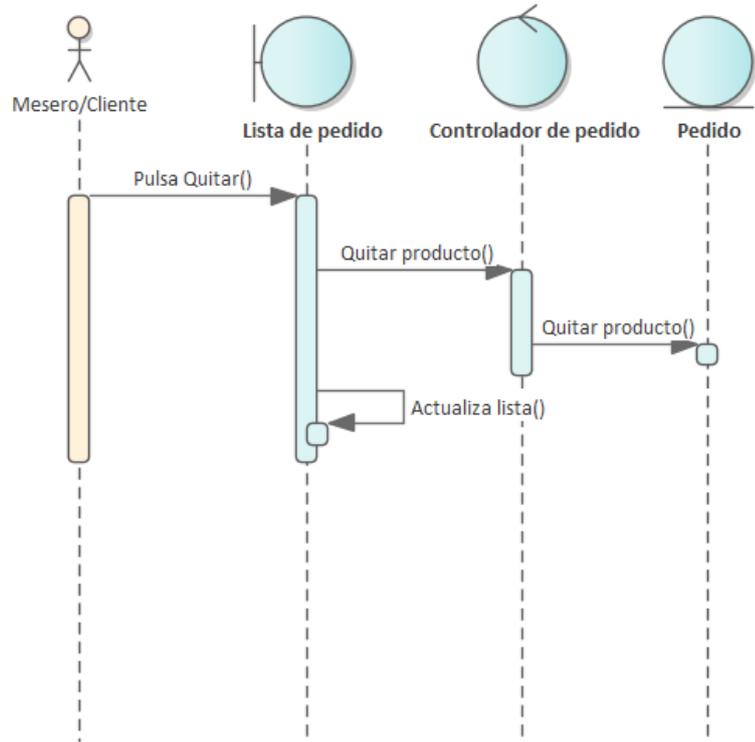
sd Agregar productos a pedido

Flujo normal
 El usuario pulsa el botón "Agregar a pedido".
 El sistema muestra la opción de elegir la cantidad de productos.
 El usuario presiona el botón "Agregar".
 El sistema agrega a la lista del pedido y muestra un mensaje indicando que la operación se realizó con éxito.



sd Quitar producto al pedido

Flujo normal
 El usuario pulsa el botón "Quitar", ubicado junto al producto que desea retirar.
 El sistema remueve el producto de la lista
 El sistema muestra la lista actualizada.



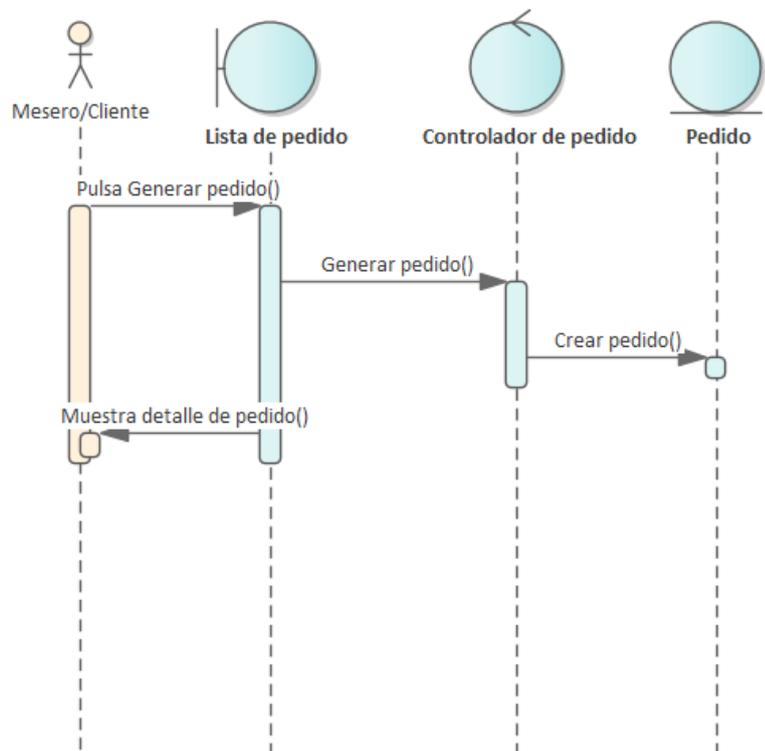
sd Generar pedido

Flujo normal
 El usuario pulsa el botón "Generar pedido".
 El sistema cambia el estado del pedido a "Pendiente"
 El sistema envía la orden a cocina.
 El sistema muestra el detalle del pedido al cliente.

Flujo alternativo
 Cancelar pedido

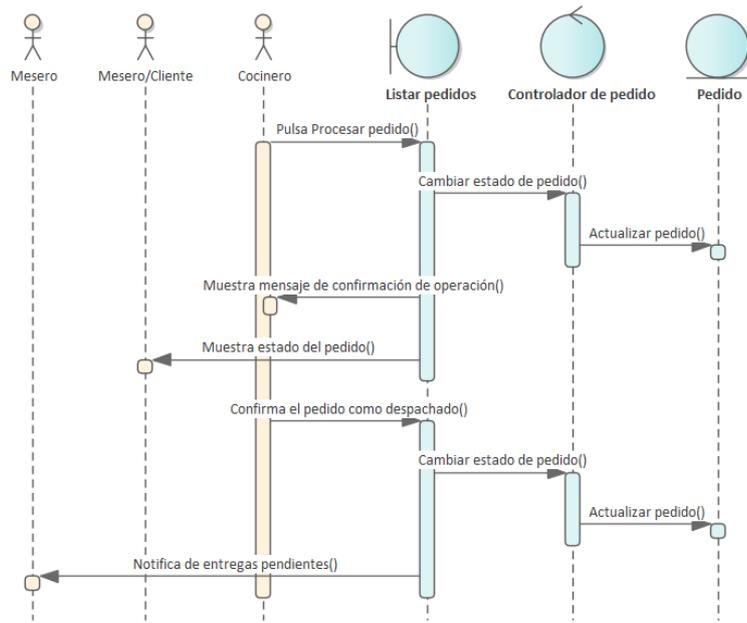
1.
 1. El usuario pulsa el botón "Cancelar pedido"

1.2. El sistema borra la lista del pedido y muestra un mensaje de pedido cancelado.



sd Procesar pedido

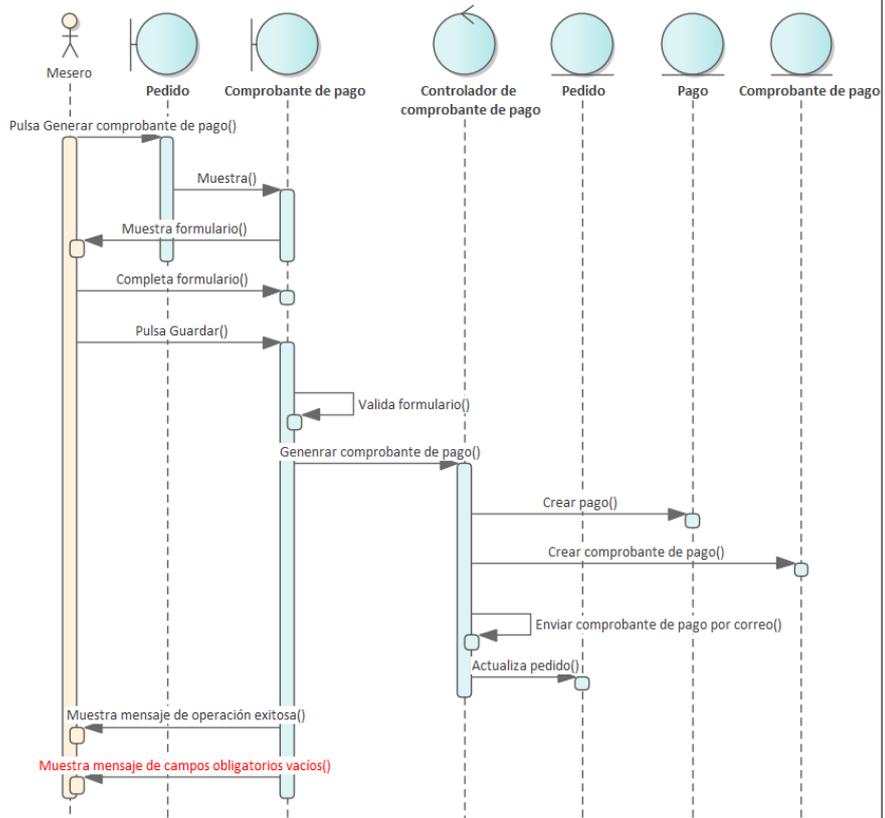
Flujo normal
 El cocinero pulsa el botón "Procesar pedido", ubicado junto a la información del pedido. El sistema muestra un mensaje de confirmación de la acción al cocinero. El sistema muestra para el usuario que realizó el pedido, sea cliente o mesero, el estado del pedido. El cocinero, una vez terminado de preparar el pedido, confirma el pedido como despachado. El sistema notifica a los meseros sobre el pedido listo para entregar a mesa.



sd Generar comprobante de pago

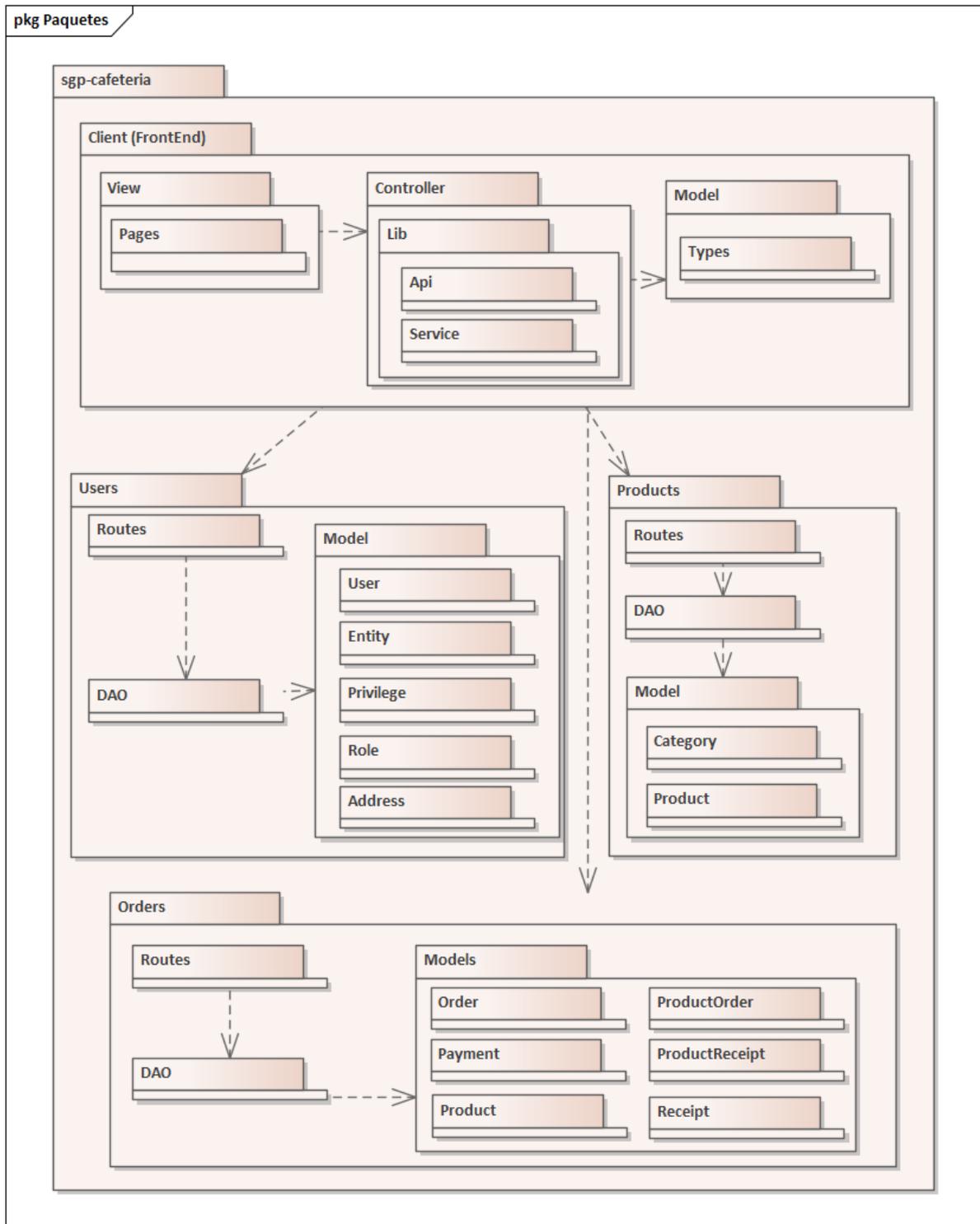
Flujo normal
 El mesero pulsa el botón "Generar comprobante de pago". El sistema muestra un formulario para registrar datos del cliente, del pago y el detalle de consumo. El mesero completa el formulario. El mesero, una vez realizado y verificado el pago, pulsa el botón "Guardar". El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. El sistema guarda el comprobante de pago y el pago. El sistema envía el comprobante de pago via correo electrónico. El sistema actualiza el estado del pedido a "Terminado" y muestra un mensaje indicando que la operación se ha realizado con éxito.

Flujo alterno
 Consumidor final
 2.1. El cliente puede decidir que en su comprobante no salgan sus datos, sino información por defecto.
 7.1. El sistema no envía comprobante de pago.
 Datos diferentes a los del registro de usuario (de existir)
 3.1. El cliente solicita al mesero que se ingresen otros datos para el comprobante de pago.
 3.2. El mesero registra los datos indicados por el cliente.
 Campos obligatorios vacíos
 6.1. El sistema muestra un mensaje indicando que los campos obligatorios

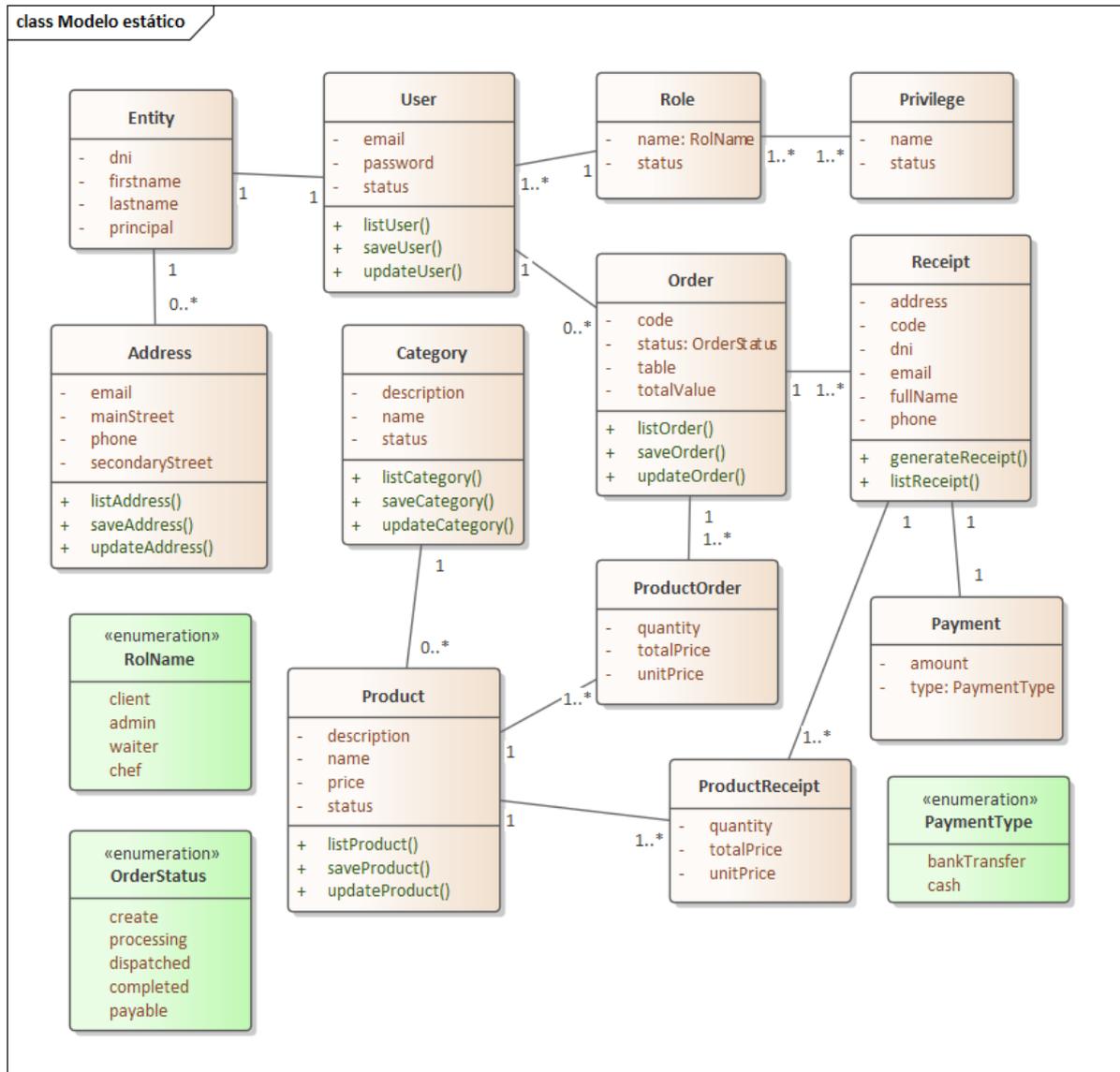


Modelado estructural mediante diagrama de paquetes y diagrama de clases

Seguidamente se presenta el diagrama de paquetes que alberga los servicios identificados. Para cada microservicio se proporciona un detalle de los modelos que lo componen; asimismo, se incluye el paquete que engloba la construcción del FrontEnd. Cada servicio responde a un estilo arquitectónico basado en capas.



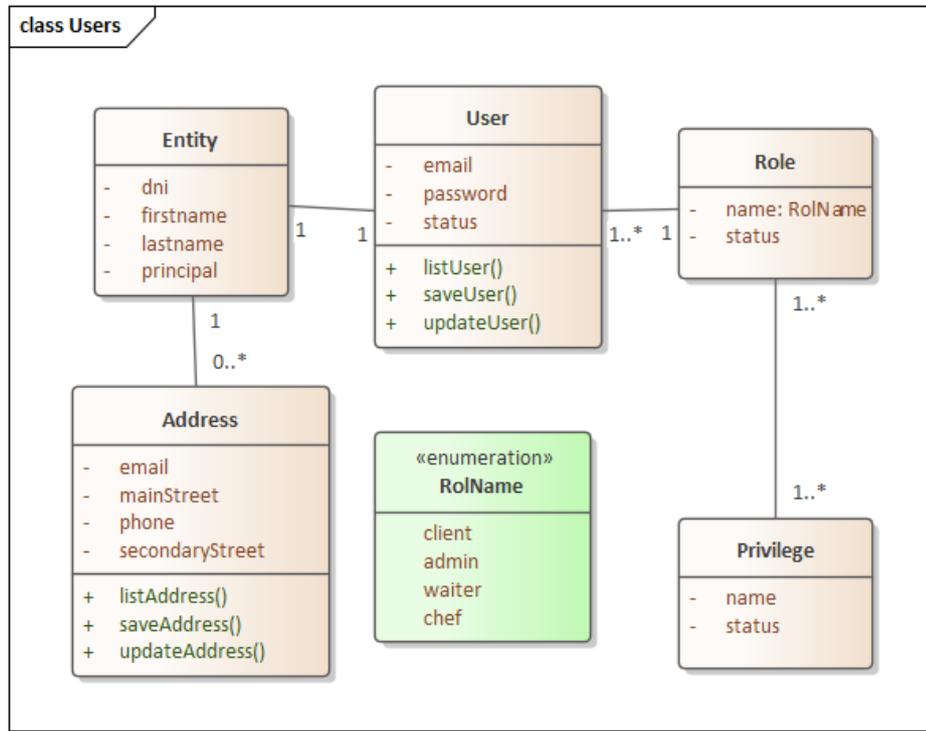
A continuación, se ve el modelo estático final del dominio o diagrama de clases basado en la representación del diagrama de conceptos, además incluye la identificación de atributos y operaciones.



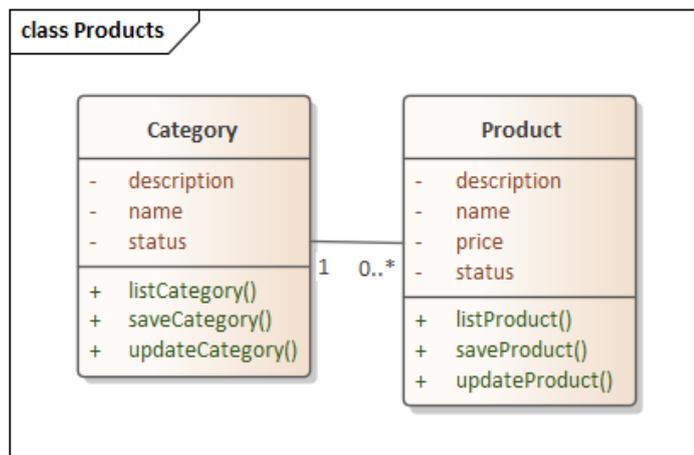
En la figura anterior se observa la integración y comunicación de todos los conceptos como si se tratara de un modelo tradicional monolítico. Las enumeraciones, conocidas como clases de tipo Enum, se han empleado con el propósito de establecer un conjunto discreto de valores. En este contexto, el nombre asignado a los roles se determina mediante la enumeración "RoleName".

El modelo estático correspondiente a cada servicio, se puede ver en las figuras a continuación:

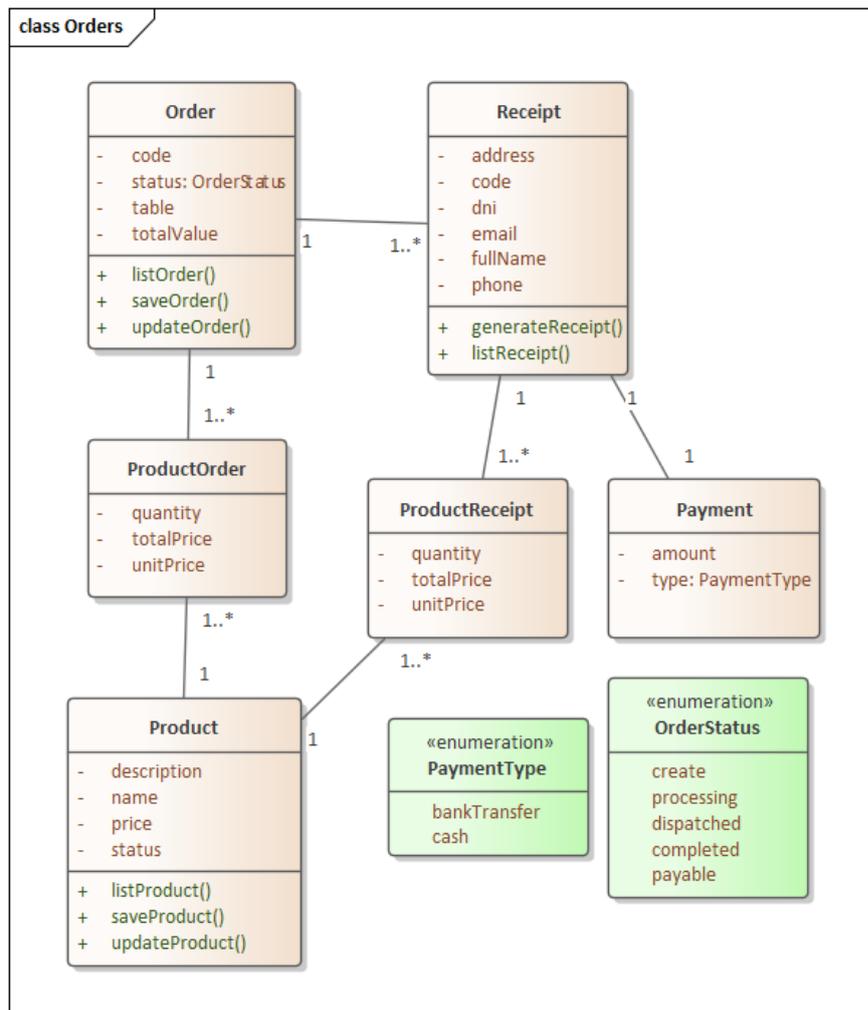
Servicio de usuarios



Servicio de productos



Servicio de pedidos



Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless)

Manual de instalación del sistema informático

1. Introducción

En el presente documento se cubren aspectos relacionados a la configuración e instalación de tecnologías y herramientas para el correcto funcionamiento de la Solución Informática en el presente TT. El código fuente se encuentra disponible en el [repositorio de GitHub de la Carrera de Computación](#).

1.1. Objetivo

El objetivo principal de este manual es describir los pasos necesarios para la configuración e instalación de las distintas tecnologías y herramientas indispensables para la ejecución de la Aplicación Web relacionada a la gestión de pedidos en cafeterías físicas de la ciudad de Loja.

1.2. Alcance

El presente manual de instalación está dirigido para todas las personas interesadas que pretendan instalar y ejecutar de manera correcta la presente solución informática.

2. Configuración del entorno en un ambiente de desarrollo

Los requisitos necesarios para el funcionamiento de la solución informática se detallan a continuación:

2.1. Docker

1. Para la instalación de Docker en un entorno local es necesario seguir las instrucciones descritas en la documentación oficial de Docker: <https://docs.docker.com/engine/install/>
2. Además, es muy importante seguir las instrucciones de post instalación: <https://docs.docker.com/engine/install/linux-postinstall/>

2.2 Kubernetes

Existen muchas formas de instalar Kubernetes en tu entorno local, pero en este caso se va a utilizar Minikube, que es una herramienta que te permite ejecutar Kubernetes de manera local de forma sencilla.

1. Para instalar Minikube, sigue las instrucciones proporcionadas por la documentación oficial de Minikube: <https://minikube.sigs.k8s.io/docs/start/>
2. Una vez instalado Minikube, necesitas iniciar el clúster de Kubernetes:

```
minikube start
```

3. Habilita el controlador Ingress NGINX, que es un objeto API que gestiona el acceso externo a los servicios en un clúster, típicamente HTTP. Ingress puede proporcionar equilibrio de carga, terminación SSL y alojamiento virtual basado en nombres. Para habilitar el controlador Ingress NGINX, ejecuta el siguiente comando:

```
minikube addons enable ingress
```

4. También necesitarás instalar kubectl, que es una herramienta de línea de comandos que te permite ejecutar comandos en clústeres de Kubernetes. Esta herramienta es necesaria para que skaffold funcione. Para instalar kubectl, sigue las instrucciones proporcionadas por la documentación oficial de Kubernetes: <https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/>
5. Opcional: Para verificar si el controlador de Ingress NGINX está habilitado, ejecuta el siguiente comando:

```
kubectl get pods -n ingress-nginx
```

2.3 Skaffold

Skaffold es una herramienta de línea de comandos que facilita el desarrollo continuo de aplicaciones de Kubernetes. Puedes iterar en el código fuente de tu aplicación localmente y luego implementar en clústeres de Kubernetes locales o remotos. Skaffold maneja el flujo de trabajo para compilar, enviar y desplegar tu aplicación. También proporciona bloques de construcción y descripciones de personalizaciones para un pipeline de CI/CD.

1. Para instalar Skaffold, sigue las instrucciones proporcionadas por la documentación oficial de Skaffold: <https://skaffold.dev/docs/install/>

2.4 Variables de Entorno

1. En primer lugar, para configurar las variables de entorno, debes hacer una copia del archivo `secrets.dev.yaml.example` y renombrarlo a `secrets.dev.yaml`. Luego, debes completar las variables con los valores correspondientes.
2. Una vez que hayas completado las variables, necesitas crear los secretos en el clúster de Kubernetes. Para hacer esto, ejecuta el siguiente comando:

```
kubectl apply -f secrets.dev.yaml
```

3. Para verificar si los secretos se crearon correctamente, ejecuta el siguiente comando:

```
kubectl get secrets
```

4. Y deberías ver algo como esto:

NAME	TYPE	DATA	AGE
activation-token-max-age	Opaque	1	2m
api-key	Opaque	1	2m
api-secret	Opaque	1	2m
cloud-name	Opaque	1	2m
jwt-secret	Opaque	1	2m

2.5. Alias de dirección IP

Crear un alias para la dirección IP del clúster de minikube es necesario para acceder a los servicios desde el navegador.

1. Después de todos los pasos anteriores, necesitas ejecutar el siguiente comando para obtener la dirección IP del clúster de minikube:

```
minikube ip
```

2. Simplemente copia la dirección IP, la necesitarás más adelante.
3. Luego, necesitas editar el archivo `/etc/hosts`. Puedes hacerlo con tu editor de texto favorito, en este caso se usará nano:

```
sudo nano /etc/hosts
```

4. Ahora debes agregar la siguiente línea:

```
<minikube ip address> cafeterias.dev
```

5. Finalmente, necesitas guardar los cambios y salir del editor de texto.

2.6. Ejecutar la aplicación

1. Para ejecutar la aplicación, debes estar en el directorio raíz del proyecto y ejecutar el siguiente comando:

```
skaffold dev
```

2. Ahora puedes acceder a la aplicación desde el navegador usando la siguiente URL: <http://cafeterias.dev>
3. Probablemente verás un mensaje de advertencia que dice que la conexión no es privada, esto se debe a que el certificado es autofirmado. Para los usuarios de Chrome, pueden hacer clic en el botón "Avanzado" y luego en "Acceder a cafeterias.dev (inseguro)", o simplemente escribir thisisunsafe.

3. Configuración del entorno en un ambiente de producción

Los pasos para el correcto funcionamiento de la solución informática en un ambiente de Producción se detallan a continuación. Sin embargo, es primordial tener en consideración que se requieren las siguientes características mínimas:

- Un host compatible de Linux. Kubernetes provee instrucciones genéricas para múltiples distribuciones de Linux.
- 2 GB o más de Memoria RAM.
- 2 CPUs o más
- Conectividad a la red.
- Entre otros requisitos que se encuentran listados en el [sitio oficial de Kubernetes](#).

Dentro de la Carrera de Computación de la Universidad Nacional de Loja se dispuso de un Servidor Privado Virtual (VPS), en el cual se instaló la solución informática con los siguientes pasos:

3.1. Instalación del motor de Docker

De acuerdo a la distribución del sistema Linux, se debe seguir las instrucciones de instalación detalladas en el [sitio oficial de Docker](#).

3.2. Instalación de las herramientas kubeadm, kubelet y kubectl

La instalación de estos paquetes permitirá:

- kubeadm: comando para la creación del clúster.
- kubelet: permite la creación de Pods y contenedores a través del clúster.
- kubectl: línea de comandos que permite la comunicación con el clúster.

Para la debida configuración de estas herramientas se debe seguir los pasos descritos en el [sitio web de Kubernetes](#).

3.3. Creación de un clúster con kudeadm

Para la inicialización de un nuevo clúster en el sistema se debe ejecutar el siguiente comando:

```
sudo kubeadm init --pod-network-cidr=192.168.0.0/16
```

Luego es necesario ejecutar los siguientes comandos para configurar kubectl:

```
mkdir -p $HOME/.kube  
  
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
  
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

3.4. Instalación del plugin de red

El clúster Kubernetes anteriormente inicializado necesitará de un plugin de red para administrar sus capacidades de red y seguridad. Existe una variedad de [plugins disponibles](#), pero para efectos del presente manual se utilizará Calico. Para instalar Calico en se debe ejecutar la siguiente sentencia en la línea de comandos:

```
kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.27.0/manifests/tigera-operator.yaml
```

3.5. Instalación del controlador de Ingress

Para la instalación del controlador del Ingress se seguirán las instrucciones detalladas en el sitio oficial de [Ingress Nginx](#). Simplemente es necesario ejecutar la siguiente sentencia:

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.8.2/deploy/static/provider/cloud/deploy.yaml
```

3.6. Creación de variables de entorno

En primer lugar, para configurar las variables de entorno, se debe hacer una copia del archivo `secrets.prod.yaml.example` y renombrarlo a `secrets.prod.yaml`. Luego, se necesita completar las variables con los valores correspondientes.

Una vez que se haya completado las variables con sus respectivos valores, se necesita crear los secretos en el clúster de Kubernetes. Para hacer esto, se ejecuta el siguiente comando:

```
kubectl apply -f secrets.prod.yaml
```

Para verificar si los secretos se crearon correctamente, ejecuta el siguiente comando:

```
kubectl get secrets
```

Y se debería ver algo como esto:

NAME	TYPE	DATA	AGE
activation-token-max-age	Opaque	1	2m
api-key	Opaque	1	2m
api-secret	Opaque	1	2m
cloud-name	Opaque	1	2m
jwt-secret	Opaque	1	2m

3.6. Aplicando los archivos de manifiesto

Para ejecutar la solución informática dentro del clúster es necesario aplicar los archivos de manifiesto ubicados en las carpetas `/infra/k8s/` e `/infra/k8s-prod/`. Pero antes de realizar esta acción, es primordial verificar que el dominio web en donde se expondrá la aplicación hacia la internet se encuentre correctamente establecido en el archivo `ingress-srv.yaml` ubicado dentro del directorio `/infra/k8s-prod/`, tal como se muestra a continuación:

```
sgp-cafeaterias - ingress-srv.yaml
12 spec:
13   tls:
14     - hosts:
15         - computacion.unl.edu.ec
16         secretName: echo-tls
17     ingressClassName: nginx
18   rules:
19     - host: computacion.unl.edu.ec
20     http:
21     paths:
```

Una vez establecido el dominio web solo queda ejecutar el siguiente comando:

```
kubectl apply -f infra/k8s && kubectl apply -f infra/k8s-prod
```

Eso es todo, la sentencia anterior creara los pods y servicios necesarios para que la aplicación funcione. Estos pasos fueron aplicados en el VPS de la carrera de Computación de la Universidad Nacional de Loja, y la dirección URL para acceder a la aplicación web es <https://computacion.unl.edu.ec/apps/>. La administración de los recursos desplegados en el clúster Kubernetes configurado se pueden ver y gestionar en el siguiente [enlace](#).

Anexo 5. Modelo de Encuesta de Satisfacción



Encuesta de satisfacción | Sistema de Gestión de Pedidos para Cafeterías de Loja

Descripción del proyecto: Proyecto de Trabajo de Titulación enfocado en la toma de pedidos en cafeterías de Loja, busca ser un sistema de apoyo y asistencia en presentación de menú o carta de platos, registro de pedidos, cobro por consumos.

Objetivo de encuesta: Evaluar el grado de satisfacción de usuarios respecto a la aplicación web para la toma de pedidos en cafetería de Loja.

Agradecemos su participación en esta encuesta. La información obtenida será usada, única y exclusivamente, con fines académicos y los datos de contacto recolectados (dirección de correo electrónico) servirá de validador de la información, NO se usará para envío posterior de información, mailing, noticias, actualizaciones ni otros fines.

Dirección de correo electrónico					
Pregunta	Respuesta				
¿La aplicación es fácil de usar?	Totalmente en desacuerdo	En desacuerdo	Neut ral	De acuerdo	Totalmente de acuerdo
¿Considera que la manera de presentar el menú y categorías es adecuada?	Totalmente en desacuerdo	En desacuerdo	Neut ral	De acuerdo	Totalmente de acuerdo
Al realizar un pedido de un plato/bebida ¿la aplicación agiliza el proceso?	Totalmente en desacuerdo	En desacuerdo	Neut ral	De acuerdo	Totalmente de acuerdo
¿Cómo calificaría el proceso de realizar un pedido?	Muy difícil	Difícil	Neut ral	Fácil	Muy Fácil
El proceso de búsqueda de productos en la aplicación, para usted, representó una dificultad de:	Muy difícil	Difícil	Neut ral	Fácil	Muy Fácil
¿Qué tan probable es que usted vuelva a utilizar la aplicación?	Nada probable	Poco probable	Neut ral	Probable	Muy probable
¿Qué tan probable es que recomiende la aplicación a otras personas?	Nada probable	Poco probable	Neut ral	Probable	Muy probable

¿Qué le pareció la experiencia de usar la aplicación?	Nada agradable	Poco agradable	Neutral	Agradable	Muy agradable
¿En qué podría mejorar la presente aplicación? Por favor, Indique su recomendación.					

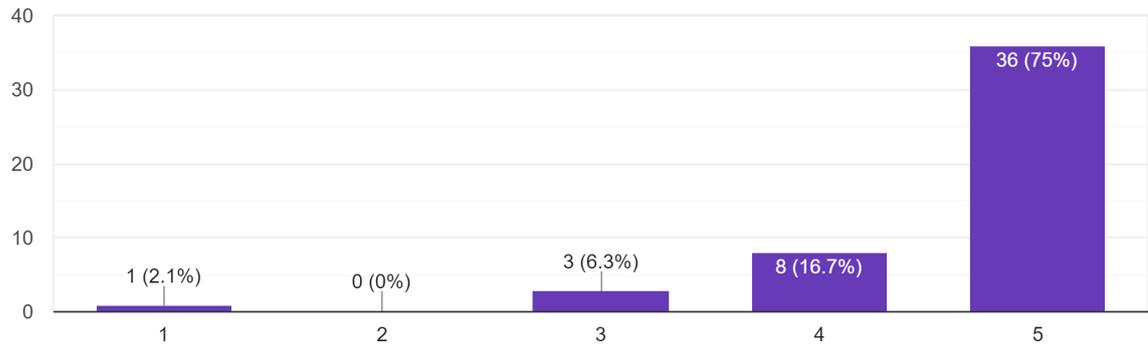
¡Gracias por tu participación en esta encuesta!

david.soto@unl.edu.ec	5	5	5	5	5	5	5	5
joseph.rios@unl.edu.ec	5	5	5	5	4	3	3	4
luis.negron@unl.edu.ec	5	5	5	5	5	5	5	5
kevin.j.jaramillo@unl.edu.ec	3	3	4	4	4	4	4	4
david.pacheco@unl.edu.ec	5	5	5	5	5	5	5	5
maalvarezp@unl.edu.ec	5	4	5	5	5	5	5	5
galoget.latorre@owasp.org	5	5	5	5	5	5	5	5
n.solano.b90@gmail.com	4	4	4	4	4	4	4	4
razuordonez@gmail.com	5	5	5	5	5	5	5	5
jhon.carrion.sys@gmail.com	4	5	5	5	4	5	5	5
puntocreativoloja@gmail.com	5	5	5	4	5	5	5	5
victorsbayas@gmail.com	5	5	5	5	5	5	5	5
leduincuenca@gmail.com	3	3	4	4	4	4	4	4
david21leiton@gmail.com	5	5	5	5	4	5	5	4
edukt15@gmail.com	5	5	5	5	5	5	5	5
davidponce.dpa@gmail.com	5	4	5	5	5	5	5	5
guillermovizcainoo@gmail.com	4	3	4	3	4	2	3	4
goussasalexander@gmail.com	3	5	3	3	2	3	3	3
livins.eduardo@gmail.com	5	4	5	5	5	5	4	5
joelemelesista@gmail.com	4	3	3	3	3	4	4	4
zj.bazurto@gmail.com	5	5	5	5	5	5	5	5
vinicioxavierlv@gmail.com	1	1	1	5	5	5	5	5
danny.cuichan@gmail.com	4	2	4	4	3	4	4	3

joselyndayanasanchezarmijos@gmail.com	5	2	5	4	4	5	5	4
dayanna.gutierrez@unl.edu.ec	4	4	2	2	3	3	3	2
josselyn.fierro@unl.edu.ec	5	5	5	5	5	5	5	5
luis.a.narvaez@unl.edu.ec	5	4	3	4	4	4	4	4
jhandry.tocto@unl.edu.ec	4	4	4	4	4	4	4	5
patricio.paredes@unl.edu.ec	5	5	5	5	5	5	5	5
jainer.pinta@unl.edu.ec	5	4	5	5	5	5	5	5
israteneda@gmail.com	5	5	5	4	5	5	5	5
miguel.rojas@unl.edu.ec	5	4	5	5	5	5	5	5
jorge.ortega.p88@gmail.com	5	5	5	5	5	5	5	5
joshdil99@gmail.com	5	5	5	5	5	5	5	5
luis.gutierrez@unl.edu.ec	5	5	5	5	5	5	5	5
josue.sauca@unl.edu.ec	5	5	5	5	2	5	5	5
josephpaezec@gmail.com	5	4	5	5	5	5	5	5
josue.ortega@unl.edu.ec	5	5	4	5	4	4	5	5

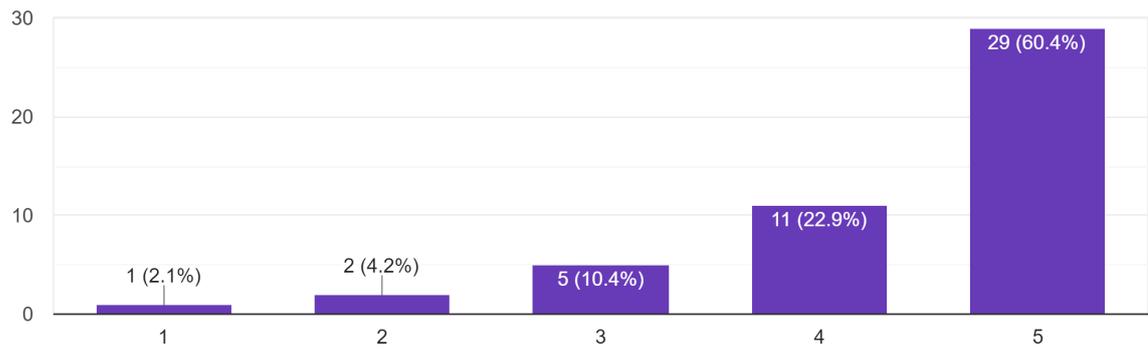
¿La aplicación es fácil de usar?

48 responses



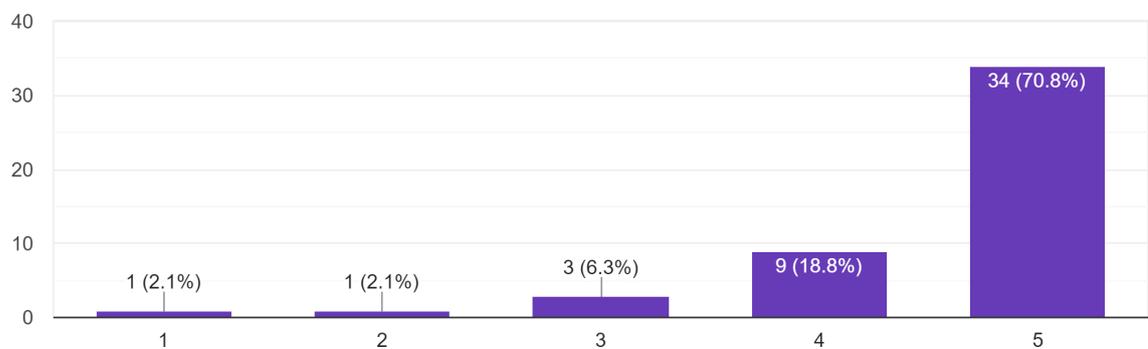
¿Considera que la manera de presentar el menú y categorías es adecuada?

48 responses



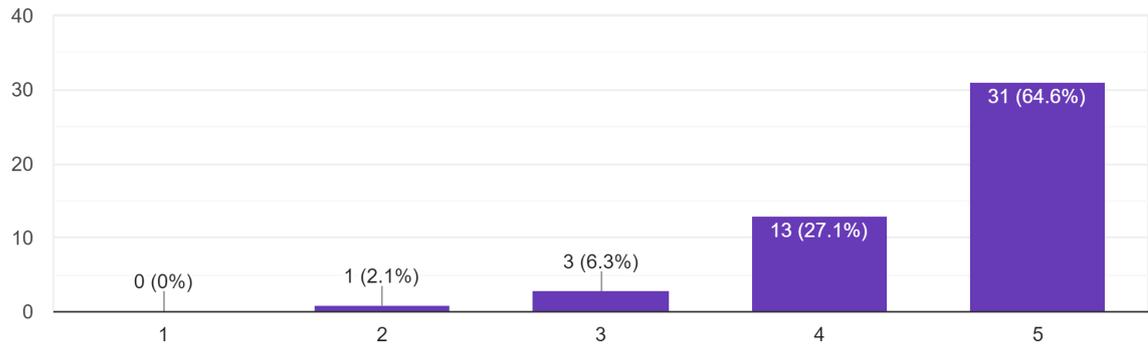
Al realizar un pedido de un plato/bebida ¿la aplicación agiliza el proceso?

48 responses



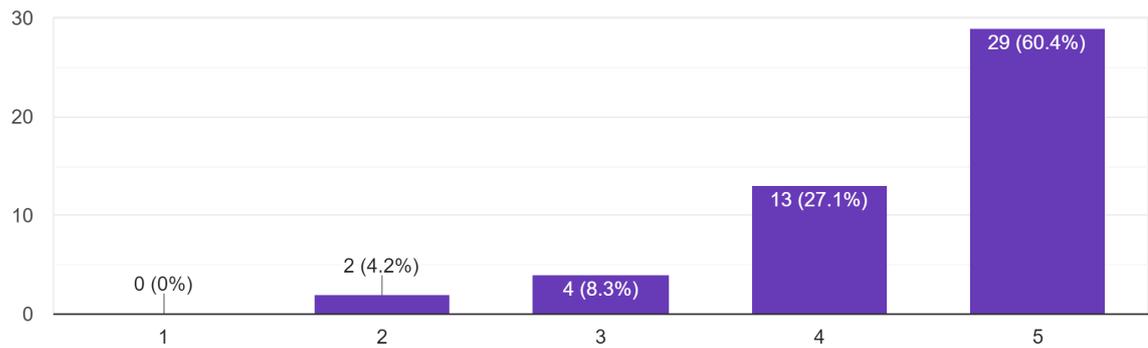
¿Cómo calificaría el proceso de realizar un pedido?

48 responses



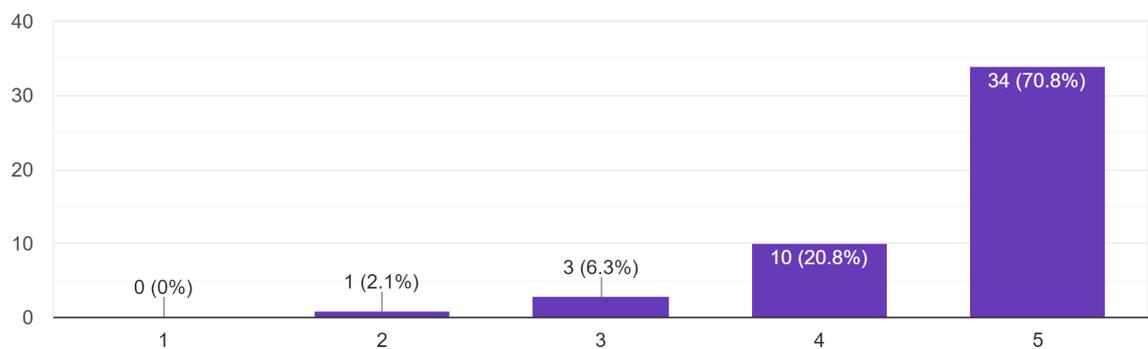
El proceso de búsqueda de productos en la aplicación, para usted, representó una dificultad de:

48 responses



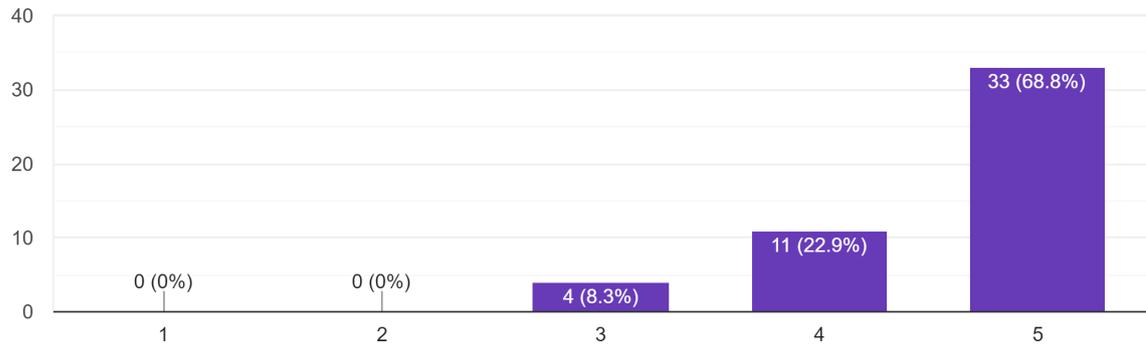
¿Qué tan probable es que usted vuelva a utilizar la aplicación?

48 responses



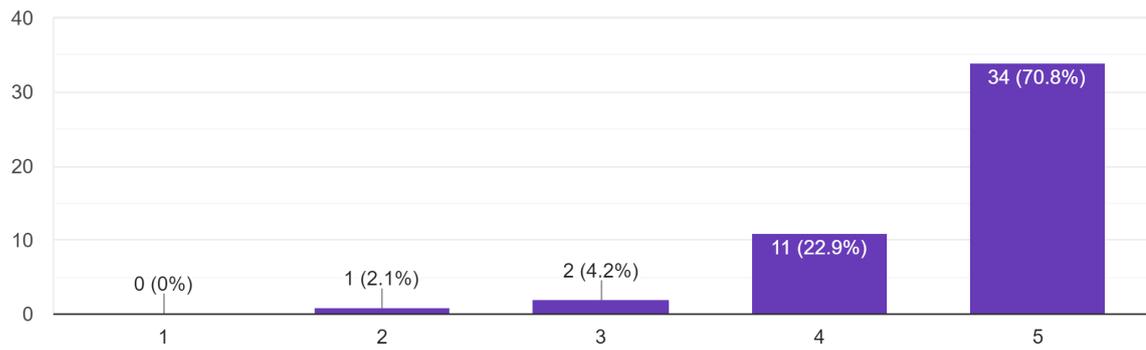
¿Qué tan probable es que recomiende la aplicación a otras personas?

48 responses



¿Qué le pareció la experiencia de usar la aplicación?

48 responses



A continuación, se muestra el resumen final de los resultados de la Encuesta de Satisfacción que se aplicó a los usuarios:

Tabla 22. Resumen de resultados de la Encuesta de Satisfacción

Pregunta	1	2	3	4	5	Promedio de Aceptación
¿La aplicación es fácil de usar?	2.08%	0.00%	6.25%	16.67%	75.00%	97.92%
¿Considera que la manera de presentar el menú y categorías es adecuada?	2.08%	4.17%	10.42%	22.92%	60.42%	93.75%
Al realizar un pedido de un plato/bebida ¿la aplicación agiliza el proceso?	2.08%	2.08%	6.25%	18.75%	70.83%	95.83%
¿Cómo calificaría el proceso de realizar un pedido?	0.00%	2.08%	6.25%	27.08%	64.58%	97.92%
El proceso de búsqueda de productos en la aplicación, para usted, representó una dificultad de:	0.00%	4.17%	8.33%	27.08%	60.42%	95.83%
¿Qué tan probable es que usted vuelva a utilizar la aplicación?	0.00%	2.08%	6.25%	20.83%	70.83%	97.92%
¿Qué tan probable es que recomiende la aplicación a otras personas?	0.00%	0.00%	8.33%	22.92%	68.75%	100.00%
¿Qué le pareció la experiencia de usar la aplicación?	0.00%	2.08%	4.17%	22.92%	70.83%	97.92%
Promedio Final de Aceptación						97.14%

Anexo 7. Acta de validación de pruebas del sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless)

Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless)

Acta de validación de Pruebas de la Aplicación Web

1. Introducción

En la ciudad de Loja, se realizaron reuniones de trabajo para comparar flujos de trabajo entre las operaciones manuales en cafeterías de la ciudad respecto de las automatizadas logradas luego de la implementación de un sistema informático para la toma de pedidos en los locales. Este proyecto corresponde al Trabajo de Titulación (TT) presentado por la estudiante Maria Encalada. Se inició con una breve presentación del modelo desarrollado, así como el cumplimiento de casos de usos, tomando 3 escenarios para el contraste, estos fueron: 1) Registro de producto nuevo y actualización de la carta/menú; 2) Toma del pedido, incluyendo vista de menú, toma de orden, paso a cocina y entrega del pedido en la mesa; y 3) Cobro por consumo, es decir, identificación de consumo y cobro de cuentas separadas por múltiples consumos en una misma mesa.

2. Agenda

1.	Ejecución de pruebas de aceptación de los escenarios en la forma automatizada (Solución informática desarrollada).
2.	Ejecución de pruebas de los escenarios en la forma manual (Procedimiento actual llevado por 3 cafeterías de Loja).
3.	Resolución de firmas de aprobación de pruebas de la Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja.

3. Desarrollo

3.1. Prueba de los escenarios en la forma automatizada

Aquí se describen los casos de prueba de aceptación contruidos en base a los casos de uso que responden a los escenarios citados en la introducción de este documento. Se detalla una sola vez, dado que el flujo es el mismo indistintamente de la cafetería en donde se pruebe.

Caso de Prueba de Aceptación CU09		Nro: 1
Título	Registrar categoría	
Pantalla	Categorías	
Área de prueba	Aplicación Web	
Objetivos	Registro de nuevas categorías	
Acciones a Realizar		

<ol style="list-style-type: none"> 1. El mesero pulsa el botón "Registrar categoría". 2. El sistema muestra un formulario registrar categoría. 3. El mesero ingresa la información de la categoría (nombre y descripción) 4. El mesero pulsa el botón "Guardar". 			
Resultado(s) Esperado(s)			
<ul style="list-style-type: none"> - El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. - El sistema guarda la categoría muestra un mensaje indicando que el registro ha sido exitoso. 			
Validado	Si	No	Parcialmente
Observación(es)			
Nombre tester			
Tiempo empleado	3 min.		
Referencia al caso de prueba manual	Nro 1. Registro de producto nuevo (y actualización de la carta/menú)		

Caso de Prueba de Aceptación CU12			Nro: 2
Título	Buscar categoría		
Pantalla	Categorías		
Área de prueba	Aplicación Web		
Objetivos	Buscar categoría por denominación		
Acciones a Realizar			
<ol style="list-style-type: none"> 1. El mesero ingresa el nombre de la categoría en el campo ubicado en la parte superior de la lista de categorías. 2. El mesero presiona el botón "Buscar" 			
Resultado(s) Esperado(s)			
<ul style="list-style-type: none"> - El sistema busca la categoría y muestra. 			
Validado	Si	No	Parcialmente
Observación(es)			
Nombre tester			
Tiempo empleado	3 min.		
Referencia al caso de prueba manual	Nro 1. Registro de producto nuevo (y actualización de la carta/menú)		

Caso de Prueba de Aceptación CU13		Nro: 3	
Título	Registrar producto		
Pantalla	Productos		
Área de prueba	Aplicación Web		
Objetivos	Registro de nuevos productos		
Acciones a Realizar			
<ol style="list-style-type: none"> 1. El mesero pulsa el botón "Registrar producto". 2. El sistema muestra un formulario de registro de producto. 3. El mesero ingresa la información (nombre, descripción y precio). 4. El mesero pulsa el botón "Guardar". 			
Resultado(s) Esperado(s)			
<ul style="list-style-type: none"> - El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. - El sistema guarda el producto y muestra un mensaje indicando que el registro ha sido exitoso. 			
Validado	Si	No	Parcialmente
Observación(es)			
Nombre tester			
Tiempo empleado	3 min.		
Referencia al caso de prueba manual	Nro 1. Registro de producto nuevo (y actualización de la carta/menú)		

Caso de Prueba de Aceptación CU17		Nro: 4	
Título	Agregar productos al pedido		
Pantalla	Menú		
Área de prueba	Aplicación Web		
Objetivos	Agregar productos a la lista de un pedido		
Acciones a Realizar			
<ol style="list-style-type: none"> 1. El usuario pulsa el botón "Agregar a pedido". 2. El sistema muestra la opción de elegir la cantidad de productos. 3. El usuario presiona el botón "Agregar" 			
Resultado(s) Esperado(s)			
<ul style="list-style-type: none"> - El sistema agrega a la lista del pedido y muestra un mensaje indicando que la operación se realizó con éxito. 			
Validado	Si	No	Parcialmente
Observación(es)			
Nombre tester			

Tiempo empleado	3 min.
Referencia al caso de prueba manual	Nro 2. Toma del pedido (incluyendo vista de menú, toma de orden, paso a cocina y entrega del pedido en la mesa)

Caso de Prueba de Aceptación CU18		Nro: 5	
Título	Quitar producto al pedido		
Pantalla	Menú		
Área de prueba	Aplicación Web		
Objetivos	Quitar productos a la lista de un pedido		
Acciones a Realizar			
1. El usuario pulsa el botón "Quitar", ubicado junto al producto que desea retirar.			
Resultado(s) Esperado(s)			
<ul style="list-style-type: none"> - El sistema remueve el producto de la lista - El sistema muestra la lista actualizada. 			
Validado	Si	No	Parcialmente
Observación(es)			
Nombre tester			
Tiempo empleado	3 min.		
Referencia al caso de prueba manual	Nro 2. Toma del pedido (incluyendo vista de menú, toma de orden, paso a cocina y entrega del pedido en la mesa)		

Caso de Prueba de Aceptación CU19		Nro: 6	
Título	Generar pedido		
Pantalla	Resumen del Pedido		
Área de prueba	Aplicación Web		
Objetivos	Confirmar pedido		
Acciones a Realizar			
1. El usuario pulsa el botón "Generar pedido".			
Resultado(s) Esperado(s)			
<ul style="list-style-type: none"> - El sistema cambia el estado del pedido a "Pendiente" - El sistema envía la orden a cocina. - El sistema muestra el detalle del pedido al cliente. 			
Validado	Si	No	Parcialmente
Observación(es)			

Nombre tester	
Tiempo empleado	3 min.
Referencia al caso de prueba manual	Nro 2. Toma del pedido (incluyendo vista de menú, toma de orden, paso a cocina y entrega del pedido en la mesa)

Caso de Prueba de Aceptación CU20		Nro: 7
Título	Procesar pedido	
Pantalla	Detalles del Pedido	
Área de prueba	Aplicación Web	
Objetivos	Procesar el estado del pedido	
Acciones a Realizar		
1. El cocinero pulsa el botón "Procesar pedido", ubicado junto a la información del pedido.		
Resultado(s) Esperado(s)		
<ul style="list-style-type: none"> - El sistema muestra un mensaje de confirmación de la acción al cocinero. - El sistema muestra para el usuario que realizó el pedido, sea cliente o mesero, el estado del pedido. - El cocinero, una vez terminado de preparar el pedido, confirma el pedido como despachado. - El sistema notifica a los meseros sobre el pedido listo para entregar a mesa. 		
Validado	Si	No
		Parcialmente
Observación(es)		
Nombre tester		
Tiempo empleado	3 min.	
Referencia al caso de prueba manual	Nro 2. Toma del pedido (incluyendo vista de menú, toma de orden, paso a cocina y entrega del pedido en la mesa)	

Caso de Prueba de Aceptación CU21		Nro: 8
Título	Generar comprobante de pago	
Pantalla	Detalles del Pedido	
Área de prueba	Aplicación Web	
Objetivos	El mesero cobra por los productos consumidos	
Acciones a Realizar		
<ol style="list-style-type: none"> 1. El mesero pulsa el botón "Generar comprobante de pago". 2. El sistema muestra un formulario para registrar datos del cliente, del pago y el detalle de consumo. 3. El mesero completa el formulario. 4. El mesero, una vez realizado y verificado el pago, pulsa el botón "Guardar". 		

Resultado(s) Esperado(s)			
<ul style="list-style-type: none"> - El sistema valida que el formulario este completo y correcto de acuerdo al tipo de información que se espera en cada campo. - El sistema guarda el comprobante de pago y el pago. - El sistema envía el comprobante de pago vía correo electrónico. - El sistema actualiza el estado del pedido a "Terminado" y muestra un mensaje indicando que la operación se ha realizado con éxito. 			
Validado	Si	No	Parcialmente
Observación(es)			
Nombre tester			
Tiempo empleado	3 min.		
Referencia al caso de prueba manual	Nro 3. Cobro por consumo (identificación de consumo y cobro de cuentas separadas por múltiples consumos en una misma mesa)		

3.2. Pruebas de escenarios de forma manual

Aquí se presentan las pruebas de los escenarios ejecutadas en forma manual, es decir, el flujo de actividades desarrolladas por las cafeterías observadas. Estos casos representan una copia exacta de las actividades ejecutadas en escenarios reales por los locales en su interacción física sin la mediación del sistema informático.

La comparación se realizó en tres cafeterías: 1) Cafés de Loja; 2) La llorona; y 3) cafetería universitaria Carrera de Turismo.

Caso 1

Caso de Prueba		Nro.: 1
Título	Registro de producto nuevo (y actualización de la carta/menú)	
Área de prueba	Local físico de cafés de Loja	
Objetivos	Registrar un nuevo producto en el menú	
Acciones Realizadas	Tiempo empleado	Encargado
Abrir archivo digital	30 s.	Mesero/a

Buscar la categoría al que corresponde el producto	2 min.	Mesero/a
Si no existe la categoría, registrar categoría	1 min. 30 s.	Mesero/a
Registrar un nuevo producto	1 min.	Mesero/a
Verificar que el archivo mantenga el formato y no haya saltos de página en el medio del detalle de un producto.	2 min.	Mesero/a
Imprimir la actualización del menú.	1 h.	Mesero/a
Tiempo total empleado	1 h. 7 min.	

Caso de Prueba		Nro.: 2
Título	Toma del pedido (incluyendo vista de menú, toma de orden, paso a cocina y entrega del pedido en la mesa)	
Área de prueba	Local físico de cafeterías	
Objetivos	Tomar un nuevo pedido y entregarlo en la mesa.	
Acciones Realizadas	Tiempo empleado	Encargado
Identificar el número de visitantes en mesa	30 s.	Mesero/a
Acercar las cartas con el menú, según el número de visitantes	30 s.	Mesero/a

Tiempo de espera para que las cartas sean revisadas y productos seleccionados	3 min.	Mesero/a
Toma de pedido, junto con el número de mesa	3 min.	Mesero/a
Comunicación de pedido en cocina	2 min.	Mesero/a
procesamiento del pedido.	10 min.	Cocinero/a
Entregar pedido desde cocina hasta la mesa.	2 min.	Mesero/a
Tiempo total empleado	21 min.	

Caso de Prueba		Nro.: 3
Título	Cobro por consumo (identificación de consumo y cobro de cuentas separadas por múltiples consumos en una misma mesa)	
Área de prueba	Local físico de cafeterías	
Objetivos	Registrar el pago de una venta y generar el comprobante de la venta	
Acciones Realizadas	Tiempo empleado	Encargado
Identificar detalle del consumo (comunicar el valor total)	2 min.	Mesero/a

Realizar registro de cobro (registrar datos del cliente)	1 min.	Mesero/a
Para pago en una sola cuenta por consumo: registrar ingreso en caja y cobrar al cliente.	3 min.	Mesero/a
Para pago en cuentas separadas por consumo: Separar consumos por productos, registrar entradas en caja y cobrar al cliente.	5 min.	Mesero/a
Verificar que todos los productos que componen el consumo fueron cancelados.	1 min.	Mesero/a
Imprimir el comprobante de venta	2 min.	Mesero/a
Tiempo total empleado	15 min.	

Caso 2

Caso de Prueba		Nro.: 1
Título	Registro de producto nuevo (y actualización de la carta/menú)	
Área de prueba	Local físico de cafés de Loja	
Objetivos	Registrar un nuevo producto en el menú	
Acciones Realizadas	Tiempo empleado	Encargado
Abrir archivo digital	30 s.	Mesero/a

Buscar la categoría al que corresponde el producto	3 min.	Mesero/a
Si no existe la categoría, registrar categoría	1 min.	Mesero/a
Registrar un nuevo producto	1 min.	Mesero/a
Verificar que el archivo mantenga el formato y no haya saltos de página en el medio del detalle de un producto.	2 min.	Mesero/a
Imprimir la actualización del menú.	1 h.	Mesero/a
Tiempo total empleado	1 h. 7 min. 30 s.	

Caso de Prueba		Nro.: 2
Título	Toma del pedido (incluyendo vista de menú, toma de orden, paso a cocina y entrega del pedido en la mesa)	
Área de prueba	Local físico de cafeterías	
Objetivos	Tomar un nuevo pedido y entregarlo en la mesa.	
Acciones Realizadas	Tiempo empleado	Encargado
Identificar el número de visitantes en mesa	1 min.	Mesero/a
Acercar las cartas con el menú, según el número de visitantes	1 min.	Mesero/a

Tiempo de espera para que las cartas sean revisadas y productos seleccionados	3 min.	Mesero/a
Toma de pedido, junto con el número de mesa	3 min.	Mesero/a
Comunicación de pedido en cocina	2 min.	Mesero/a
Procesamiento del pedido.	7 min.	Cocinero/a
Entregar pedido desde cocina hasta la mesa.	2 min.	Mesero/a
Tiempo total empleado	19 min.	

Caso de Prueba		Nro.: 3
Título	Cobro por consumo (identificación de consumo y cobro de cuentas separadas por múltiples consumos en una misma mesa)	
Área de prueba	Local físico de cafeterías	
Objetivos	Registrar el pago de una venta y generar el comprobante de la venta	
Acciones Realizadas	Tiempo empleado	Encargado
Identificar detalle del consumo (comunicar el valor total)	3 min.	Mesero/a

Realizar registro de cobro (registrar datos del cliente)	1 min.	Mesero/a
Para pago en una sola cuenta por consumo: registrar ingreso en caja y cobrar al cliente.	3 min.	Mesero/a
Para pago en cuentas separadas por consumo: Separar consumos por productos, registrar entradas en caja y cobrar al cliente.	5 min.	Mesero/a
Verificar que todos los productos que componen el consumo fueron cancelados.	1 min.	Mesero/a
Imprimir el comprobante de venta	1 min.	Mesero/a
Tiempo total empleado	14 min.	

Caso 3

A diferencia de los casos anteriores, no dispone de una carta impresa con el menú disponible, además, sus platos cambian de un día a otro. Dispone de 2 menús, 1) cafetería, pensado para el consumo de estudiantes; y 2) almuerzos, pensado para docentes y administrativos.

Con este antecedente, para este caso no se hace registro de nuevos productos, el flujo consiste en una categorización visual de los clientes y un anuncio verbal del menú disponible. Este proceso implica que las personas que comunican los platos/bebidas en oferta, memoricen cada día la nueva disponibilidad.

Caso de Prueba		Nro.: 1
Título	Registro de producto nuevo (y actualización de la carta/menú)	
Área de prueba	Cafetería Universitaria gestionada por la carrera de Turismo	

Objetivos	Registrar un nuevo producto en el menú	
Acciones Realizadas	Tiempo empleado	Encargado
Borrado de los tableros (pizarras de acetato).	1 min.	Mesero/a
Escribir las categoría y lista de productos que corresponden para este día	6 min.	Mesero/a
Ubicar los tableros en los lugares frecuentes.	3 min.	Mesero/a
Tiempo total empleado	10 min.	

Caso de Prueba	Nro.: 2	
Título	Toma del pedido (incluyendo vista de menú, toma de orden, paso a cocina y entrega del pedido en la mesa)	
Área de prueba	Cafetería Universitaria gestionada por la carrera de Turismo	
Objetivos	Tomar un nuevo pedido y entregarlo.	
Acciones Realizadas	Tiempo empleado	Encargado
Identificar el grupo al que pertenece el visitante	30 s.	Mesero/a
Mencionar de forma oral los productos que dispone de acuerdo a la categorización que estimó.	2 min.	Mesero/a

Dar aclaraciones de los precios y detalles incluidos en cada producto.	1 min.	Mesero/a
Toma de pedido	30 s.	Mesero/a
Cobrar por pedido	2 min.	Mesero/a
Comunicación de pedido en cocina	1 min.	Mesero/a
Procesamiento del pedido.	5 min.	Cocinero/a
Entregar pedido desde cocina hasta la mesa.	1 min.	Mesero/a
Tiempo total empleado	13 min.	

Esta cafetería no tiene el proceso de imprimir comprobantes ni factura, tampoco permite el pago mediante transferencias, sino que se destaca por el empleo de pago en efectivo una vez hecho el pedido, es decir, antes de recibirlo.

3.3. Resolución de firmas de aprobación de pruebas del sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja

Tras realizar las reuniones respectivas que permitieron verificar el correcto funcionamiento del sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja, se procedió a realizar las firmas de aprobación de pruebas con los involucrados en esta actividad.

Gerente - Administrador

Estudiante

Maria Anabel Encalada Córdova

Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless)

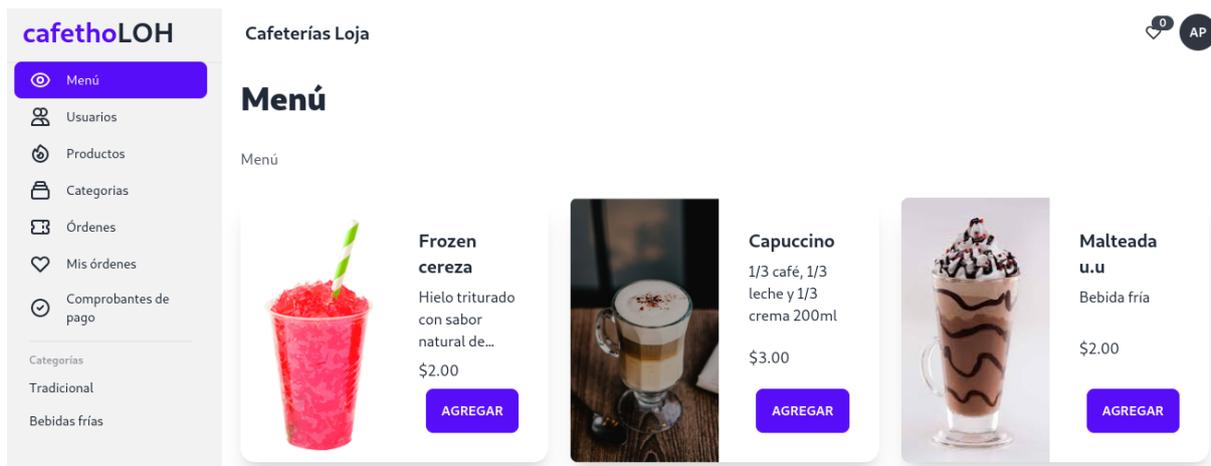
Manual de usuario del sistema informático

Introducción

Este manual de usuario tiene como propósito brindar una guía detallada sobre los procedimientos y pasos esenciales para la correcta ejecución de las diversas funcionalidades integradas en el sistema informático. Este sistema está diseñado para gestionar eficientemente los pedidos en las cafeterías locales de la ciudad de Loja, haciendo uso de una arquitectura distribuida basada en microservicios y/o serverless. A lo largo de este manual, se abordarán las múltiples perspectivas asociadas a los roles desempeñados en este proyecto, con el objetivo de proporcionar una comprensión completa y holística de su funcionamiento y aplicación.

Procesos principales

Al ingresar al sistema, se muestran los productos registrados y sus detalles. En la parte lateral izquierda se encuentra un menú de navegación de las funciones del sistema con variación según el rol del usuario en sesión. Al final de las opciones, aparecen las categorías de los productos, una forma de filtrar los resultados del menú.



1. Administrador

Como parte de los permisos del rol administrador consta la gestión de usuario (ver, crear, modificar, dar de baja). Desde la vista Listar usuarios, se puede ver todas las cuentas creadas, se puede cambiar el estado de cada una de ellas, así como modificar. Al presionar el botón **Registrar usuario** aparece el botón un formulario para crear una nueva cuenta.

Usuarios

 REGISTRAR USUARIO

Lista de usuarios registrados.

NOMBRE	CORREO	CÉDULA	ROL	ESTADO	ACCIONES
Administrador Principal	admin@localhost.com	9999999999	Administrador	<input checked="" type="checkbox"/>	CAMBIAR NOMBRE
Belen Castro	mencalada@openlab.ec	2222222222	Cocinero	<input checked="" type="checkbox"/>	CAMBIAR NOMBRE
Karen Cat	maria.a.encalada@unl.edu.ec	1102114160	Cliente	<input checked="" type="checkbox"/>	CAMBIAR NOMBRE

Mostrando 1-3 de 3 « 1 »

El formulario de registro de usuario pide información de identidad y credenciales de acceso. Al presionar el botón Registrar se lanza la petición y se crea la cuenta.

Nueva Cuenta de Usuario

[← REGRESAR](#)

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Veniam, earum.

Nombre o denominación	Apellido
<input type="text" value="Jenny"/>	<input type="text" value="Pérez"/>
Cédula o RUC	Correo electrónico
<input type="text" value="0707031662"/>	<input type="text" value="maria.a.encalada@unl.edu.ec"/>
Clave	Rol
<input type="password" value="....."/>	<input type="text" value="Mesero"/>
REGISTRAR	

La gestión de categorías es un rol compartido entre el rol administrador y mesero y consta de listar, crear, modificar y dar de baja.

Categorías

+ REGISTRAR CATEGORÍA

Categorías registradas

NOMBRE	DESCRIPCIÓN (OPCIONAL)	ESTADO	
Tradicional	Comida tradicional	<input checked="" type="checkbox"/>	EDITAR

Mostrando 1-1 de 1

« 1 »

Desde la vista de categorías, presionar Registrar categoría, esto presenta un formulario con los campos para ingresar la información de la categoría.

Nueva Categoría

← REGRESAR

Registro de una nueva categoría.

Nombre - denominación

Descripción (opcional)

REGISTRAR

La gestión de productos incluye funciones de crear, modificar, listar y dar de baja. En la vista de listado se permite buscar por el nombre del producto.

Productos

+ REGISTRAR PRODUCTO

Lista de productos registrados.

NOMBRE	PRECIO	CATEGORÍA	ESTADO	ACCIONES
 Malteada u.u Bebida fría	\$2.00	Tradicional	<input checked="" type="checkbox"/>	EDITAR

Mostrando 1-1 de 1

« 1 »

Al presionar el botón Registrar producto, se presenta un formulario para ingresar la información del plato/bebida. Por cada producto se puede agregar una única foto y el formato de archivo permitido es solo imagen, no acepta otros formatos.

Guardar Producto

Registro de un nuevo producto.

[← REGRESAR](#)

Nombre - denominación
Capuccino

Descripción
1/3 café, 1/3 leche y 1/3 crema
200ml

Precio del producto (\$) \$3

Categoría Bebidas frías

Arrastra y suelta archivos aquí (.jpg/.png) (Tamaño max.: 1 MB)
o
BUSCAR ARCHIVOS

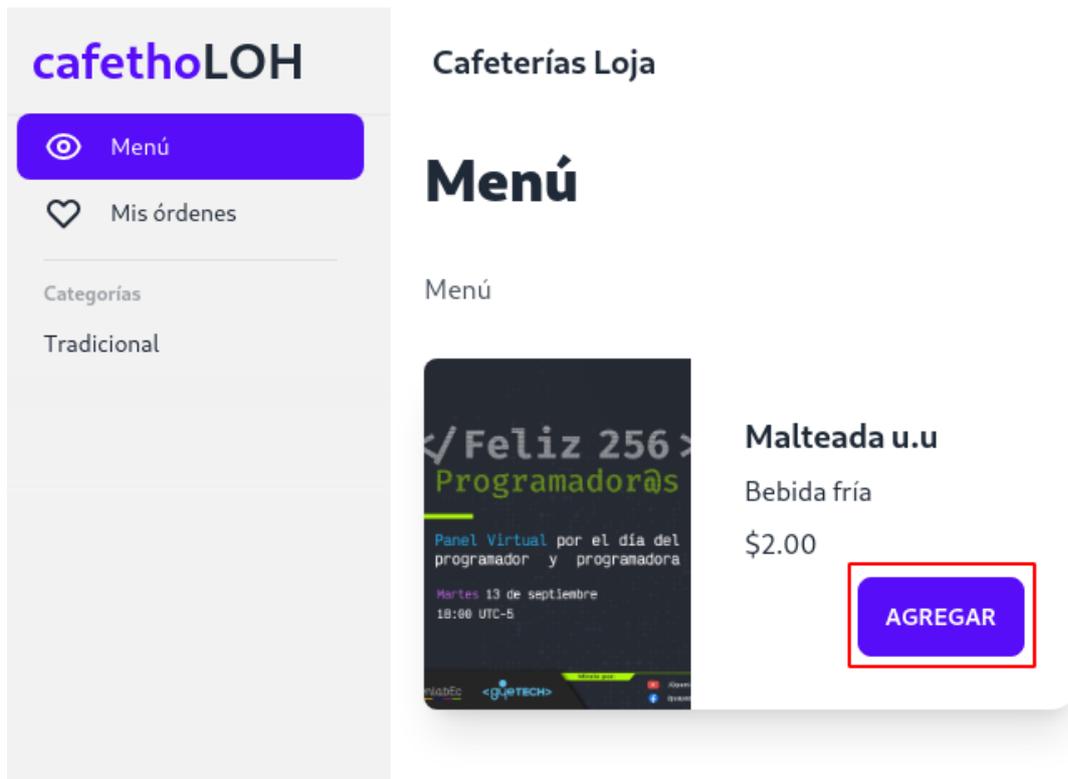
Archivos subidos
¡Has subido 1 archivo!



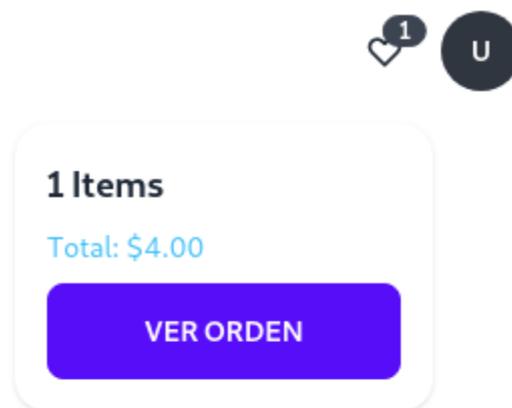
REGISTRAR

2. Cliente

La vista principal del cliente le lleva al menú disponible de la cafetería, además muestra opciones en la izquierda con un listado de las categorías registradas, al presionar sobre alguna de ellas se presentarán solo los productos que correspondan a la selección.



Por cada producto y junto a este, aparece el botón Agregar, este permite que se incluya el plato/bebida en el carrito de compras. Cada vez que se agregue un nuevo producto, la lista de items y el precio total se actualizará mostrando información como lo que se ve en la figura.



En el detalle del carrito se presenta la opción Ver orden, al seleccionar se presenta una vista general de los productos agregados con la posibilidad de editar la cantidad, además del total a pagar por el consumo.

Mi orden

Detalles de orden

DETALLES DE PRODUCTO	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL	Resumen de orden	
Malteada u.u Bebida fría Eliminar	- 2 +	\$2.00	\$4.00	ITEMS 1	\$4.00
				COSTO TOTAL	\$4.00
				REALIZAR ORDEN	

Al presionar el botón Realizar órdenes, se lanza la notificación al rol mesero sobre la nueva orden. Para el cliente, en la vista de Mis órdenes, ubicada en la derecha, aparece la información de las órdenes realizadas, así como su estado y la opción Ver.

cafethoLOH

Menú

Mis órdenes

Categorías

Tradicional

Cafeterías Loja

Mis órdenes

Lista de mis órdenes realizadas.

NUEVA ORDEN

CÓDIGO	FECHA DE ORDEN	VALOR TOTAL	N° DE MESA	ESTADO	ACCIONES
HKT1U2	10 de noviembre de 2022 a las 08:51	\$6.00	Sin especificar	Creado	VER

Mostrando 1-1 de 1

1

En la vista del detalle de la orden se presenta el desglose de los productos pedidos, precio unitario y total.

Orden

[← REGRESAR](#)

Detalle de orden

HkT1U2 Código de referencia	Creado Estado	\$6.00 Total	No especificado N° de mesa
---------------------------------------	-------------------------	------------------------	--------------------------------------

CANTIDAD	PRODUCTO	PRECIO UNITARIO	PRECIO TOTAL
3	 Malteada u.u Bebida fría	\$2.00	\$6.00
TOTAL A PAGAR:			\$6.00

Mostrando 1-1 de 1 « 1 »

3. Cocinero

Para el rol Cocinero, siempre que un cliente crea una nueva orden, le aparece una actualización en la lista de órdenes por procesar/preparar. Al presionar el botón Ver se muestran todos los detalles.

Órdenes

Lista de órdenes registradas.

CÓDIGO	FECHA DE ORDEN	VALOR TOTAL	N° DE MESA	ESTADO	ACCIONES
HkT1U2	10 de noviembre de 2022 a las 08:51	\$6.00	Sin especificar	Creado	VER

Mostrando 1-1 de 1 « 1 »

En el detalle de la orden, puede cambiar el estado, al ejecutar esta acción, se entiende que el pedido fue tomado y pasado a un estado de “preparación”, esta actualización será visible para el cliente.

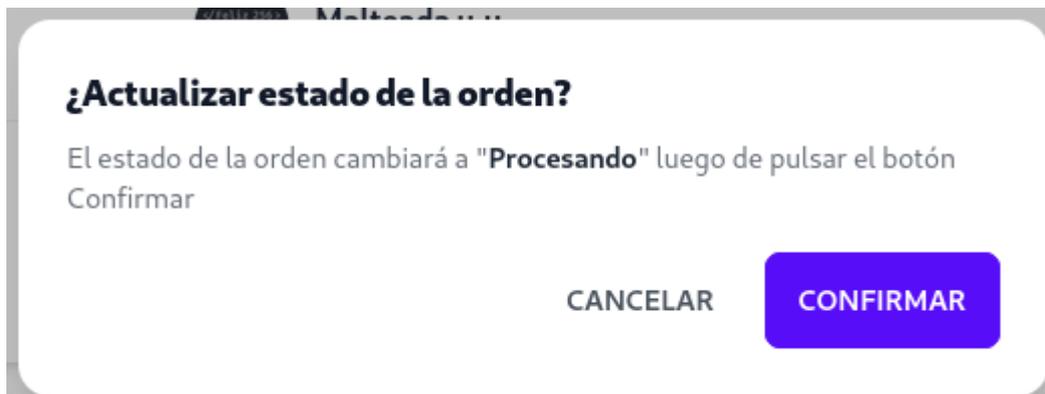
Orden

← REGRESAR

Detalle de orden

Hkt1U2 Código de referencia	Creado Estado	\$6.00 Total	No especificado N° de mesa
CANTIDAD	PRODUCTO	PRECIO UNITARIO	PRECIO TOTAL
3	 Malteada u.u. Bebida fría	\$2.00	\$6.00
TOTAL A PAGAR:			\$6.00
Mostrando 1-1 de 1			« 1 »

Antes de cambiar el estado, el sistema hace una confirmación de acción. Una vez presionado el botón Confirmar, el estado es cambiado.



El estado del pedido ha sido modificado, al repetir la acción el estado vuelve a ser cambiado. Es importante tener presente que es estado Procesando y Despachado sólo pueden ser hechos por el rol Cocinero, mientras no esté Despachado, no le aparece la orden al mesero.

Orden

← REGRESAR

Detalle de orden

Hkt1U2 Código de referencia	Procesando Estado	\$6.00 Total	No especificado N° de mesa
CANTIDAD	PRODUCTO	PRECIO UNITARIO	PRECIO TOTAL
3	 Malteada u.u. Bebida fría	\$2.00	\$6.00
TOTAL A PAGAR:			\$6.00
Mostrando 1-1 de 1			« 1 »

4. Mesero

Una vez el cocinero haya cambiado el estado de la orden a Despachado, el pedido le aparece en lista al rol Mesero para que pueda entregarlo al cliente.

Órdenes

Lista de órdenes registradas.

CÓDIGO	FECHA DE ORDEN	VALOR TOTAL	N° DE MESA	ESTADO	ACCIONES
HkT1U2	10 de noviembre de 2022 a las 08:51	\$6.00	Sin especificar	Despachado	VER GENERAR COMPROBANTE

Mostrando 1-1 de 1

Al seleccionar la opción Ver aparecen los detalles de la orden.

Orden

Detalle de orden

[← REGRESAR](#)

HkT1U2 Código de referencia	Despachado Estado	\$6.00 Total	No especificado N° de mesa
---------------------------------------	-----------------------------	------------------------	--------------------------------------

CANTIDAD	PRODUCTO	PRECIO UNITARIO	PRECIO TOTAL
3	 Malteada u.u Bebida fría	\$2.00	\$6.00

TOTAL A PAGAR: **\$6.00**

Mostrando 1-1 de 1

Una vez hecho el consumo y seleccionada la opción Generar comprobante, se presenta el detalle de la orden, junto a cada ítem del pedido hay un botón Añadir al comprobante. Se pueden agregar uno de los ítems o varios, según sea indicado la forma de pago por parte del cliente.

Generar Comprobante

[← REGRESAR](#)

Generación de comprobante para una orden registrada

HkT1U2 Código de referencia	Completado Estado	\$6.00 Total	No especificado N° de mesa
---------------------------------------	-----------------------------	------------------------	--------------------------------------

CANTIDAD	PRODUCTO	PRECIO UNITARIO	PRECIO TOTAL	
1 Cant. Max: 1	 Malteada u.u Bebida fría	\$2.00	\$2.00	AÑADIR AL COMPROBANTE

Mostrando 1-1 de 1

« 1 »

Según la cantidad de items agregados, se actualiza el valor a pagar. Para el detalle de consumidor se presentan por defecto datos de consumidor final y, sobre estos, un buscador para elegir entre alguno de los usuarios registrados en el sistema. Si el cliente no se ha encontrado, se puede actualizar dirección, si se selecciona esta función, aparece una casilla para decidir si se guarda la dirección como nuevo, por defecto está seleccionada. Al final aparecen los tipos de pagos permitidos, en esta versión: Efectivo y Transferencia Bancaria.

<p>Buscar Usuario</p> <input type="text" value="Belen Castro (222222222)"/>	<p>Lista de productos Productos añadidos para el comprobante de pago</p> <table><thead><tr><th>CANTIDAD</th><th>PRODUCTO</th><th>PRECIO UNITARIO</th><th>PRECIO TOTAL</th><th></th></tr></thead><tbody><tr><td>2</td><td> Malteada u.u Bebida fría</td><td>\$2.00</td><td>\$4.00</td><td></td></tr></tbody></table>	CANTIDAD	PRODUCTO	PRECIO UNITARIO	PRECIO TOTAL		2	 Malteada u.u Bebida fría	\$2.00	\$4.00	
CANTIDAD	PRODUCTO	PRECIO UNITARIO	PRECIO TOTAL								
2	 Malteada u.u Bebida fría	\$2.00	\$4.00								

<p>Datos del comprobante Información general del comprobante</p> <p>Belen Castro (222222222) ACTUALIZAR DIRECCIÓN</p> <p>Nombre completo Belen Castro</p> <p>Cédula o RUC 222222222</p> <p>Correo electrónico maria.a.encalada@unl.edu.ec</p> <p>Teléfono 0912345678</p> <p>Dirección Pio Jaramillo & Juan Montalvo</p> <p>Pago Información de pago para el comprobante</p> <p>Monto a pagar (\$) \$4</p> <p>Tipo de pago Efectivo</p> <p>REGISTRAR COMPROBANTE</p>

Una vez completado el proceso de Generar comprobante, la lista de comprobantes es actualizada. Desde la lista de comprobantes se puede ver el comprobante y ver la orden.

Comprobantes de pago

Lista de comprobantes de pago registrados.

CÓDIGO	FECHA DE EMISIÓN	ADQUIRIENTE	PAGO	ACCIONES
GvokFKEtJW	10 de noviembre de 2022 a las 07:31	Belen Castro 222222222	\$4.00 Efectivo	VER VER ORDEN

Mostrando 1-1 de 1 « 1 »

El comprobante sigue el formato de factura, pero no tiene valor tributario. Al pasar el mouse sobre el comprobante, aparece la opción de descargar en formato PDF.

Cafeterias Loja

09687843548984
cafeterias@email.com

Adquiriente:
Belen Castro
222222222
0912345678
maria.a.encalada@unl.edu.ec
Pio Jaramillo & Juan Montalvo

Comprobante

Código: GvokFKEtJW
Fecha de emisión: 10/11/2022
Método de pago: Efectivo

Producto	Cantidad	Precio Unitario	Total
Malteada u.u	2	\$2.00	\$4.00
TOTAL			\$4.00

\$4.00 pagado el 10 de noviembre de 2022

¡Gracias por su compra!



BRENTWOOD LANGUAGE CENTER

Making a difference

Loja, 23 de agosto de 2023

Lic. Laura Dayanna Ramos Montaña

LICENCIADA EN CIENCIAS DE LA EDUCACIÓN MENCIÓN INGLÉS

CERTIFICA:

Que el documento aquí compuesto es traducción fiel y completa al idioma inglés de un documento redactado en español, el mismo que corresponde a un trabajo de titulación denominado como **“Sistema informático para la gestión de pedidos en los locales físicos de cafeterías de la ciudad de Loja aplicando arquitectura distribuidas (microservicios y/o serverless)”** bajo la autoría de Maria Anabel Encalada Córdova con CI: 0707031662, egresada de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja.

Lo certifica en honor a la verdad y autorizo al interesado hacer uso del presente en lo que a sus intereses convenga.

LAURA DAYANNA RAMOS MONTAÑO

DOCENTE DEL INSTITUTO DE INGLÉS BRENTWOOD

CI: 1150489811

1031-2021-2295814