



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE TEGNOLOGÍAS DE LA INFORMACIÓN

Tema:

**SISTEMA WEB APLICANDO ARQUITECTURA SERVERLESS PARA
AUTOMATIZAR LA GESTIÓN FINANCIERA DE LA MICROEMPRESA
“ZOMI”**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Tecnologías de la Información.

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de software

AUTOR: Paul Bernardo Hidalgo Silva

TUTOR: Ing. Leonardo David Torres Valverde.

Ambato - Ecuador

febrero – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA WEB APLICANDO ARQUITECTURA SERVERLESS PARA AUTOMATIZAR LA GESTIÓN FINANCIERA DE LA MICROEMPRESA “ZOMI”, desarrollado bajo la modalidad Proyecto de Investigación por el señor Paul Bernardo Hidalgo Silva, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2024.

Ing. Leonardo David Torres Valverde.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA WEB APLICANDO ARQUITECTURA SERVERLESS PARA AUTOMATIZAR LA GESTIÓN FINANCIERA DE LA MICROEMPRESA “ZOMI” es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2024.



Paul Bernardo Hidalgo Silva

C.C. 0105747323

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2024.



Paul Bernardo Hidalgo Silva

C.C. 0105747323

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Paul Bernardo Hidalgo Silva estudiante de la Carrera de Tecnologías de la Información., de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA WEB APLICANDO ARQUITECTURA SERVERLESS PARA AUTOMATIZAR LA GESTIÓN FINANCIERA DE LA MICROEMPRESA “ZOMI”, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, febrero 2024.

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Oscar Ibarra Torres, Mg.
PROFESOR CALIFICADOR

Ing. Sandra Carrillo Ríos, Mg.
PROFESOR CALIFICADOR

DEDICATORIA

A mi amada familia y a mi novia Paola,

Este logro no habría sido posible sin su amor, apoyo y paciencia inquebrantables a lo largo de este viaje académico. Cada uno de ustedes ha sido mi fuente de inspiración y motivación, guiándome con su sabiduría y alentándome con su amor incondicional.

A mi familia, gracias por creer en mí desde el principio, por su sacrificio y dedicación para brindarme las oportunidades que hoy celebro. Cada uno de ustedes ha sido mi faro en los momentos oscuros, iluminando mi camino con amor y esperanza.

A mis amigos y seres queridos, gracias por su ánimo constante y sus palabras de aliento. Vuestra presencia en mi vida ha sido un regalo invaluable, y cada momento compartido ha sido una bendición.

A todos aquellos que han dejado una huella en mi vida, ya sea a través de palabras de aliento, momentos de alegría o desafíos que me han hecho crecer, les doy las gracias.

Este logro es también suyo.

Con amor y gratitud,

Paul Hidalgo.

AGRADECIMIENTO

Quisiera expresar mi más sincero agradecimiento a todas las personas que han contribuido de alguna manera a la realización de esta tesis:

A mi tutor de tesis, Leonardo Torres, por su orientación experta, su paciencia y su apoyo a lo largo de todo el proceso de investigación.

A mis compañeros/as de clase y amigos/as, por su ánimo constante, y su apoyo emocional durante los momentos más desafiantes

A mi familia, por su amor incondicional, su comprensión y su sacrificio para permitirme perseguir mis sueños académicos. Su apoyo moral y emocional ha sido mi mayor fortaleza y motivación.

A Paola Ramos, por su comprensión, su paciencia y su constante aliento a lo largo de este proceso. Su presencia ha sido mi mayor fuente de inspiración y su apoyo ha sido fundamental para alcanzar este logro.

A todas las personas que de alguna manera contribuyeron a este trabajo, ya sea con sus comentarios, sus sugerencias o su mera presencia en mi vida, les doy las gracias de todo corazón.

Este logro no habría sido posible sin ustedes.

Con gratitud,

Paul Hidalgo.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA.....	i
APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS.....	viii
ÍNDICE DE TABLAS	xiv
ÍNDICE DE FIGURAS.....	xix
ÍNDICE DE ANEXOS.....	xxiv
RESUMEN EJECUTIVO	xxv
ABSTRACT.....	xxvi
CAPÍTULO I. MARCO TEÓRICO.....	27
1.1 Tema de investigación.....	27
1.1.1 Planteamiento del problema.....	27
1.2 Antecedentes investigativos	28
1.3 Fundamentación teórica	30

1.3.1 Arquitectura de sistemas informáticos	30
1.3.2 Cloud computing	30
1.3.3 Tipos de servicios de cloud computing:	30
1.3.4 Arquitectura serverless	31
1.3.5 Sistemas de información	33
1.3.6 Gestión Empresarial	33
1.3.7 Gestión financiera	33
1.3.8 Gestión de la información	34
1.3.9 Control de inventario	34
1.4 Objetivos	35
1.4.1 Objetivo general	35
1.4.2 Objetivos específicos	35
CAPÍTULO II. METODOLOGÍA	36
2.1 Materiales	36
2.2 Métodos	36

2.2.1	Modalidad de la investigación	36
2.2.2	Población y muestra	37
2.2.3	Recolección de información.....	37
2.2.3.1	Entrevista realizada a la gerente.....	37
2.2.3.2	Entrevista realizada al personal de ventas.....	43
2.2.3.3	Entrevista realizada al personal de producción	51
2.2.3.4	Resultados de la matriz de observación de venta.....	55
2.2.3.5	Resultados de la matriz de observación de compra	57
2.2.4	Procesamiento y análisis de datos	59
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....		60
3.1	Análisis de procesos	60
3.1.1	Diagramas de proceso con la gestión financiera actual	60
3.1.2	Diagrama de procesos optimizado	63
3.2	Proveedores de servicio en la nube	64
3.2.1	Modelo de pago y características técnicas de los servicios orientados a la arquitectura serverless en AWS	68
3.3	Servicios serverless	70
3.3.1	AWS lambda	70
3.3.2	DynamoDB	71
3.3.3	Amazon API Gateway	72
3.4	Herramientas para desarrollo web.....	73

3.5 Metodologías de desarrollo WEB	75
3.5.1 Extreme Programming	75
3.5.2 KANBAN.....	76
3.5.3 SCRUM.....	76
3.6 Desarrollo de la propuesta.....	78
3.6.1 Fase I: Planificación.....	79
3.6.2 Requisitos Funcionales del Sistema.....	79
3.6.3 Roles Asignados en el proyecto	80
3.6.4 Arquitectura de la Aplicación	80
3.6.5 Requerimientos de Hardware.....	81
3.6.6 Requerimientos de Software	81
3.6.7 Historias de Usuario.....	81
3.6.8 Estimación de Historias de Usuario	88
3.6.9 Plan de Entrega	88
3.7 Fase II: Diseño	89
3.7.1 Plan de iteraciones.....	89
3.7.2 Tarjetas CRC.....	96
3.7.3 Diseño de la base de datos	100
3.7.4 Diseño de Iteraciones	103
3.8 Fase III: Codificación.....	111

3.8.1	Codificación de archivo serverless.yml	112
3.8.2	Codificación de las funciones lambdas	113
3.8.3	Despliegue del archivo serverless.yml	115
3.8.4	Codificación del front-end	115
3.8.5	Codificación de los servicios con los endpoints	116
3.8.6	Codificación de los componentes	119
3.8.7	Codificación de los Layouts.....	122
3.8.8	Codificación de las páginas.....	123
3.8.9	Codificación de las funciones	129
3.8.9.1	Funciones de autenticación	130
3.8.9.2	Funciones para módulo de compras	133
3.8.9.3	Funciones para facturas.....	136
3.8.9.4	Funciones para informes	136
3.8.9.5	Funciones para el inventario	137
3.8.9.6	Funciones para ventas	139
3.8.10	Codificación de las integraciones.....	141
3.9	Fase IV: Pruebas	141
3.9.1	Pruebas de aceptación	141
3.9.2	Pruebas de usabilidad.....	147
3.10	Fase V: Implantación	149

3.10.1 Proceso de Configuración	149
3.10.2 Capacitaciones.....	153
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES.....	154
4.1 Conclusiones	154
4.2 Recomendaciones.....	155
REFERENCIAS BIBLIOGRÁFICAS.....	156
ANEXOS	159

ÍNDICE DE TABLAS

Tabla 1 Población.....	37
Tabla 2. Matriz de observación para el proceso de venta	55
Tabla 3. Matriz de observación para el proceso de compra.....	57
Tabla 4. Comparación entre proveedores de servicio en la nube.....	65
Tabla 5. Modelo de pago y características técnicas de los servicios	68
Tabla 6. Características de AWS lambda.....	70
Tabla 7. Características de DynamoDB	71
Tabla 8. Características Amazon API Gateway	72
Tabla 9. Comparación de frameworks front-end	74
Tabla 10. Comparación de metodologías.....	77
Tabla 11. Definición de roles	80
Tabla 12. Modelo de tabla para historia de usuario	82
Tabla 13. Historia de usuario registro de usuarios.....	83
Tabla 14. Historia de usuario inicio de sesión	83
Tabla 15. Historia de usuario recuperación de contraseña.....	83
Tabla 16. Historia de usuario menú de opciones	84
Tabla 17. Historia de usuario registro de productos	84
Tabla 18. Historia de usuario información del producto.....	84
Tabla 19. Historia de usuario lista de productos del inventario.....	85
Tabla 20. Historia de usuario registro de compras.....	85

Tabla 21. Historia de usuario información de la compra	85
Tabla 22. Historia de usuario historial de compras	86
Tabla 23. Historia de usuario registro de ventas	86
Tabla 24. Historia de usuario detalle de la venta	86
Tabla 25. Historia de usuario historial de ventas	87
Tabla 26. Historia de usuario informes	87
Tabla 27. Historia de usuario finalizar sesión	87
Tabla 28. Estimación de historias de usuario	88
Tabla 29. Plan de entrega	89
Tabla 30. Historia de usuario Iteración 1	90
Tabla 31. Actividades de la historia de usuario 01	90
Tabla 32. Actividades de la historia de usuario 02	90
Tabla 33. Actividades de la historia de usuario 03	91
Tabla 34. Actividades de la historia de usuario 04	91
Tabla 35. Historia de usuario Iteración 2	91
Tabla 36. Actividades de la historia de usuario 05	91
Tabla 37. Actividades de la historia de usuario 06	92
Tabla 38. Actividades de la historia de usuario 07	92
Tabla 39. Historias de usuario Iteración 3	92
Tabla 40. Actividades de la historia de usuario 08	93
Tabla 41. Actividades de la historia de usuario 09	93

Tabla 42. Actividades de la historia de usuario 10	93
Tabla 43. Historia de usuario Iteración 4	94
Tabla 44. Actividades de la historia de usuario 11	94
Tabla 45. Actividades de la historia de usuario 12	94
Tabla 46. Actividades de la historia de usuario 13	95
Tabla 47. Historia de usuario Iteración 4	95
Tabla 48. Actividades de la historia de usuario 14	95
Tabla 49. Actividades de la historia de usuario 15	95
Tabla 50. Tarjeta CRC – Inicio de sesión	96
Tabla 51. Tarjeta CRC – Registro de usuarios.....	96
Tabla 52. Tarjeta CRC – Recuperación de contraseña	96
Tabla 53. Tarjeta CRC – Menú de Opción	97
Tabla 54. Tarjeta CRC – Registro de productos	97
Tabla 55. Tarjeta CRC – Información del producto	97
Tabla 56. Tarjeta CRC – Lista de productos del inventario.....	97
Tabla 57. Tarjeta CRC – Registro de compras	98
Tabla 58. Tarjeta CRC – Información de la compra.....	98
Tabla 59. Tarjeta CRC – Historial de compras	98
Tabla 60. Tarjeta CRC – Registro de ventas.....	98
Tabla 61. Tarjeta CRC – Detalle de la venta	99
Tabla 62. Tarjeta CRC – Historial de ventas	99

Tabla 63. Tarjeta CRC – Informes	99
Tabla 64. Tarjeta CRC – Cerrar sesión	99
Tabla 65. Prueba de aceptación número 1.	141
Tabla 66. Prueba de aceptación numero 2	142
Tabla 67. Prueba de aceptación numero 3	142
Tabla 68. Prueba de aceptación numero 4	142
Tabla 69. Prueba de aceptación numero 5	143
Tabla 70. Prueba de aceptación numero 6	143
Tabla 71. Prueba de aceptación numero 7	143
Tabla 72. Prueba de aceptación numero 8	144
Tabla 73. Prueba de aceptación numero 9	144
Tabla 74. Prueba de aceptación numero 10	145
Tabla 75. Prueba de aceptación numero 11	145
Tabla 76. Prueba de aceptación numero 12	146
Tabla 77. Prueba de aceptación numero 3	146
Tabla 78. Prueba de aceptación numero 14	147
Tabla 79. Prueba de aceptación numero 15	147
Tabla 80. Resultados prueba usabilidad usuario administrador.....	148
Tabla 81. Resultados prueba usabilidad usuario ventas.....	148
Tabla 82. Capacitaciones.	153
Tabla 83. Matriz de observación de ventas.....	168

Tabla 84. Matriz de observación de compras..... 169

ÍNDICE DE FIGURAS

Figura 1. Proceso actual de compra de mercadería.....	60
Figura 2. Proceso actual de venta de productos	61
Figura 3 Proceso de registro y almacenamiento de productos.....	62
Figura 4 Flujo de proceso para compra e ingreso de productos.....	63
Figura 5. Flujo de proceso para la venta de producto	63
Figura 6. Flujo de proceso XP [24].....	75
Figura 7. Flujo de trabajo KANBAN. [25]	76
Figura 8. Flujo de proceso SCRUM [24]	77
Figura 9.Arquitectura del proyecto	80
Figura 11. Estructura de DynamoDB.....	100
Figura 12. Tablas de DynamoDB para el proyecto.....	100
Figura 13. Inicio de sesión.	104
Figura 14 Registro de usuario.	104
Figura 15. Recuperar contraseña.....	105
Figura 16. Menú principal.....	105
Figura 17. Registro de productos.	106
Figura 18. Información del producto.	106
Figura 19. Lista de productos del inventario.....	107
Figura 20. Registro de compras.	107
Figura 21. Información de la compra.....	108

Figura 22. Historial de compras.	109
Figura 23. Registro de ventas.	109
Figura 24. Detalle de venta.	110
Figura 25. Historial de las ventas.	110
Figura 26. Informes.	111
Figura 27. Cerrar sesión.	111
Figura 28. Configuración de AWS CLI.	112
Figura 29. Codificación del archivo serverless.yml.	112
Figura 30. Función lambda para insertar datos.	113
Figura 31. Función lambda para actualizar datos.	113
Figura 32. Función lambda para obtener los datos.	114
Figura 33. Función lambda para eliminar datos.	114
Figura 34. Despliegue del archivo serverless.yml.	115
Figura 35. Nuevo proyecto con Astro.	116
Figura 36. Carpetas del proyecto.	116
Figura 37. Servicio de inventario.	117
Figura 38. Servicio de usuarios.	117
Figura 39. Servicio de compras.	118
Figura 40. Servicio de ventas.	118
Figura 41. Servicio de fecha.	119
Figura 42. Componente de ventas.	119

Figura 43. Componente de compras.....	120
Figura 44. Componente de inventario.....	120
Figura 45. Componente facturas.....	121
Figura 46. Componente de registro usuario.....	121
Figura 47. Componente header.....	122
Figura 48. Layout de la página principal.....	122
Figura 49. Layout para el archivo index, astro.....	123
Figura 50. Páginas del proyecto.....	123
Figura 51. Página de registro de usuarios.....	124
Figura 52. Página de la información de la compra.....	124
Figura 53. Página de la información de la factura.....	125
Figura 54. Página de la información del producto.....	125
Figura 55. página de compras.....	126
Figura 56. Página de facturas.....	126
Figura 57. Página principal.....	127
Figura 58. Página de inicio de sesión.....	127
Figura 59. Página de informes.....	128
Figura 60. Página de inventario.....	128
Figura 61. Página de ventas.....	129
Figura 62. Funciones del proyecto.....	129
Figura 63. Función configuración AWS Cognito.....	130

Figura 64. Función recuperar contraseña.	130
Figura 65. Función ingresar al sistema	131
Figura 66. Función para salir del sistema.....	131
Figura 67. Función para registrar usuario	132
Figura 68. Función verificar si estas logueado.....	132
Figura 69. Función actualizar compra.....	133
Figura 70. Función auto completado proveedor.	133
Figura 71. función búsqueda compra	134
Figura 72. Función llenar inventario por compras	134
Figura 73. Función nueva compra.....	135
Figura 74. función lista de proveedores	135
Figura 75. Función llenar datos factura.	136
Figura 76. función informes compras.	136
Figura 77. Función informe ventas	137
Figura 78. Función actualizar producto inventario	137
Figura 79. Función búsqueda inventario.....	138
Figura 80. Función nuevo producto inventario.	138
Figura 81. Función llenado categorías inventario	139
Figura 82. Función auto completado cliente	139
Figura 83. Función guardar vena	140
Figura 84. Función llenar categorías	140

Figura 85. Integraciones del proyecto.....	141
Figura 86. Creación de la cuenta en Cloudflare.com.....	149
Figura 87. Menú principal de Cloudflare.....	150
Figura 88. Nueva página con Git.....	150
Figura 89. Selección de repositorio Git.....	150
Figura 90. Selección de framework y despliegue.....	151
Figura 91. Fases del despliegue.....	152
Figura 92. Información del despliegue finalizado.....	152
Figura 93. Aplicación implementada.....	153

ÍNDICE DE ANEXOS

Anexo A. Entrevistas y Matriz de observación.....	159
Anexo B. Manual de usuario.....	170

RESUMEN EJECUTIVO

En la actualidad se ha incrementado el uso de las tecnologías dentro del área financiera lo que ha permitido automatizar procesos y gestionar las operaciones financieras de una organización. La mayoría de las empresas y negocios cuentan con aplicaciones o dispositivos tecnológicos que ayudan al desempeño de las actividades de la organización con el fin de optimizar tiempos y recursos de ejecución.

El objetivo de este proyecto es la implementación de un sistema web usando la arquitectura serverless en la microempresa ZOMI con el fin de automatizar la gestión financiera y almacenar los datos de cada transacción de forma efectiva dentro de las áreas de ventas, compras e inventario, y así obtener una información más precisa y actualizada que se puede obtener en cualquier momento. Además, se evita el uso de hojas de papel o cuadernos para llenar esta información ya que el sistema web es muy fácil de manejar.

El sistema web se desarrolló utilizando la metodología XP, donde las entrevistas realizadas al personal y la interacción con los mismos ayudaron a facilitar el desarrollo de las actividades necesarias para el sistema web, esto además permitió organizar de forma adecuada los tiempos de desarrollo.

Para el desarrollo de la arquitectura serverless se trabajó con el proveedor de la nube AWS, que tiene como referencia la parte de backend como, por ejemplo: DynamoDB, Funciones Lambdas y API Gateway que son los principales recursos que se utilizó para el desarrollo del sistema web.

Palabras clave: Serverless, DynamoDB, Lambdas, XP, AWS, Tailwind

ABSTRACT

In recent times, there has been an increased use of technologies within the financial sector, enabling the automation of processes and the management of financial operations within an organization. The majority of companies and businesses now utilize applications or technological devices to enhance the performance of organizational activities, aiming to optimize time and resource execution.

The objective of this project is to implement a web system using serverless architecture in the microenterprise ZOMI, with the aim of automating financial management and effectively storing transaction data within the areas of sales, purchases, and inventory. This approach allows for more precise and up-to-date information that can be accessed at any given moment. Furthermore, it eliminates the need for paper sheets or notebooks to record this information, as the web system is easy to manage.

The web application was developed using the XP methodology, where interviews with personnel and interaction with them facilitated the necessary activities for the application. This approach also helped in organizing development timelines effectively.

For the development of the serverless architecture, we collaborated with the AWS cloud provider, utilizing backend components such as DynamoDB, Lambda Functions, and API Gateway, which are the primary resources used in the application development.

Keywords: Serverless, DynamoDB, Lambdas, XP, AWS, Tailwind.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

Sistema web aplicando arquitectura serverless para automatizar la gestión financiera de la microempresa “ZOMI”.

1.1.1 Planteamiento del problema

En Latinoamérica, desde la aparición del internet, se ha incrementado la innovación empresarial, y esto conlleva a que grandes, medianas y pequeñas empresas utilicen herramientas tecnológicas, con fines de optimización y creación de nuevos servicios o productos que influye de manera positiva a la empresa, siendo esto un recurso importante para su supervivencia en el mercado.

Además, el uso de las tecnologías de la información se considera como una ayuda a las PYMES, ofreciendo nuevas oportunidades que les permite ampliar y llegar a más consumidores.[1]

Sin embargo, a pesar de la gran importancia sobre el uso de las tecnologías de la información como herramienta de desarrollo empresarial, en Ecuador aún se evidencian bajos niveles de innovación tecnológica [2], principalmente en pequeñas y medianas empresas que influye directamente a la competitividad de estas.

Por otra parte, en la provincia de Tungurahua, cantón Ambato existe una mayor utilización de herramientas tecnológicas para apoyar y mejorar las operaciones de marketing y venta. Luis Mantilla mencionó que *"un buen número de pymes afirmaron emplear la tecnología, esencialmente computadoras y software especializado, para las tareas relacionadas con la contabilidad, finanzas y tareas administrativas"* [2], dando como resultado la importancia del uso de las tecnológicas para mejorar el desarrollo empresarial.

“ZOMI” es una microempresa de carácter familiar situado en la ciudad de Ambato en la provincia de Tungurahua, ubicado en el Centro Comercial Artesanal local 79. Esta

empresa ha permanecido en el mercado por más de 5 años, por lo que tiene una amplia variedad de clientes.

“ZOMI” no cuenta con un software de gestión financiera; por esta razón, tiene un manejo deficiente de los productos, inventarios, ingresos y gastos dentro de la microempresa, dando como resultado mala gestión financiera.

1.2 Antecedentes investigativos

Basado en la recopilación y análisis de proyectos de investigación de distintas universidades, se encontró los siguientes antecedentes:

E. Capuz [3] en su tema de investigación de la Universidad Técnica de Ambato denominando “GESTIÓN ADMINISTRATIVA FINANCIERA Y SU INCIDENCIA EN LA RENTABILIDAD DE FERRETERÍA AYALCA EN EL PERIODO 2019-2021” el investigador concluye que:

- La gestión es una herramienta que desarrolla estrategias para alcanzar los objetivos empresariales, y es de gran importancia porque permite tener el control fijo de la organización.
- Es fundamental implementarla desde sus inicios para generar métodos que ayudarán a la toma de decisiones eficientes y oportunas durante el desarrollo de las actividades empresariales.

La investigación desarrollada por D. Boris [4] en la universidad estatal del sur de Manabí año 2022 denominado “DESARROLLO E IMPLEMENTACIÓN DE APLICACIÓN WEB PARA GESTIÓN DE INVENTARIOS Y FACTURACIÓN ELECTRÓNICA EN LA EMPRESA SUMINISTROS DELGADO.” El investigador menciona que:

- Las aplicaciones web desempeñan un papel crucial en la optimización de los procesos administrativos de las organizaciones, contribuyendo significativamente a mejorar la funcionalidad y eficiencia de sus operaciones.

- Concluye además que los sistemas no solo ofrecen soluciones económicas, sino que también abordan la problemática asociada al manejo de la información de los datos, proporcionando un marco sólido para la toma de decisiones y el desarrollo organizacional.

El proyecto de investigación realizado por E. Romero [5] de la Universidad de Cooperativa de Colombia en el año 2019 denominada “DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE INVENTARIOS PARA EL ALMACÉN DE PINTURAS Y FERRETERÍA FERRECOLOR”. En su investigación, el autor concluye que:

- Todo proceso de desarrollo de software requiere una serie de pasos para dar solución a requerimientos del cliente, para ello se usan las metodologías de desarrollo de software.
- Para toda empresa que se dedica a la comercialización de productos o bienes, es fundamental conocer el estado de su inventario, y entre más precisas, sea esta información va a ser mejor la gestión financiera de este.
- El uso de soluciones de software fue un factor fundamental en la mejora y/o automatización de procesos en una empresa.

En el proyecto de investigación de D. Pacios [6] de la Universidad Complutense Madrid año 2022 denominado “SERVERLESS ARCHITECTURE FOR DATA PROCESSING AND DETECTING ANOMALIES IN MARSISINSTRUMENT,” el autor menciona que:

- La arquitectura serverless como ventaja para las empresas es que sólo pagan por el tiempo de los recursos consumidos.
- El uso de serverless es una nueva opción que les permite a las empresas mejorar su economía en el contexto de que la empresa no paga costos fijos de los recursos si no solo lo tiempo que consume los servicios.

En el proyecto de investigación de M. Maisanche y H. Chicaiza [6] , en su tema de investigación, “La tecnología como factor determinante para el mejoramiento de la calidad en empresas de servicios de la provincia de Tungurahua”, menciona que:

- En la provincia de Tungurahua, el uso de las herramientas tecnológicas ha permitido que varias empresas logren incrementar sus ingresos y mejorar la calidad de sus servicios, dando como ejemplo las redes sociales, eso ha permitido llegar a un mayor número de clientes.

1.3 Fundamentación teórica

1.3.1 Arquitectura de sistemas informáticos

la rama de las ciencias informáticas conocido como arquitectura de sistemas tiene como objeto la definición de un modelo teórico desarrollado a partir del estudio, análisis y diseño de la organización lógica, el comportamiento y los elementos funcionales de un ordenador, con el fin de ser empleado en el desarrollo tecnológico de los componentes físicos que lo integran. [7]

1.3.2 Cloud computing

El NIST (National Institute of Standards and Technology) defines cloud computing es “un modelo tecnológico que permite el acceso ubicuo, adaptado y bajo demanda en red a un conjunto compartido de recursos de computación configurables y compartidos (por ejemplo: redes, servidores, equipos de almacenamiento, aplicaciones y servicios), que pueden ser rápidamente aprovisionados y liberados con un esfuerzo de gestión reducido o interacción mínima con el proveedor del servicio”. [8]

1.3.3 Tipos de servicios de cloud computing:

- *IaaS (Infrastructure as a service)*

Este servicio provee de servidores, almacenamiento, redes de comunicaciones entre otros muchos servicios que se gestionan fácilmente al gusto del cliente a través de una web o vía API. [9]

- ***PaaS (Platform as a service)***

El servicio provee de una plataforma para el desarrollo, despliegue, ejecución y administración del software; de esta forma, se ahorran costos a la hora de construir y mantener servicios como los servicios de IaaS. [9]

- ***SaaS (Software as a service)***

Este servicio es el más común; los usuarios finales no se tienen que preocupar por su desarrollo ni del despliegue; simplemente el usuario final hace uso del servicio y el proveedor se encarga del mantenimiento, operación y soporte del producto.[9]

- ***BaaS(Backend as a service)***

Los servicios backend comprenden aplicaciones de terceros que sirven para administrar la lógica y el estado del servidor. Estos sistemas son más complejos y usan entornos con numerosas bases de datos y servicios de autenticación que dependen de la nube.[10]

- ***FaaS(Function as a service)***

Proporciona una plataforma que permite desarrollar, ejecutar y administrar una funcionalidad sin necesidad de crear ni mantener una infraestructura. Algunos ejemplos de servicios FaaS son: AWS Lambda, Google Functions o Azure Functions, estos servicios permiten ejecutar funciones sin administrar ninguna infraestructura.[9]

1.3.4 Arquitectura serverless

Las arquitecturas serverless son un modelo arquitectónico en el que los servidores (físicos o en la nube) dejan de existir para el desarrollador y el código corre en “ambientes de ejecución” sin estado (stateless) y efímeros. Habitualmente los proveedores cloud son: Amazon con AWS Lambda, Azure con Azure Functions, Google con Cloud Functions, IBM, Alibaba etc.[9]

La principal ventaja de la arquitectura reside en que el proveedor soporta ciertos lenguajes, de forma que el programador inserte su código en una función pública (Función Lambda) y el proveedor sea el que se encargue de administrar toda la infraestructura. Esto permite el escalado horizontal y vertical, además, una de las características más destacadas es que solo se paga por el uso que se haga de la misma, a diferencia de tener un servidor propio dando servicio constantemente (que sería el caso de uso normal) por lo que se ahorra todo el tiempo ocioso de la máquina, ya que se cobra por ejecución. Normalmente, se busca que cada una de estas funciones tenga una funcionalidad muy concreta. Además, las arquitecturas serverless se componen de servicios FaaS y BaaS. Los servicios FaaS son serverless porque el desarrollador no se tiene que preocupar de ninguna infraestructura, el componente FaaS es la funcionalidad principal, la ejecución de la lógica programada sin la preocupación de la infraestructura básica como el entorno de ejecución o el almacenamiento. El componente BaaS se encuentra en la infraestructura que se puede manejar a través de una función con servicios externos u otros servicios proporcionados por el propio proveedor dado el alto acoplamiento, tales como la autenticación, almacenamiento en bases de datos relacionales y no relacionales... etc, y todo ello realizado desde el código[11]

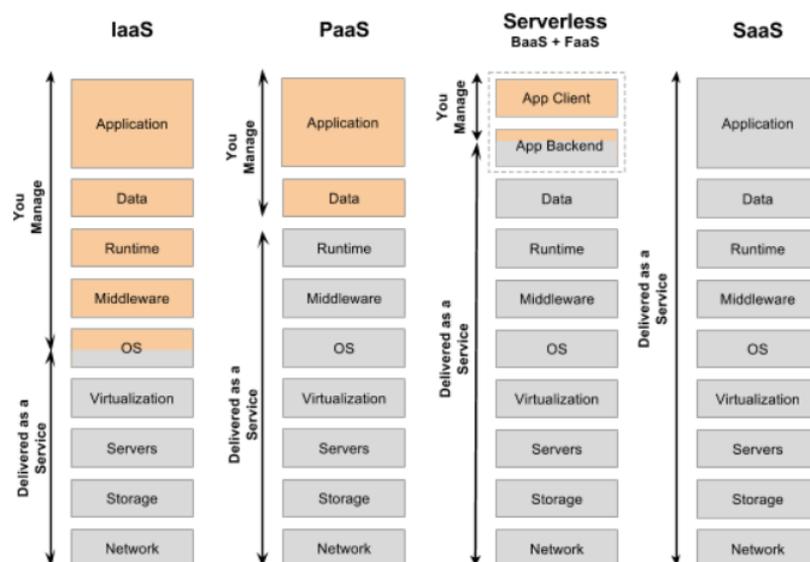


Figura 1 Grado de automatización usando serverless [11]

En resumen, la arquitectura serverless es versátil y se puede aplicar a una amplia gama de casos de uso, desde aplicaciones web y móviles hasta procesamiento de eventos y datos en tiempo real. Su escalabilidad automática y su modelo de precios basado en el consumo lo hacen especialmente atractivo para muchas empresas y desarrolladores.

1.3.5 Sistemas de información

Los sistemas de información son conjuntos de componentes interrelacionados que trabajan juntos para recopilar, procesar, almacenar, y distribuir información para apoyar las operaciones, la gestión y la toma de decisiones en una organización. Estos sistemas incluyen hardware, software, bases de datos, telecomunicaciones, y personas, que se integran en un sistema de información organizacional más grande.[12]

1.3.6 Gestión Empresarial

La gestión empresarial es un componente esencial en el éxito de cualquier organización. Se trata de un enfoque interdisciplinario que abarca una amplia gama de temas, incluyendo estrategias, finanzas, marketing, recursos humanos, operaciones, y tecnologías de la información. Esta disciplina se centra en la toma de decisiones y la resolución de problemas en una empresa, con el objetivo de mejorar la eficiencia y productividad, aumentar las ganancias, y mantener la satisfacción del cliente.[13]

1.3.7 Gestión financiera

La gestión financiera se trata del proceso de planificación, organización, control, y monitoreo de los recursos financieros esto con el objetivo de maximizar la rentabilidad y minimizar los riesgos. Entre las actividades incluidas en la gestión financiera, destacan la identificación de oportunidades de inversión, la gestión de tesorería, la toma de decisiones de inversión y la gestión de la deuda. Todas estas actividades son esenciales para asegurar una gestión eficiente y efectiva de los recursos financieros. No solo se enfoca en la obtención de una rentabilidad máxima, sino también en la gestión responsable y ética de los recursos financieros.[14]

1.3.8 Gestión de la información

La gestión de la información se refiere a los procesos sistemáticos de la recopilación, organización, almacenamiento, mantenimiento, uso y disposición de información para lograr objetivos efectivos y eficientes en una organización. Incluyendo la identificación de las necesidades de información, la recopilación de datos relevantes, su clasificación y análisis, y la distribución de la información a los interesados. La gestión de la información también aborda la seguridad y la privacidad de la información, así como la necesidad de mantener la información actualizada y accesible.[15]

1.3.9 Control de inventario

Control de inventario es un proceso fundamental en la gestión empresarial que se enfoca en mantener un equilibrio óptimo entre la cantidad de productos disponibles y la demanda de los clientes. Esto implica la identificación, registro y monitoreo constante de los productos, materiales y suministros en existencia.

Control de inventario es esencial para garantizar la eficiencia y rentabilidad de una empresa. Por esta razón, es indispensable utilizar técnicas y métodos adecuados para llevar a cabo un control de inventario efectivo. Es por esta razón que la tecnología ha tenido un impacto significativo en este proceso, facilitando la recopilación y análisis de datos en tiempo real y permitiendo un monitoreo más preciso y eficiente.

En resumen, el control de inventario es una parte fundamental en la gestión financiera de las empresas además es crucial para el éxito de cualquier empresa que maneje una gran cantidad de productos o suministros.[16]

1.4 Objetivos

1.4.1 Objetivo general

Implantar un sistema Web aplicando arquitectura serverless para automatizar la gestión financiera de la microempresa "ZOMI".

1.4.2 Objetivos específicos

- Analizar los procesos de control de compra, venta e inventario de productos de la microempresa "ZOMI".
- Investigar el funcionamiento de arquitectura serverless
- Desarrollar el sistema Web utilizando arquitectura serverless para automatizar la gestión financiera de la microempresa "ZOMI".

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Para realizar el proceso de la recolección de la información, se realizó dos matrices de observación, una enfocada al proceso de ventas y la segunda enfocada en el proceso de adquisición de productos. Además, se realizó cuatro entrevistas estructuradas, una destinada a la gerente de la microempresa, dos destinadas al personal de ventas y una al personal de producción. La información recopilada permitirá recolectar los requerimientos para el desarrollo del sistema web.

2.2 Métodos

2.2.1 Modalidad de la investigación

Para esta investigación se utilizará las siguientes modalidades:

- ***Investigación de campo***

Para el desarrollo de esta investigación se utilizará esta modalidad ya que es necesario estar en constante convivencia con el objeto de estudio, en este caso permitirá entrar en contacto con el problema mediante la recolección de la información que será obtenida en la microempresa “ZOMI” así mismo se contará con la colaboración de su propietaria, la de proporcionar y ayudar con la información de la microempresa ya que será de vital importancia en la investigación.

- ***Investigación Bibliográfica – Documental***

Para esta investigación, se utilizarán algunas páginas web como revistas digitales y buscadores, con el propósito de seleccionar información importante para este trabajo. estas investigaciones estarán basadas en distintos lugares y criterios.

- ***Investigación Aplicada***

Para esta investigación, se utilizarán algunas páginas web como revistas digitales, repositorios de universidades y buscadores, con el propósito de seleccionar información importante para este trabajo. Estas investigaciones estarán basadas en distintos lugares y criterios.

2.2.2 Población y muestra

El presente proyecto de investigación trabajara con toda la población de la microempresa “ZOMI” dedicada a la producción de artesanías.

Tabla 1 Población

POBLACIÓN	NUMERO	PORCENTAJE
Gerente	1	25%
Personal de producción	1	25%
Personal de ventas	2	50%
TOTAL	4	100%

2.2.3 Recolección de información

Los resultados obtenidos de las entrevistas aplicadas y observaciones al personal seleccionado de la microempresa “ZOMI” son los siguientes:

2.2.3.1 Entrevista realizada a la gerente

ENTREVISTA DIRIGIDA AL GERENTE PROPIETARIO DE LA MICROEMPRESA “ZOMI”

Sistema de gestión financiera

Fecha: 17/10/2023

Entrevistador: Paul Hidalgo

Entrevistado: Michelle Hidalgo

Cargo: Gerente

Objetivo:

Analizar los procesos de control de compra, venta e inventario de productos la Microempresa “ZOMI”.

Introducción:

Entrevistador: Buen día, agradecemos tu participación en esta entrevista que tiene como objetivo obtener información valiosa para la implementación exitosa de nuestro nuevo sistema web de gestión financiera en "ZOMI". Queremos escuchar tus opiniones y experiencias para garantizar que el sistema se adapte de la mejor manera a nuestras operaciones diarias. ¿Estás listo para comenzar?

Preguntas:

Pregunta 1: ¿Se lleva un registro detallado de las compras realizadas dentro de la microempresa "ZOMI"?

Respuesta:

Actualmente, no llevamos un registro detallado de las compras realizadas en "ZOMI". La mayoría de las compras se realizan en efectivo o con pagos directos a los proveedores, lo que dificulta mantener un seguimiento preciso de las transacciones. En su mayoría, las compras se basan en la necesidad inmediata de materias primas o suministros y no se registran sistemáticamente.

Observación:

Parece que actualmente la empresa carece de un sistema de registro de compras. Esto puede tener implicaciones en la gestión de gastos y la toma de decisiones financieras.

Pregunta 2: ¿Cuáles son las principales dificultades que enfrenta la microempresa "ZOMI" en relación con la gestión de inventarios?

Respuesta:

Las principales dificultades que enfrentamos en relación con la gestión de inventarios en "ZOMI" son las siguientes:

- No tenemos un sistema que nos permita llevar un seguimiento preciso de la cantidad de productos que tenemos en stock en tiempo real. Esto dificulta saber cuándo un producto está agotado o cuántas unidades disponibles tenemos.
- La gestión de inventario se realiza de manera manual, lo que puede dar lugar a errores humanos en el registro de cantidades y productos. Esto afecta la precisión de nuestros datos de inventario.
- Desafíos en la producción: La falta de un sistema de gestión de inventario también se traduce en desafíos en la producción. A menudo, producimos productos sin un registro claro de la demanda, lo que lleva a la producción excesiva de algunos productos y a la escasez de otros.

Observación:

La falta de un sistema de gestión de inventarios y la gestión manual de productos han llevado a desafíos en la eficiencia operativa y la toma de decisiones en "ZOMI".

Pregunta 3: ¿cuáles son los principales desafíos que enfrentas actualmente en proceso de ventas de productos?

Respuesta:

Los principales desafíos que enfrentamos en el proceso de ventas de productos en "ZOMI" son los siguientes:

- No tenemos un sistema que nos permita llevar un seguimiento detallado de las ventas. Esto dificulta saber cuántos productos se han vendido y cuántos están disponibles en el inventario en tiempo real.
- Las ventas se registran manualmente, lo que aumenta la probabilidad de errores en el registro y la falta de precisión en los datos de ventas.

Observación:

Entiendo que la falta de un sistema de ventas y el registro manual han llevado a desafíos en la gestión de ventas, eficiencia operativa en "ZOMI".

Pregunta 4: ¿Podrías describir el proceso que sigue la microempresa para evaluar y analizar las ganancias generadas durante un mes?

Respuesta:

Nuestro proceso actual para evaluar y analizar las ganancias generadas durante un mes en "ZOMI" es principalmente manual, solo se evalúa las ganancias según lo que se ha registrado, sin embargo, esta evaluación no se lleva a cabo de manera precisa y detallada sin saber con exactitud las verdaderas ganancias o pérdidas en un mes

Observación:

Su proceso actual de evaluación de ganancias se basa en un método manual de recopilación de datos, pero no es el adecuado por la falta de precisión en los registros.

Pregunta 5: ¿Considera usted que al aplicar un sistema web dentro de la microempresa "ZOMI" mejoraría la gestión financiera? Si es así, ¿podrías mencionar algunos aspectos específicos?

Respuesta:

Sí, considero que la implementación de un sistema web en la microempresa "ZOMI" podría mejorar significativamente nuestra gestión financiera. Algunos aspectos específicos que podrían beneficiarse incluyen:

- Un sistema web permitiría un registro automatizado de todas las transacciones financieras, lo que reduciría la carga de trabajo manual y minimizaría la posibilidad de errores.

- Tener acceso a datos financieros en tiempo real nos proporcionaría una visión actualizada de nuestras ganancias y pérdidas, lo que facilitaría la toma de decisiones.
- Un sistema web podría generar informes financieros que nos brinden información detallada sobre ingresos, gastos y ganancias. Esto facilitaría la evaluación de rendimiento y la planificación financiera.
- Si el sistema web se integra con la gestión de inventario, podríamos llevar un seguimiento preciso de las existencias de productos, lo que evitaría problemas de exceso o escasez de inventario.

Observación:

Estos son puntos sólidos que destacan cómo un sistema web podría beneficiar la gestión financiera en varios aspectos.

Pregunta 6: ¿Qué características y funcionalidades específicas les gustaría que tuviera un sistema web de gestión financiera dentro microempresa "ZOMI"?

Por ejemplo:

- Que permita registrar fácilmente todos los ingresos y gastos de la empresa
- Tenga la capacidad de generar informes financieros precisos y detallados
- Que permita llevar un seguimiento del inventario, con alertas de reabastecimiento

Respuesta:

Las características y funcionalidades específicas que nos gustaría que tuviera un sistema web de gestión financiera en "ZOMI" son las siguientes:

- Queremos que el sistema permita el registro automatizado de ventas, gastos y otros movimientos financieros, eliminando la necesidad de un registro manual propenso a errores.
- Deseamos que el sistema esté vinculado a la gestión de inventario para llevar un control preciso de las existencias de productos y evitar problemas de exceso o escasez.
- Nos gustaría la capacidad de generar informes financieros que incluyan detalles sobre ingresos, gastos, márgenes de beneficio y otros indicadores clave.

- Es importante que el sistema proporcione acceso en tiempo real a los datos financieros

Observación:

Estas características y funcionalidades son esenciales para mejorar la gestión financiera en "ZOMI".

Pregunta 7: ¿Cuáles crees que son los posibles desafíos que podrían surgir durante la implementación de un sistema web para la gestión financiera en la microempresa 'ZOMI'?

Respuesta:

En mi opinión, algunos de los posibles desafíos que podrían surgir durante la implementación de un sistema web para la gestión financiera en la microempresa 'ZOMI' incluyen:

- La adaptación al nuevo sistema requerirá capacitación para todo el personal. Algunos empleados pueden no estar familiarizados con sistemas web, lo que podría llevar tiempo.
- Quiero asegurarnos de que el sistema se integre eficazmente con nuestros procesos comerciales actuales podría ser un desafío. Debe ser lo más compatible posible con nuestras operaciones existentes.
- Costos iniciales: La implementación de un sistema web implica costos iniciales, incluyendo la contratación de servicios. Debemos asegurarnos de que estos costos sean manejables y proporcionen un retorno de la inversión positivo a largo plazo.

Observación:

Estos son desafíos importantes que considerar durante la implementación de un sistema web. La capacitación, la integración y la seguridad de datos son factores críticos para una transición exitosa.

Pregunta 8: ¿A trabajado alguna vez con un sistema WEB? Si es así, ¿puede mencionar uno de ellos?

Respuesta:

No, en la microempresa 'ZOMI', no hemos trabajado con sistemas web anteriormente. Actualmente, nuestras operaciones financieras se basan en métodos manuales y registros en papel.

Observación:

Entiendo que no tienen experiencia previa con sistemas web. Esto significa que la transición a un sistema web podría requerir un período de adaptación y capacitación para el personal.

Pregunta 9: Si se implantara un sistema web para la gestión financiera en la microempresa 'ZOMI', ¿cómo crees que este cambio afectaría tu día a día y tu forma de realizar las tareas relacionadas con las ventas y la gestión de productos?

Respuesta:

Creo que la implementación de un sistema web para la gestión financiera tendría un impacto significativo en mi día a día y en la forma en que realizo las tareas relacionadas con las ventas y la gestión de productos. Algunos de los cambios que anticiparía incluyen:

- El sistema web automatizaría tareas que actualmente realizamos manualmente, como el registro de ventas y la generación de informes. Esto liberaría tiempo que puedo dedicar a otras actividades.
- Tener acceso en tiempo real a datos financieros me permitiría tomar decisiones más informadas de manera más rápida.
- La automatización reduciría la probabilidad de errores en los registros, lo que mejoraría la precisión de los datos financieros.
- El sistema web facilitaría el análisis de datos financieros, lo que me permitiría identificar tendencias y áreas de mejora con mayor facilidad.
- Si el sistema se integra con la gestión de inventario, sería más fácil mantener un control preciso de las existencias de productos.

Observación:

Los cambios que menciona son beneficios importantes que podría obtener de la implementación de un sistema web. Es fundamental reconocer que, si bien estos beneficios son significativos, la transición a un sistema web también requerirá tiempo y esfuerzo en términos de capacitación y adaptación. Además, mantener una

comunicación abierta y eficaz durante el proceso de implementación ayudará a abordar cualquier desafío que pueda surgir.

Conclusiones:

La conclusión general es la implementación de un sistema web para la gestión financiera y se percibe como una oportunidad para mejorar significativamente la eficiencia y precisión de las operaciones financieras. Con referencia a la pregunta 5, el gerente reconoce la importancia de la automatización de tareas, el acceso en tiempo real a datos financieros, la generación de informes y la integración con la gestión de inventario.

Sin embargo, también se destaca que la transición a un sistema web requerirá capacitación y adaptación por parte del personal. Además, se reconoce la necesidad de abordar desafíos como la capacitación, la integración con los procesos actuales y la seguridad de datos.

Finalmente, la implementación de un sistema web se ve como una oportunidad para mejorar la gestión financiera, pero se reconoce que es importante gestionar la transición de manera efectiva y abordar los desafíos potenciales para lograr una transición exitosa.

2.2.3.2 Entrevista realizada al personal de ventas

a. Primera Persona de ventas.

ENTREVISTA DIRIGIDA AL PERSONAL VENTAS DE LA MICROEMPRESA “ZOMI”

Sistema de gestión financiera

Fecha: 17/10/2023

Entrevistador: Paul Hidalgo

Entrevistado: Evelyn Barreno

Cargo: ventas

Objetivo:

Analizar los procesos de ventas e inventario de productos la Microempresa “ZOMI”.

Introducción:

Entrevistador: Buen día, agradecemos tu participación en esta entrevista que tiene como objetivo obtener información valiosa para la implementación exitosa de nuestro nuevo sistema web de gestión financiera en "ZOMI". Queremos escuchar tus opiniones

y experiencias para garantizar que el sistema se adapte de la mejor manera a nuestras operaciones diarias. ¿Estás listo para comenzar?

Preguntas:

Pregunta 1: ¿Cuál es el proceso actual de venta de la microempresa "ZOMI", desde la recepción de un pedido hasta la entrega del producto al cliente?

Respuesta:

Nuestro proceso actual de venta en "ZOMI" se centra en la interacción directa con los clientes. Cuando un cliente quiere comprar un producto, ya sea por pedido o eligiéndolo en nuestra tienda, registramos sus elecciones y cantidades en un papel. Luego, calculamos el total y el cliente paga en efectivo. Después de la compra, si es necesario, coordinamos la entrega del producto.

Observación:

El proceso de ventas actual proporciona una base importante para considerar cómo la implementación de un sistema de ventas más eficiente.

Pregunta 2: ¿Has enfrentado desafíos en la consulta de cantidades y/o precios de los productos durante la realización de una venta? Si es así, ¿podrías describir algunos ejemplos específicos?

Respuesta:

En efecto, hemos enfrentado algunos desafíos al consultar cantidades y precios de productos durante las ventas, considerando que no contamos con un sistema de ventas y de gestión de inventario. Por ejemplo, en ocasiones, los clientes consultan la disponibilidad de un producto específico y no tenemos una manera rápida de verificar si está en stock en ese momento. También, debido a la falta de un registro actualizado de precios, pueden surgir malentendidos cuando los precios cambian.

Observación:

La falta de un sistema de ventas y de gestión de inventario ha generado desafíos en la consulta de cantidades y precios, lo que puede afectar la satisfacción del cliente y la eficiencia de las ventas.

Pregunta 3: ¿De qué manera se lleva a cabo el registro de inventario de productos actualmente en la microempresa "ZOMI"?

Respuesta:

El registro de inventario de productos en 'ZOMI' se lleva a cabo de manera manual. Utilizamos hojas de inventario donde anotamos la cantidad de productos que tenemos en stock. Sin embargo, este proceso manual a veces se torna complicado, ya que puede haber errores humanos en el registro, y en ocasiones, olvidamos registrar una venta, lo que puede causar diferencias en las cantidades reales en stock.

Observación:

El proceso de registro manual de inventario puede ser propenso a errores y dificultades. Esto destaca la importancia de contar con una solución que ofrezca una gestión de inventario más precisa y eficiente, especialmente cuando la empresa opera sin un sistema formal de gestión.

Pregunta 4: ¿cuáles son las expectativas o necesidades que crees que este sistema debería cumplir para mejorar el desempeño financiero de la microempresa “ZOMI”?

Respuesta:

En mi opinión, un sistema de ventas y gestión de inventario debería desempeñar varias necesidades como las de llevar un control actualizado del inventario para evitar pérdidas debido a productos agotados. También, esperaría que el sistema facilite la emisión de notas de ventas y el registro de ventas de manera eficiente. Además, sería beneficioso si el sistema generara informes de ventas y márgenes de beneficio para evaluar nuestro desempeño.

Observación:

Las necesidades destacan la importancia de contar con un control preciso de inventario y la eficiencia en la emisión de notas de venta y el registro de ventas. La generación de informes también se menciona como una herramienta valiosa para evaluar el desempeño financiero de la empresa.

Pregunta 5: En tu opinión, ¿cuáles son los posibles beneficios que este sistema puede aportar a la eficiencia y productividad en ventas de la microempresa “ZOMI”?

Respuesta:

Desde mi perspectiva, el sistema puede aportar significativamente a la eficiencia y productividad en ventas de la microempresa "ZOMI" al permitir una gestión precisa del inventario, evitando excesos o faltantes de productos. Además, facilita la supervisión de las ganancias y gastos de la microempresa.

Observación:

El sistema web aporta beneficios a la microempresa "ZOMI", mejorando la gestión financiera, lo que es esencial para el éxito y la productividad.

Pregunta 6: ¿Qué tipo de información financiera consideras crucial y que debería estar fácilmente accesible para el personal de ventas en este sistema?

Respuesta:

La información financiera que debería estar fácilmente accesible para el personal de ventas en este sistema incluye: los costos y cantidades de los productos disponibles, así como las ganancias y gastos asociados a mes.

Observación:

Es fundamental destacar la disponibilidad de información actualizada sobre inventario y precios es esencial para brindar un servicio de ventas eficiente ya que el personal de ventas utilizará con frecuencia el sistema web para acceder a esta información.

Pregunta 7: Desde tu rol en ventas, ¿visualizas algún desafío potencial que podríamos enfrentar durante la implementación y adaptación a este nuevo sistema?

Respuesta:

Como personal de ventas, podría visualizar un posible desafío durante la implementación y adaptación a este nuevo sistema, que es el tiempo de aprendizaje inicial para familiarizarse con la interfaz y las funcionalidades del sistema. Esto puede requerir tiempo y capacitación adecuada

Observación:

Es esencial tener en cuenta que la adaptación al nuevo sistema puede generar cierta resistencia inicial por parte del equipo de ventas debido a la introducción de nuevos cambios en sus procesos habituales.

Pregunta 8: ¿En qué áreas crees que este sistema podría automatizar tareas manuales en ventas y ayudar a ahorrar tiempo en relación con las operaciones financieras?

Respuesta:

Creo que un sistema de ventas y gestión financiera puede automatizar tareas manuales en ventas, especialmente en áreas como la generación de informes financieros,

seguimiento de inventario, cálculo de ganancias y gastos, así como la gestión de transacciones y emisión de notas de ventas. Al eliminar la necesidad de realizar estos procesos de forma manual.

Observación:

La respuesta resalta claramente cómo la automatización de tareas manuales puede ahorrar tiempo y mejorar la eficiencia en ventas y operaciones financieras. Estos son beneficios importantes de un sistema de ventas y gestión de inventario.

Conclusiones:

La información obtenida sobre los desafíos actuales que enfrenta la microempresa "ZOMI" en sus operaciones de ventas, ha destacado la falta de un sistema de ventas, lo que ha llevado a problemas en la gestión de inventario, la consulta de cantidades y precios de productos, y la generación manual de documentos financieros. Además, en la pregunta 8 la vendedora, ha subrayado la importancia de la automatización de tareas manuales a través de un sistema de ventas y gestión financiera, lo que podría ahorrar tiempo y mejorar la eficiencia en ventas y operaciones financieras. Sus respuestas indican claramente que la implementación de un sistema adecuado será beneficioso para la microempresa.

b. Segunda Persona de ventas.

ENTREVISTA DIRIGIDA AL PERSONAL DE VENTAS LA MICROEMPRESA
"ZOMI"

Sistema de gestión financiera

Fecha: 22/10/2023

Entrevistador: Paul Hidalgo

Entrevistado: Juan Barreno

Cargo: ventas

Objetivo:

Analizar los procesos de control de ventas e inventario de productos la Microempresa "ZOMI".

Introducción:

Entrevistador: Buen día, agradecemos tu participación en esta entrevista que tiene como objetivo obtener información valiosa para la implementación exitosa de nuestro nuevo sistema web de gestión financiera en "ZOMI". Queremos escuchar tus opiniones y experiencias para garantizar que el sistema se adapte de la mejor manera a nuestras operaciones diarias. ¿Estás listo para comenzar?

Preguntas:

Pregunta 1: ¿Cuál es el proceso actual de venta de la microempresa "ZOMI", desde la recepción de un pedido hasta la entrega del producto al cliente?

Respuesta:

En "ZOMI", nuestro proceso de venta actual es bastante simple. Cuando un cliente desea realizar una compra, generalmente nos visitan nuestra tienda física. Tomamos nota de los productos que desean y sus cantidades. Luego, calculamos el total de la compra. El cliente paga en efectivo, si es necesario, coordinamos la entrega del producto en ese momento.

Observación:

Se observa que la empresa actualmente opera sin un sistema de ventas o pedidos formal. Esto proporciona un punto de partida importante para la posible implementación de un sistema más eficiente.

Pregunta 2: ¿Has enfrentado desafíos en la consulta de cantidades y/o precios de los productos durante la realización de una venta? Si es así, ¿podrías describir algunos ejemplos específicos?

Respuesta:

Sí, hemos enfrentado desafíos en la consulta de cantidades y precios de productos debido a la falta de un sistema de ventas o pedidos. Por ejemplo, a veces un cliente pide un producto que no está en la tienda en ese momento, y no tenemos una forma rápida de verificar el inventario actual. También, en ocasiones, los precios de algunos productos pueden variar, y esto ha llevado a confusión o malentendidos con los clientes.

Observación:

Entiendo que la falta de un sistema de ventas o pedidos ha llevado a desafíos en la gestión de inventario y precios, lo que podría mejorarse con una solución más eficiente.

Pregunta 3: ¿De qué manera se lleva a cabo el registro de inventario de productos actualmente en la microempresa “ZOMI”?

Respuesta:

Actualmente, el registro de inventario de productos en "ZOMI" se realiza de manera manual. Usamos registros físicos para llevar un seguimiento de la cantidad de productos que tenemos en stock. Cuando vendemos un producto, en ocasiones se olvida de registrar la venta y eso ocasiona que se descuadre las cantidades en stock.

Observación:

La gestión manual de inventario puede ser laboriosa y menos precisa. Esto resalta la necesidad de mejorar la gestión de inventario mediante un sistema más eficiente.

Pregunta 4: ¿cuáles son las expectativas o necesidades que crees que este sistema debería cumplir para mejorar el desempeño financiero de la microempresa “ZOMI”?

Respuesta:

Creo que un sistema de ventas y gestión de inventario debería cumplir varias expectativas para mejorar el desempeño financiero de "ZOMI". En primer lugar, debería permitirnos llevar un registro preciso del inventario en tiempo real, lo que ayudaría a evitar pérdidas debido a productos agotados. Además, espero que el sistema agilice el proceso de ventas.

Observación:

La importancia de una solución que mejore la precisión y eficiencia para mejorar el desempeño financiero de la empresa. Estas serán consideraciones clave al seleccionar una solución adecuada.

Pregunta 5: ¿cuáles son los posibles beneficios que este sistema puede aportar a la eficiencia y productividad en ventas de la microempresa “ZOMI”?

Respuesta:

En mi opinión, la implementación de un sistema de ventas y gestión de inventario podría aportar una serie de beneficios significativos. En primer lugar, nos ayudaría a eliminar los errores humanos en el registro de ventas y en la gestión del inventario, lo que mejoraría la precisión y reduciría pérdidas económicas. También, nos permitiría atender a los clientes de manera más rápida y eficiente, ya que podríamos acceder a información actualizada sobre productos y precios al instante. Además, al automatizar ciertas tareas, como la actualización de inventario y la generación de notas de venta.

Observación:

La respuesta resalta los beneficios clave que se pueden obtener con la implementación de un sistema. La eficiencia, la precisión y la productividad son factores críticos para el éxito de cualquier empresa.

Pregunta 6: ¿Qué tipo de información financiera consideras crucial y que debería estar fácilmente accesible para el personal de ventas en este sistema?

Respuesta:

Considero que varios tipos de información financiera son cruciales y deberían estar fácilmente accesibles para el personal de ventas en este sistema. En primer lugar, la disponibilidad del inventario actualizado en tiempo real es fundamental. Esto nos permitiría saber si un producto está en stock y cuántas unidades están disponibles. Además, la información de precios actualizados sería esencial para proporcionar a los clientes información precisa sobre costos. También, tener acceso a datos de ventas diarias y márgenes de beneficio

Observación:

Acceder a esta información sería esencial para una gestión eficiente y sería de gran ayuda agilizar procesos en las ventas.

Pregunta 7: Desde tu rol en ventas, ¿visualizas algún desafío potencial que podríamos enfrentar durante la implementación y adaptación a este nuevo sistema?

Respuesta:

Sí, desde mi perspectiva en ventas, veo algunos desafíos potenciales durante la implementación y adaptación a un nuevo sistema. Em primer lugar La capacitación adecuada será esencial para garantizar que todos se sientan cómodos utilizando el nuevo sistema. Además, es importante asegurarse de que el sistema sea fácil de usar y que no agregue complejidad adicional a nuestras tareas diarias.

Observación:

Estos son desafíos válidos que deben abordarse durante la implementación de un nuevo sistema. La capacitación y la usabilidad son factores clave para garantizar una transición exitosa.

Pregunta 8: ¿En qué áreas crees que este sistema podría automatizar tareas manuales en ventas y ayudar a ahorrar tiempo en relación con las operaciones financieras?

Respuesta:

Creo que un sistema de ventas y gestión de inventario podría automatizar varias tareas manuales en ventas y, en consecuencia, ahorrar tiempo en las operaciones financieras. En primer lugar, la generación automática de nota de ventas y recibos basada en los detalles del pedido ahorraría tiempo en la creación manual de documentos financieros. Además, la actualización en tiempo real del inventario reduciría la necesidad de contar productos físicamente y evitaría errores humanos en el registro.

Observación:

La respuesta complementa con la persona de Ventas 1. Ambas destacan la importancia de la generación automática de documentos, la actualización de inventario en tiempo real y el seguimiento financiero automatizado. Estos aspectos son fundamentales para mejorar la eficiencia en ventas y operaciones financieras.

Conclusiones:

En conclusión, la implementación de un sistema de ventas y gestión de inventario parece ser una necesidad evidente para mejorar la eficiencia, la precisión y el desempeño financiero de la microempresa "ZOMI". La capacitación y la usabilidad son consideraciones importantes durante este proceso de transición. La automatización de tareas manuales proporcionaría beneficios adicionales en términos de ahorro de tiempo y mejora de la productividad.

2.2.3.3 Entrevista realizada al personal de producción

ENTREVISTA DIRIGIDA AL PERSONAL DE PRODUCCIÓN DE LA
MICROEMPRESA "ZOMI"

Sistema de gestión financiera

Fecha: 22/10/2023

Entrevistador: Paul Hidalgo

Entrevistado: Lida Pérez

Cargo: producción

Objetivo:

Analizar la gestión de la producción de la Microempresa “ZOMI” con el fin de identificar áreas de mejora

Introducción:

Entrevistador: Buen día, agradecemos tu participación en esta entrevista que tiene como objetivo obtener información valiosa para la implementación exitosa de nuestro nuevo sistema web de gestión financiera en "ZOMI". Queremos escuchar tus opiniones y experiencias para garantizar que el sistema se adapte de la mejor manera a nuestras operaciones diarias. ¿Estás listo para comenzar?

Preguntas:

Pregunta 1. ¿Puede describir brevemente su papel y responsabilidades como persona encargada de producción en ZOMI?

Respuesta:

Mi principal responsabilidad en “ZOMI” se centra en la elaboración de figuras de porcelana fría. Mi función implica la preparación de la masa y la creación de las figuras hasta cierto paso, pero sin la terminación del producto la terminación es realizada por la gerente.

Observación:

La persona encargada de la producción no realiza en su totalidad el producto, esto quiere decir que el producto solo esta parcialmente terminado, pero aún no está completamente listo para la venta.

Pregunta 2. ¿Cuál es el criterio que utilizan actualmente para determinar las cantidades de productos a producir?

Repuesta:

Hay dos criterios, uno se realiza los productos alusivos a fechas festivas, y las cantidades dependen de la cantidad de la materia prima, y la otra es bajo pedido según el cliente lo requiera.

Observación:

En fechas festivas no tienen un control de producción relacionado a cantidades lo que implica pérdidas económicas si existen sobrantes.

Pregunta 3. ¿Cuáles son los principales desafíos que enfrenta la producción en ZOMI en la actualidad?

Respuesta:

Uno de los desafíos a los que nos enfrentamos es la falta de un seguimiento de nuestra producción. A menudo, producimos en exceso ciertos productos, mientras que otros escasean. Esto se debe a la falta de un registro adecuado de la producción, lo que significa que a veces fabricamos productos sin tener una idea clara de cuáles son necesarios y cuáles están en exceso.

Observación:

No existe un control adecuado de la producción y no sabe la cantidad total exacta de los productos que realizo.

Pregunta 4. ¿Actualmente el registro de la producción se realiza por medio de un software o manualmente?

Respuesta:

Solo cuando es bajo pedido tenemos un control manual para no exceder la cantidad de productos caso contrario no se lleva control de la producción.

Observación:

En gran parte de la producción no lleva un registro adecuado lo que existe desconocimiento de las cantidades exactas producidas.

Pregunta 5. ¿Qué tipo de información sería importante para mejorar la producción?

Respuesta:

Para mí sería importante saber las cantidades de productos que hay en el inventario y con esto podemos evitar el exceso de producción o escases de un solo productos.

Observación:

Menciona la importancia de llevar un control de registro de la producción para mejorar la eficiencia en el área de producción.

Pregunta 6. ¿Cómo cree que la implementación de un sistema web podría beneficiar a “ZOMI” en términos de producción?

Respuesta:

En el área de producción, creo que un sistema web mejoraría la producción. Además, la forma en que se registra la cantidad producida sería más detallada, por lo que no habría excesos ni escases de productos.

Observación:

En base a la respuesta la implementación de un sistema web en la microempresa “ZOMI” mejoraría en cierto grado el área de producción al tener una mejor gestión en relación con los productos.

Pregunta 7. Desde tu rol en producción, ¿visualizas algún desafío al implantar y este nuevo sistema?

Respuesta:

El desafío para mi es la utilización del sistema web porque no he utilizado un sistema web anteriormente

Observación:

Se debe realizar una capacitación para evitar inconvenientes con el sistema web.

Conclusión:

En conclusión, la implementación de un sistema web en la producción de "ZOMI" se presenta como una oportunidad para mejorar el control y la eficiencia en la producción, evitando problemas de inventario y produciendo la cantidad adecuada de productos. La capacitación del personal en el uso del sistema sería esencial para una implementación exitosa.

2.2.3.4 Resultados de la matriz de observación de venta

Tabla 2. Matriz de observación para el proceso de venta

FECHA:		17/10/2023			
HORA DE INICIO:		11:00			
HORA DE FIN:		14:00			
OBSERVADOR:		PAUL HIDALGO			
NO	CATEGORÍA DE OBSERVACIÓN	DESCRIPCIÓN DE LA SITUACIÓN	ACCIONES Y PROCESOS OBSERVADOS	ESCALA (1-5)	COMENTARIOS /NOTAS
1	Toma de pedidos	Proceso de toma de pedidos corto plazo	La vendedora anota el pedido del cliente en un cuaderno. Anota las especificaciones, el tiempo de entrega y el valor inicial que el cliente cancela.	3	Se evidencia que las anotaciones son poco entendibles lo que dificulta en la realización del pedido.
		Proceso de toma de pedidos al instante	La vendedora solo escucha las especificaciones del pedido.		La vendedora pregunta algunas veces al cliente si el pedido es de varios productos o solo de uno esto hace que el cliente se incomode.
2	Disponibilidad de productos	Disponibilidad de productos	La vendedora verifica la disponibilidad del producto observando y buscando los productos en el almacén.	2	La búsqueda y el conteo de un producto puede ser un proceso lento, y los clientes suelen valorar la rapidez en saber si el producto que desean está disponible o no.

		Cantidad de productos	Si el o los productos están disponible la vendedora realiza un conteo y verifica si la cantidad satisface al pedido		
		No hay suficientes productos en stock.	Si el producto no existe en la vendedora explica el tiempo de realización y de la entrega del producto.		Los clientes quieren que los productos sean entregados en ese momento, y en caso de no tener el producto, hace que el cliente quede insatisfecho.
3	Manejo de Pago	Proceso de pago en ventas bajo pedido.	la vendedora realiza los cálculos manualmente del valor total y le explica al cliente que debe abonar un porcentaje con un mínimo del 25%	3	Pueden existir errores en los cálculos, además, pueden resultar en ciertas demoras y lo que se busca en una venta es la rapidez.
		Proceso de pago de ventas al instante.	La vendedora realiza los cálculos manualmente del valor total del o los productos		
4	Facturación y Documentación	Generación de factura	La vendedora menciona que su negocio esta bajo el régimen RIMPE es decir Negocio Popular, por lo que solo emiten notas de venta y no facturas.	4	El proceso de generar una nota de venta toma tiempo esto puede ocasionar que el cliente no pida su comprobante de compra, pero la gerente menciona que es una obligación emitir dicho documento.
		Documentación	La vendedora realiza una nota de venta manualmente, pero si es cantidades mininas en la venta el cliente no pide la nota de venta.		

CONCLUSIÓN:
En resumen, la microempresa enfrenta desafíos en su proceso de ventas, como la demora en la búsqueda de productos y la falta de un inventario exacto. Esto obliga a la vendedora a realizar conteos en el momento, lo que puede afectar la satisfacción del cliente. En la actualidad, los clientes valoran la rapidez y desean tener la seguridad de encontrar el producto que buscan sin necesidad de esperar o verificar su disponibilidad.

2.2.3.5 Resultados de la matriz de observación de compra

Tabla 3. Matriz de observación para el proceso de compra

FECHA:		17/10/2023	
HORA DE INICIO:		11:00	
HORA DE FIN:		14:00	
OBSERVADOR:		PAUL HIDALGO	
NO	DETALLES DE LA ADQUISICIÓN	PROCESO DE ADQUISICIÓN	COMENTARIOS /NOTAS
1	Revisión de productos	Selección de productos faltantes	Antes de realizar una compra de un producto, primero se realiza una observación y conteo manual de los productos existentes en el almacén y según el producto que este en bajas cantidades o agotado, se anota en una hoja para luego realizar su pedido.
		Cantidad de productos	La cantidad de productos que se elige depende de la demanda actual. Si las ventas aumentan, se ajusta el pedido de compra para garantizar un suministro adecuado.

2	Proveedor(es)	Selección de Proveedor	Según los productos faltantes se elige al proveedor, pero antes se estima los precios, ubicación y la calidad, esto se compara entre dos o más proveedores para evitar gastos innecesarios en la compra.
		Tipo de proveedor	Hay dos tipos de proveedores: los que son artesanos y por ley no emiten factura, pero entregan nota de venta y están los que suministran materias primas que si están obligados a emitir factura.
3	Productos Adquiridos	Negociación y Acuerdo de Compra	Durante la negociación y el acuerdo de compra, se estableció una comunicación efectiva con el proveedor. Se acordaron los detalles del pedido, incluyendo cantidades, precios, plazo de entrega.
4	Manejo de pago	Proceso de Pago	El proceso de pago se llevó a cabo sin problemas. Se acordó pagar el 50% del costo total por adelantado y el 50% restante a la entrega de los productos.
5	Facturación	Recibo de la compra	La compra se realizó en dos partes la primera con el 50%, lo cual no llevó ningún registro y la segunda parte de la compra entrega la nota de venta con la cantidad de los productos.

CONCLUSIÓN:

En resumen, el proceso de compra en la microempresa enfrenta complicaciones al seleccionar productos que se necesita, esto debido a la falta de un inventario preciso. Esto conlleva a observar y realizar un conteo de los productos para saber cuáles son necesarios, lo que resulta en pérdidas de tiempo. Además, el proceso de pago y recibo de la compra presenta deficiencias, ya que en el primer caso no se emiten comprobantes ni se registra el abono del 50%, y en el segundo, la nota de venta no se incorpora en ningún sistema, simplemente se almacena sin seguimiento.

2.2.4 Procesamiento y análisis de datos

En base a los resultados obtenidos de las entrevistas y matrices de observaciones realizadas, se logró determinar que:

- **Desafíos actuales en la Microempresa:** Se ha identificado una serie de desafíos en la gestión financiera y operativa de la microempresa "ZOMI". Estos desafíos incluyen la gestión manual del inventario, demoras en las ventas, la falta de registros precisos en la gestión de compras, y la falta de conocimiento de las cantidades exactas de productos en inventario.
- **En el área de compra y venta:** Existen deficiencias, como la falta de gestión del inventario; esto resulta muy demoroso al momento de verificar la disponibilidad de un producto en stock. Además, existe demoras durante todo el proceso de venta, ya que la vendedora realiza todo de manera manual, desde la búsqueda de productos hasta la emisión de la nota de venta. En las compras, también se enfrentan retrasos debido a la carencia de información sobre el inventario, lo que lleva a buscar manualmente los productos que faltan para realizar una compra. Además, se identifican problemas en el registro de las compras, ya que no se lleva un seguimiento adecuado de cada compra.
- **En el área de producción:** La implementación del sistema web se percibe como una solución que mejoraría la eficiencia y la productividad de la microempresa "ZOMI". Con el fin de llevar un registro más eficiente de sus productos, así evitando una producción innecesaria.
- **Reconocimiento de la Necesidad de Cambio:** Los entrevistados reconocen la necesidad de implantar un sistema web para mejorar la eficiencia, la gestión financiera y operativa de la microempresa "ZOMI". Esto además incluye el seguimiento preciso de los inventarios y la supervisión de los costos y gastos.
- **Se puede concluir que la implementación del sistema web dentro la microempresa "ZOMI", promete un futuro alentador dentro la gestión financiera.**

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis de procesos

En base a la información recopilada en la microempresa, se pudo determinar los procesos de cómo se desarrollan las funciones de compra, venta e inventario de productos dentro de la misma.

3.1.1 Diagramas de proceso con la gestión financiera actual

Actualmente, los procesos se gestionan de forma manual, cada uno de los procesos tiene inconsistencias, es decir, uno depende de otro y si uno está mal gestionado perjudica a los demás procesos.

a. Proceso actual para la adquisición de mercadería

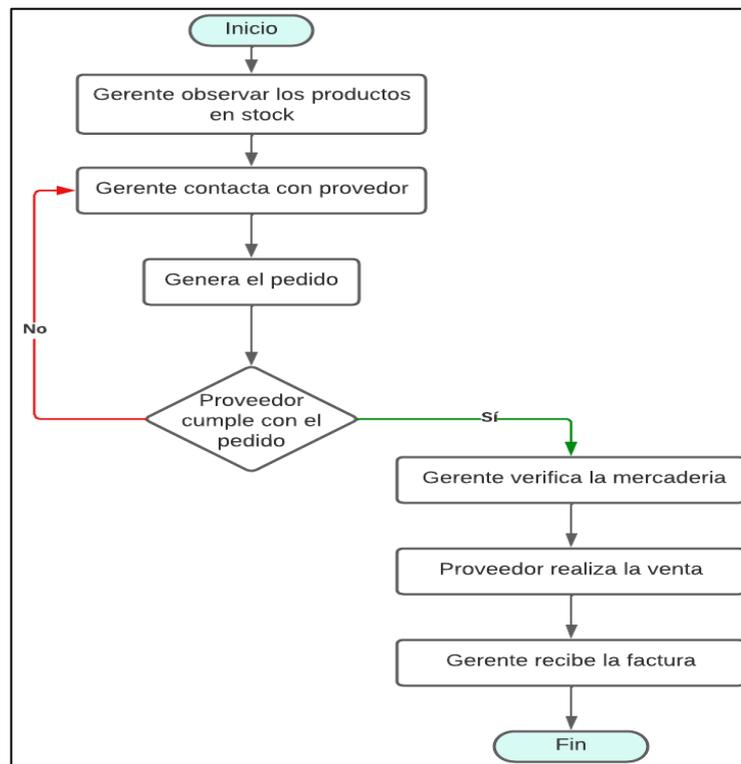


Figura 2. Proceso actual de compra de mercadería

En la Figura 2. Proceso actual de compra de mercadería describe como el gerente de la microempresa realiza cada paso del proceso de compras, todos los cuales son manualmente, especialmente el paso de la observación de los productos que tiene en stock y esto depende de la decisión de los productos que se deben comprar o no, esto ocasiona demoras en tomar una decisión o puede ocasionar la compra de productos innecesarios.

b. Proceso actual para la venta de productos

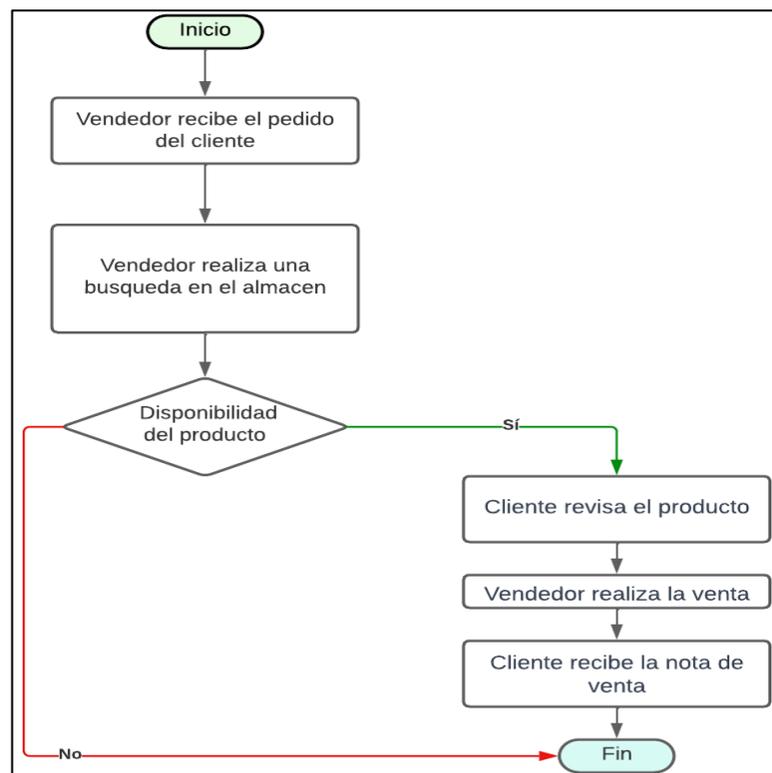


Figura 3. Proceso actual de venta de productos

En la Figura 3. Proceso actual de venta de productos describe el proceso de ventas que tiene la microempresa, la vendedora manifiesta que cada paso se realiza manualmente. Hay que especificar que existen pasos que generan mayores pérdidas de tiempo que los demás por ejemplo en el paso de buscar un producto si está disponible o no al momento de emitir una nota de venta, eso ocasiona retrasos en la venta o insatisfacción al cliente.

c. Proceso actual para el registro y almacenado de productos

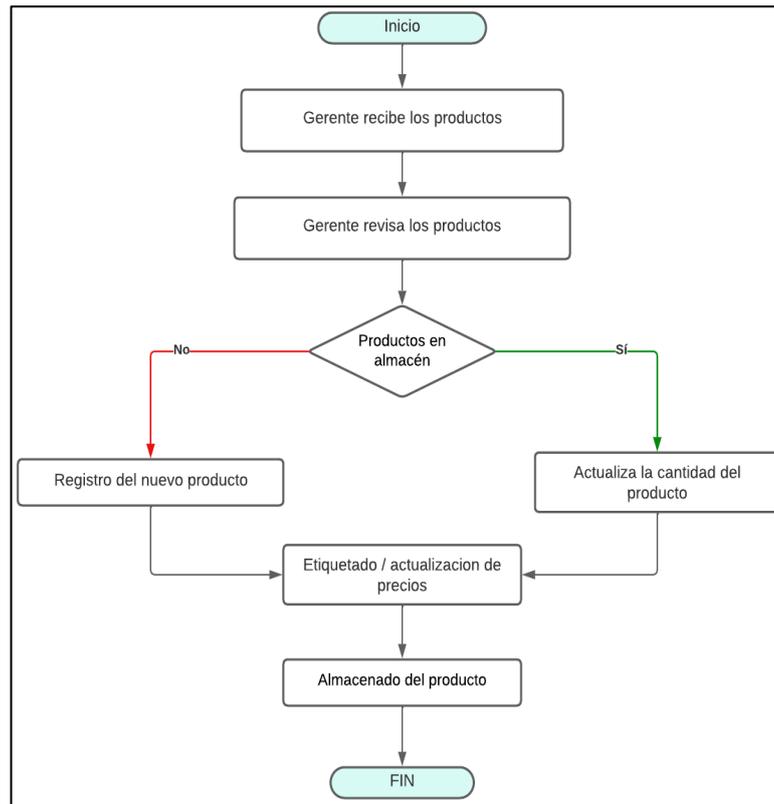


Figura 4 Proceso de registro y almacenamiento de productos

En la Figura 4 Proceso de registro y almacenamiento de productos describe como se registran los productos dentro de la microempresa “ZOMI”, cabe mencionar que estos procesos se realizan de forma manual, lo que ha provocado que en varias ocasiones estos registros tengan inconsistencias, es decir que esa información no sea cien por ciento real, cabe mencionar también que otro problema que existe dentro de la microempresa es buscar el cuaderno de apuntes donde se registran los datos de los productos existentes lo que ha provocado pérdidas de tiempo y una mala gestión del inventario.

3.1.2 Diagrama de procesos optimizado

a) Compra e ingreso de productos

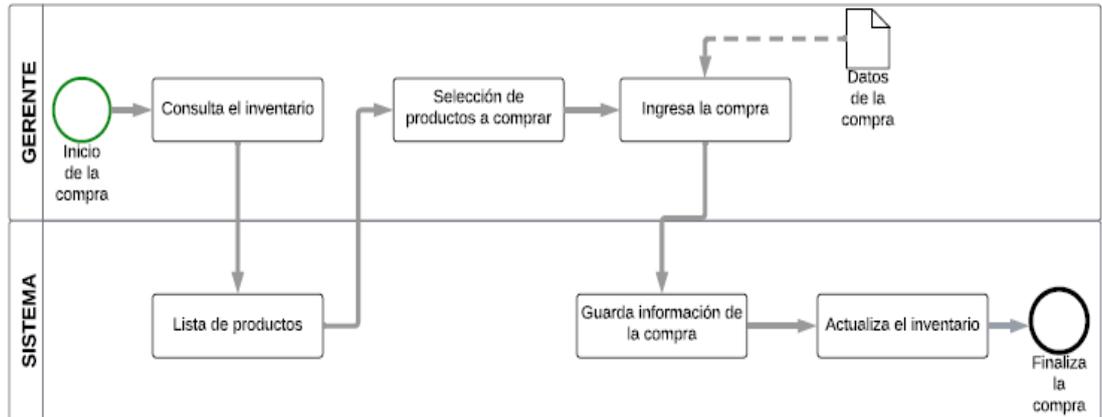


Figura 5 Flujo de proceso para compra e ingreso de productos

b) Venta de productos

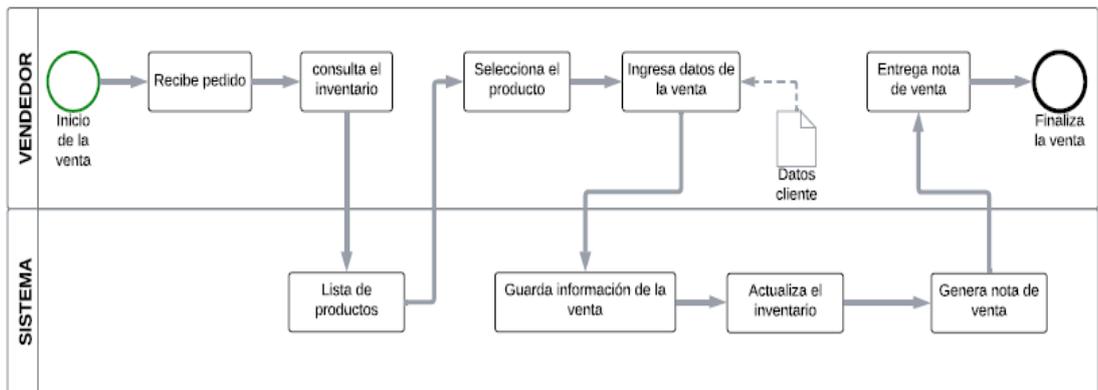


Figura 6. Flujo de proceso para la venta de producto

Los procesos manuales se automatizarán en gran medida a través del sistema web, tal como se ilustra en la Figura 5 Flujo de proceso para compra e ingreso de productos y la Figura 6. Flujo de proceso para la venta de producto, lo que conlleva una reducción considerable de tiempos y agilización de los procesos.

3.2 Proveedores de servicio en la nube

La adopción de soluciones y plataformas en la nube es algo cada vez más frecuente. Las organizaciones hoy en día utilizan los entornos digitales para trabajar y aportar a sus operaciones eficiencia y escalabilidad. Sin embargo, para el presente proyecto se escogió a los tres principales proveedores de nubes, el primero es AWS (Amazon Web Services) es uno de los proveedores más grandes y ofrece una amplia gama de servicios dentro del mismo, el segundo es Microsoft Azure este ofrece una gran variedad de servicios y es muy popular entre las empresas que ya utilizan este producto de Microsoft, tal como Windows Server o Office, por último tenemos el GCP (Google Cloud Platform) que ofrece servicios similares a AWS y Azure. Además, estas herramientas específicamente son utilizadas para la inteligencia artificial y análisis de datos.

Tabla 4. Comparación entre proveedores de servicio en la nube

PARAMETROS	AWS (Amazon Web services)	Microsoft Azure	Google Cloud Platform (GCP)
<p>Servicios Serverless Disponibles</p>	<ul style="list-style-type: none"> • AWS Lambda • AWS Fargate • Amazon EventBridge • Amazon Simple Queue Service • Amazon Simple Notification Service (SNS) • Amazon API Gateway • AWS AppSync • Amazon Simple Storage Service (Amazon S3) • Amazon Elastic File System • Amazon DynamoDB • Amazon RDS Proxy • Amazon Aurora Serverless • Amazon Redshift Serverless • Amazon Neptune Serverless • Amazon OpenSearch Serverless 	<ul style="list-style-type: none"> • Azure Functions • Azure Logic Apps • Azure Event Grid • Azure Durable Functions • Azure API Management • Azure Static Web Apps • Azure Cosmos DB • Azure Storage • Azure Service Bus • Azure Key Vault • Azure App Service 	<ul style="list-style-type: none"> • Google Cloud Functions • Google Cloud Run • Google App Engine • Google Cloud Pub/Sub • Google Cloud Storage • Google Cloud Scheduler • Google Cloud Firestore

PARAMETROS	AWS (Amazon Web services)	Microsoft Azure	Google Cloud Platform (GCP)
Serverless framework	<ul style="list-style-type: none"> • AWS Serverless Application Mode (SAM) • AWS Cloud Development Kit • AWS Chalice • Serverless Framework • Claudia.js • Pulumi • SPARTA • Flogo • APEX • Terraform • Architect 	<ul style="list-style-type: none"> • Serverless Framework • Pulumi • Terraform 	<ul style="list-style-type: none"> • Serverless Framework • Google Cloud Functions Framework • Pulumi • Terraform
Seguridad en la nube	Cuenta con plataformas seguras, adaptables y resistentes a las amenazas	Cuenta con plataformas seguras, adaptables y resistentes a las amenazas	Cuenta con plataformas seguras, adaptables y resistentes a las amenazas
Documentación y Recursos	AWS proporciona una amplia documentación, tutoriales y una comunidad activa. Ofrece recursos para principiantes y usuarios avanzados.	Azure cuenta con documentación, tutoriales y una comunidad sólida. Ofrece recursos para principiantes y una amplia base de conocimientos.	GCP ofrece documentación completa y tutoriales, con una comunidad en crecimiento. Proporciona recursos para principiantes y usuarios intermedios.

PARAMETROS	AWS (Amazon Web services)	Microsoft Azure	Google Cloud Platform (GCP)
Modelo de Precios	<ul style="list-style-type: none"> • Niveles introductorios gratuitos: <ul style="list-style-type: none"> ▪ Pruebas gratuitas ▪ 12 meses de uso gratuito ▪ Gratis para siempre • Nivel de pago por uso. 	<ul style="list-style-type: none"> • Niveles introductorios gratuitos: <ul style="list-style-type: none"> ▪ 12 meses ▪ Gratis para siempre (55 servicios) ▪ \$200 en créditos • planes de pago <ul style="list-style-type: none"> ▪ pago por uso ▪ premium 	<ul style="list-style-type: none"> • Periodo de prueba (\$300 en créditos gratuitos) • Nivel de pago por uso
Regiones	Está presente en 32 regiones y 102 zonas a través de América, Europa, Oriente Medio, África y Asia Pacífico. AWS tiene la mayor huella global de infraestructura en la nube	En más de 60 regiones distribuidas a lo largo de América, Europa, África y Asia.	37 regiones y 112 zonas a lo largo de América, Europa y Asia-Pacífico.
Liderazgo en Mercado	32% de la cuota del mercado Pioneros en la industria	20% de la cuota del mercado.	En crecimiento, con el 9% de la cuota del mercado enfocado en análisis de datos y aprendizaje automático.

Finalizado el análisis de los principales proveedores de servicios nubes, se concluye que los tres cumplen con el propósito del proyecto, pero GCP se enfoca más al análisis de datos, en cambio AWS y Azure, tienen similares características, para este caso se opta por trabajar con AWS, debido que es una plataforma en la nube más utilizada y ofrece una variedad de servicios orientados a la arquitectura serverless que se adapta a las necesidades del proyecto.

Además, cuenta con una amplia documentación detallada y entendible para el desarrollo del sistema y con una comunidad activa la cual puede apoyar en el desarrollo de la aplicación, contando también con blog dedicados al tema de serverless.

3.2.1 Modelo de pago y características técnicas de los servicios orientados a la arquitectura serverless en AWS

Los servicios más utilizados en la plataforma de AWS para el desarrollo de aplicaciones con arquitectura serverless son: AWS lambda, DynamoDB, Amazon API Gateway, el cual se realiza un análisis comparativo para conocer los precios de consumo de cómputo y almacenamiento para el proyecto.

Tabla 5. Modelo de pago y características técnicas de los servicios

SERVICIOS SERVERLESS	Nivel gratuito	Nivel de Pago	
AWS lambdas	400ms por solicitud y 1024 MB de memoria por función	400ms por solicitud y 1024MB de memoria por función	
	400 000 GB/segundos de tiempo de computación al mes	400.000 GB-s total compute	6,66 USD por mes
	Un millón de solicitudes al mes	2,000,000 de solicitudes al mes	0.40 USD por mes
	100 GiB de secuencia de Request HTTP al mes	100 GiB de secuencia de Request HTTP	0,80 USD por mes
	GRATUITO PARA SIEMPRE POR MES		

SERVICIOS SERVERLESS	Nivel gratuito	Nivel de Pago		
Dynamo DB	25 GB de almacenamiento al mes	Unidades de solicitud de escritura (WRU)	1,25 USD por millón de unidades de solicitud de escritura	
	hasta 200 millones de solicitudes de lectura y escritura al mes	Unidades de solicitud de lectura (RRU)	0,25 USD por millón de unidades de solicitud de lectura	
	MIENTRAS NO SE EXCEDA EL LIMITE DE LA CAPA GRATUITA ESTE SIGUE SIENDO GRATUITO.	Almacenamiento de datos en Tabla DynamoDB Standard	0,25 USD por GB-mes	
Amazon API Gateway	Un millón de llamadas a API recibida para las API de REST por un mes	API de HTTP (un millón)	1 USD por mes	
	Un millón de llamadas a API recibidas para las API de HTTP por un mes	API de REST (un millón)	3,50 USD por mes	
	Un millón de mensajes por un mes	Almacenamiento en caché (opcional)		
	750 000 minutos de conexión para las API de WebSocket por un mes	0,5 GB	0,02 USD por mes	
	12 MESES GRATIS	1,6 GB	0,038 USD por mes	
		API de WebSocket		
		Transferencias de mensajes (un millón)	1 USD por mes	
	Minutos de conexión	0,25 USD por millón de minutos de conexión		

Mediante el análisis realizado en la tabla 5, detallando características técnicas y precios, se concluye que, si el sistema web no supera la capa gratuita de los servicios esto significa que no se paga nada, teniendo en cuenta que el servicio de AWS lambda y DynamoDB seguirán siendo gratis y por otro lado el servicio de Amazon API Gateway los 12 meses de uso son gratuitos, así mismo mientras no se supere el límite por mes durante los 12 meses los servicios seguirán siendo totalmente gratis.

3.3 Servicios serverless

3.3.1 AWS lambda

AWS Lambda es un servicio de computación serverless y basado en eventos lo que permite ejecutar código para practicar cualquier tipo de aplicación o servicio backend, esto sin la necesidad de aprovisionar o administrar servidores (administra automáticamente los recursos de computación). Puede activar Lambda desde más de 200 servicios de AWS y aplicaciones de software como servicio (SaaS), y solo paga por lo que utiliza.[17]

Tabla 6. Características de AWS lambda

CARACTERÍSTICAS	AWS LAMBDA
Lenguajes de Programación	Java Go PowerShell Node.js C# Python Ruby
Balaneo de Carga Automático:	Para funciones asociadas con servicios como API Gateway
Configurabilidad	SI
Concurrencia y aislamiento	Lambda siempre reserva una instancia separada para una única ejecución Cada ejecución tiene su grupo exclusivo de memoria y ciclos de CPU. Por tanto, el rendimiento es totalmente predecible y estable.
Integración HTTP	Amazon API Gateway escuchara el tráfico HTTP
Escalado automático	AWS Lambda es mejor para un escalamiento horizontal rápido y para manejar cargas de trabajo masivas, tanto para API web como para aplicaciones basadas en colas

CARACTERÍSTICAS	AWS LAMBDA
Inicio en frío	<p>En dos pasos:</p> <ul style="list-style-type: none"> ▪ Descarga el código ▪ Inicia un nuevo entorno de ejecución <p>No se le cobra por el tiempo que tarda Lambda en preparar la función</p> <p>La duración de un arranque en frío varía desde menos de 100 ms hasta más de 1 segundo</p>

3.3.2 DynamoDB

Amazon DynamoDB es una base de datos NoSQL de clave-valor serverless y completamente administrada que está diseñada para ejecutar aplicaciones de alto rendimiento a cualquier escala. DynamoDB ofrece seguridad integrada, copias de seguridad continuas, replicación automatizada en varias regiones, almacenamiento de caché en memoria y herramientas de importación y exportación de datos.[18]

Tabla 7. Características de DynamoDB

CARACTERÍSTICA	DESCRIPCIÓN
Tipo de base de datos	Base de datos NoSQL altamente escalable y completamente gestionada por AWS.
Modelo de Datos	Basado en clave-valor y documentos. Admite operaciones básicas como Obtener, Poner, Actualizar y Eliminar.
Escalabilidad Horizontal	DynamoDB es completamente escalable horizontalmente y puede manejar grandes volúmenes de tráfico.
Latencia Baja y Consistente	De milisegundos a cualquier escala y garantiza un rendimiento consistente.
Particionamiento automático de datos	Distribuye automáticamente los datos en Múltiples particiones para una escalabilidad sin problemas.
Backup y Restauración Automática	Permite realizar copias de seguridad automáticas y programadas, así como la restauración en cualquier punto del tiempo.

CARACTERÍSTICA	DESCRIPCIÓN
Gestión Automática de la Infraestructura	AWS DynamoDB gestiona automáticamente la infraestructura subyacente, eliminando la carga operativa.
Soporte de Transacciones	Proporciona soporte para transacciones, garantizando la atomicidad y la consistencia en operaciones grupales.
Integración con AWS Lambda	Puede integrarse fácilmente con AWS Lambda para realizar acciones basadas en eventos en tiempo real.
Control de Acceso y Seguridad	Proporciona opciones avanzadas para controlar el acceso y garantizar la seguridad de los datos almacenados.

3.3.3 Amazon API Gateway

Amazon API Gateway es un servicio completamente administrado que facilita a los desarrolladores en la creación, la publicación, el mantenimiento, el monitoreo y la protección de API a cualquier escala. Las API actúan como la "puerta de entrada" para que las aplicaciones accedan a los datos, la lógica empresarial o la funcionalidad de sus servicios de backend. API Gateway admite cargas de trabajo en contenedores y serverless, así como aplicaciones web.[19]

Tabla 8. Características Amazon API Gateway

CARACTERÍSTICA	DESCRIPCIÓN
Creación y Gestión de APIs	Permite la creación y gestión de APIs RESTful y WebSocket.
Escalabilidad Automática	Escala automáticamente para manejar cualquier cantidad de tráfico.
Integración con Servicios AWS	Se integra fácilmente con otros servicios de AWS, como Lambda, DynamoDB y S3.
Gestión de Tráfico y Control de Acceso	Permite controlar el tráfico mediante políticas de control de acceso y autenticación.
Documentación Automática	Genera automáticamente documentación para la API, facilitando su comprensión y uso.
Monitoreo y Análisis	Proporciona herramientas para monitorear el rendimiento de la API y analizar el tráfico.

CARACTERÍSTICA	DESCRIPCIÓN
Configuración de límites	Permite establecer límites en el número de solicitudes y la velocidad de solicitud.
Personalización de Respuestas de Errores	Permite personalizar las respuestas de error devueltas por la API.
Seguridad y Autenticación	Proporciona opciones de autenticación, como AWS Identity and Access Management (IAM) y Cognito.
Integración con Servicios de Autorización	Puede integrarse con servicios de autorización, como AWS Cognito y proveedores de identidad externos.
Soporte para WebSocket	Permite la creación de APIs WebSocket en tiempo real para aplicaciones interactivas.
Despliegue en etapas	Facilita el despliegue gradual de cambios en diferentes etapas del ciclo de vida de desarrollo.
Caché de resultados	Ofrece capacidades de caché para mejorar la velocidad de las respuestas y reducir la carga del servidor.
Integración con CloudWatch	Se integra con Amazon CloudWatch para el monitoreo y la creación de alertas basadas en métricas.
Personalización de Dominios	Permite personalizar dominios de API para integrarse con la marca de la aplicación.

3.4 Herramientas para desarrollo web

- Astro**
 Astro se destaca como una plataforma de desarrollo web que ofrece ventajas significativas para la creación de sitios rápidos y eficientes. Su enfoque minimalista y modular permite construir aplicaciones web ligeras, optimizadas para un rendimiento excepcional. Astro combina características estáticas y dinámicas, permitiendo la generación de contenido estático para una carga inicial rápida y la carga dinámica de contenido adicional cuando es necesario.[20]
- React**
 React, una librería de JavaScript desarrollada por Facebook se destaca por su flexibilidad y eficiencia en la creación de interfaces de usuario interactivas. Su enfoque en la virtual DOM y el modularidad permite construir componentes reutilizables, facilitando el desarrollo de aplicaciones escalables y mantenibles.[21]

- **Angular**

Angular, un framework completo desarrollado por Google, ofrece una estructura robusta y convenciones predefinidas para el desarrollo de aplicaciones web complejas. Su enfoque basado en componentes, junto con características como enrutamiento, gestión de estado y comunicación con el servidor, proporciona una solución integral. Aunque tiene una curva de aprendizaje mayor, Angular es ideal para proyectos empresariales que requieren una arquitectura organizada y herramientas integradas.[22]

Tabla 9. Comparación de frameworks front-end

CARACTERÍSTICAS	ASTRO	REACT	ANGULAR
Tamaño	Muy liviano y minimalista	Liviano, pero requiere gestión externa para algunas funcionalidades	Más pesado debido a su conjunto completo de herramientas
Rendimiento	Enfoque en velocidad y eficiencia	Alto rendimiento debido a la virtual DOM y renderizado selectivo	Menor rendimiento en comparación con React debido a su tamaño y complejidad
Flexibilidad	Alta flexibilidad para elegir herramientas y configuraciones	Ofrece libertad en la elección de herramientas y librerías	Mayor rigidez debido a su estructura y convenciones predefinidas
Curva de aprendizaje	Más rápida debido a su enfoque simplificado	Moderada, pero requiere comprensión de conceptos de React	Mayor debido a su estructura completa y características avanzadas
Escalabilidad	Se enfoca en aplicaciones pequeñas a medianas con flexibilidad	Escalable para proyectos de cualquier tamaño	Ideal para aplicaciones grandes y complejas que requieren estructura y organización
Comunidad y soporte	Comunidad en crecimiento con soporte limitado actualmente	Gran comunidad y amplia documentación	Gran comunidad y soporte de Google

Tras analizar los tres frameworks para desarrollo web, se concluye que tanto Astro, React y Angular cumplen con los requisitos para este proyecto. Sin embargo, se opta por trabajar con Astro por su curva de aprendizaje y velocidad en rendimiento.

3.5 Metodologías de desarrollo WEB

Las metodologías ágiles representan enfoques flexibles y colaborativos para la gestión de proyectos, especialmente en desarrollo de software. Estos métodos se centran en la adaptación a cambios constantes, promueven la comunicación estrecha entre los equipos y los clientes, entregan funcionalidades de forma incremental y priorizan la satisfacción del cliente mediante la entrega continua de valor. Su flexibilidad permite ajustes en cualquier etapa del proyecto para asegurar la eficacia y calidad del producto final.

Las principales metodologías para desarrollo web son:

3.5.1 Extreme Programming

XP es una metodología ágil de gestión de proyectos centrada en potenciar la velocidad la simplicidad con ciclos de desarrollo cortos. Esta metodología se centra en la prueba y error para el desarrollo de un producto de software funcional, permitiendo la participación del cliente en todo el proceso como condición fundamental para el resultado exitoso del proceso, promoviendo el trabajo en equipo e impulsando el buen clima laboral [23]

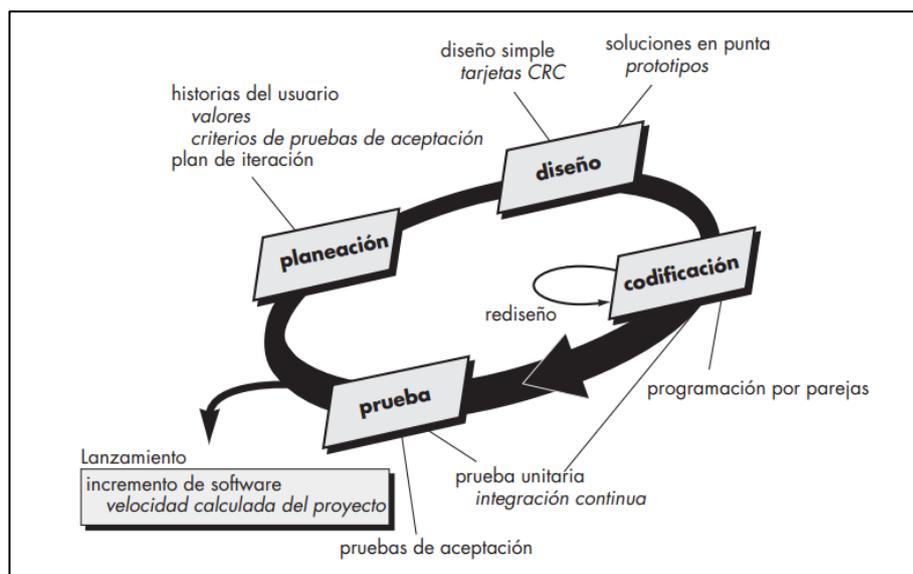


Figura 7. Flujo de proceso XP [24]

3.5.2 KANBAN

Kanban es una metodología visual de gestión y organización del trabajo.

- Divide el trabajo en bloques, escribe cada elemento en una tarjeta y ponlo en el muro.
- Utiliza columnas con nombre para ilustrar dónde está cada elemento en el flujo de trabajo.[25]

Kanban es ideal para equipos que buscan mejorar su productividad al optimizar la eficiencia del proceso de trabajo, visualizar el flujo de tareas y adaptarse rápidamente a cambios en las prioridades o requerimientos del proyecto.



Figura 8. Flujo de trabajo KANBAN. [25]

3.5.3 SCRUM

Es un método de desarrollo ágil de software interactiva e incremental y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis, se dice que es interactiva ya que se ejecuta en bloques temporales cortos y fijos (de no menos de dos semanas) que reciben el nombre de sprints y es incremental en tanto se obtienen funcionalidades del producto final al terminar cada iteración [26]

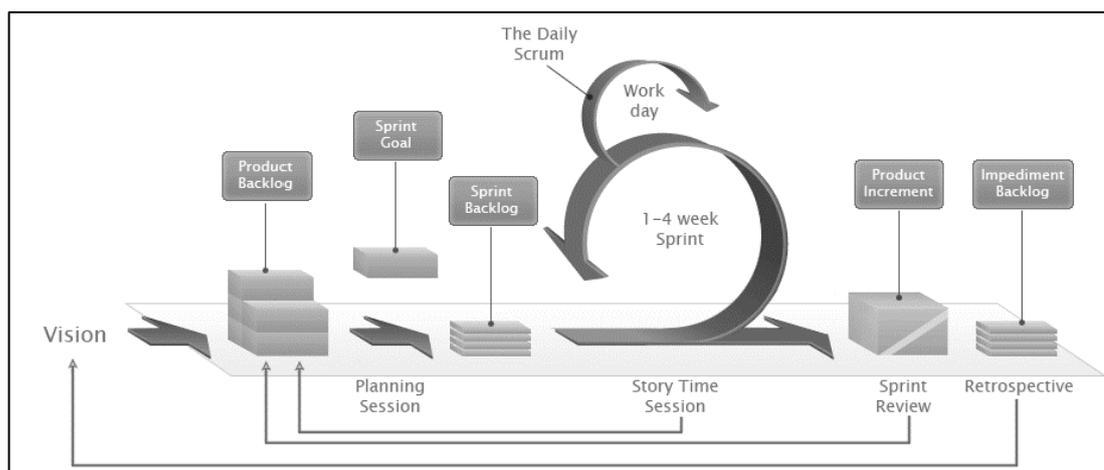


Figura 9. Flujo de proceso SCRUM [27]

Tabla 10. Comparación de metodologías

	Extreme Programming (XP)	KANBAN	SCRUM
Tipo de modelo	diciplina en prácticas de ingeniería	metodología basada en Lean	proceso - framework de management
Roles	PROGRAMADOR	Product Owner	Service request Manager
	CLIENTE	Scrum Master	Service Delivery Manager
	ENCARGADO DE PRUEBAS (TESTER)	Scrum Team	
	ENCARGADO DE SEGUIMIENTO (TRACKER)	Stakeholders	
	ENTRENADOR (COACH)		
	CONSULTOR		
	GESTOR (BIG BOSS):		
Enfoque	Entrega de software de alta calidad.	Gestión visual del flujo de trabajo.	Desarrollo iterativo e incremental.
Iteraciones	Desarrollo en iteraciones cortas.	No tiene iteraciones fijas.	Sprints con duración fija (2-4 semanas).
Principios clave	Comunicación, simplicidad, retroalimentación.	Visualización del flujo, límites de trabajo.	Transparencia, adaptabilidad, entrega incremental.
Practicar	Pair Programming, TDD, Integración Continua.	Limitación del trabajo en progreso, mejora continua.	Reuniones diarias (Daily Scrum), Sprint Planning.
Cliente	Muy involucrado, retroalimentación constante.	Colaboración con el equipo, cambios dinámicos.	Interacción con el Product Owner, feedback.

	Extreme Programming (XP)	KANBAN	SCRUM
Flexibilidad	Alta, se adapta a cambios y riesgos técnicos.	Alta, se ajusta a la capacidad del equipo.	Alta, permite ajustes durante los sprints.
Tamaño del proyecto	Pequeños y medianos.	Pequeños y medianos.	Pequeños, medianos y grandes.
Característica	método ágil radical	método simple: visualización limitación del WIP y medición	complemento de otros métodos
Complejidad	baja	media	Alta
Fases o etapas	<ul style="list-style-type: none"> • Planificación • Diseño • Codificación • Pruebas • Implementación 	<ul style="list-style-type: none"> • Visualización del Flujo de Trabajo • Establecimiento de Límites de Trabajo en Progreso (WIP - Work In Progress) • Gestión Visual • Monitorización y Mejora Continua • Adaptabilidad y Flexibilidad • Retrospectivas y Aprendizaje 	<ul style="list-style-type: none"> • Backlog del Producto. • Sprint Planning • Daily Scrum • Sprint Review • Sprint Retrospective • Entrega continua

Según el análisis efectuado en la Tabla 10 de la comparación entre las metodologías ágiles, se concluyó que XP se adecúa a las exigencias del proyecto. Esta metodología se encuentra diseñada para proyectos y equipos de trabajo más reducidos, permitiendo una comunicación directa con el cliente. Esta cercanía facilita la adaptación a cambios que puedan surgir en cualquier etapa del proyecto, al favorecer un desarrollo incremental y una comunicación fluida.

3.6 Desarrollo de la propuesta

En el desarrollo del proyecto, se ha optado por utilizar la metodología XP con el objetivo de mejorar la eficiencia en la entrega de las funcionalidades del sistema web. Esta metodología se divide en cinco etapas fundamentales: planificación, diseño, codificación, pruebas e implantación. A continuación, se detalla cada una de estas fases:

3.6.1 Fase I: Planificación

En la metodología Extreme Programming (XP), se enfatiza la elaboración de historias de usuario, las cuales representan funcionalidades específicas desde la perspectiva del usuario, con la finalidad de obtener los requisitos del sistema y los roles que tendrá cada uno.

3.6.2 Requisitos Funcionales del Sistema

Los requisitos funcionales definen las capacidades y acciones concretas que debe ofrecer el sistema web es decir todo lo que debe hacer el sistema de forma interna, sin incluir aspectos de interfaz o interacción, con el objetivo de cumplir con las expectativas del usuario y del negocio.[28]

A continuación, se especifican los requisitos funcionales que tendrá la aplicación web.

- **Sistema de seguridad y autenticación:** El sistema debe garantizar medidas sólidas de seguridad para proteger la información, utilizando autenticación con un correo y contraseña
- **Sistema de recuperación de contraseña:** Permite al usuario solicitar la recuperación de contraseña mediante un correo electrónico.
- **Gestión de usuarios:** El sistema debe tener administradores y usuarios cada uno con diferentes roles.
- **Creación de usuarios:** Solamente el administrador puede crear usuarios.
- **Gestión de inventario:** Se presentará un listado de los productos, de los cuales podrá agregar, modificar y eliminar los productos del inventario.
- **Generación de informes:** Generar informes periódicos o bajo demanda sobre el estado del inventario, compras y ventas.
- **Gestión de ventas:** Permite crear, almacenar e imprimir comprobante de pago con los datos detallados, además incluye la información de los productos, los datos del cliente y la fecha que se realizó la venta. También, se mostrará una lista del historial de ventas.
- **Gestión de compras:** Se mostrará una lista de las compras realizadas, de los cuales se podrá registrar, editar y eliminar compras.

- **Cerrar sesión:** Cada usuario podrá finalizar la conexión con el sistema para mantener la seguridad.

3.6.3 Roles Asignados en el proyecto

En la metodología XP se designan roles específicos a los miembros del equipo con el objetivo de promover la colaboración. En el contexto de este proyecto, se han seleccionado únicamente aquellos roles que mejor se ajustan al desarrollo.

Tabla 11. Definición de roles

ROL	NOMBRE	FUNCION
Programador	Paul Hidalgo	Encargado de realizar el análisis, la planificación, diseño, desarrollo y codificación, pruebas funcionales para la entrega del sistema, adicional del manual de usuario
StackHolder-Tester	Gerente y personal de la microempresa	Los interesados del sistema son los encargados de realizar las especificaciones de las reglas de negocio, pruebas y aprobación del sistema
Coach	Ing. leonardo Torres	Encargado de seguimiento del proyecto y revisión de avances de manera constante

3.6.4 Arquitectura de la Aplicación

El sistema está basado en arquitectura serverless el cual se usará los servicios que ofrece Amazon Web Service (AWS)

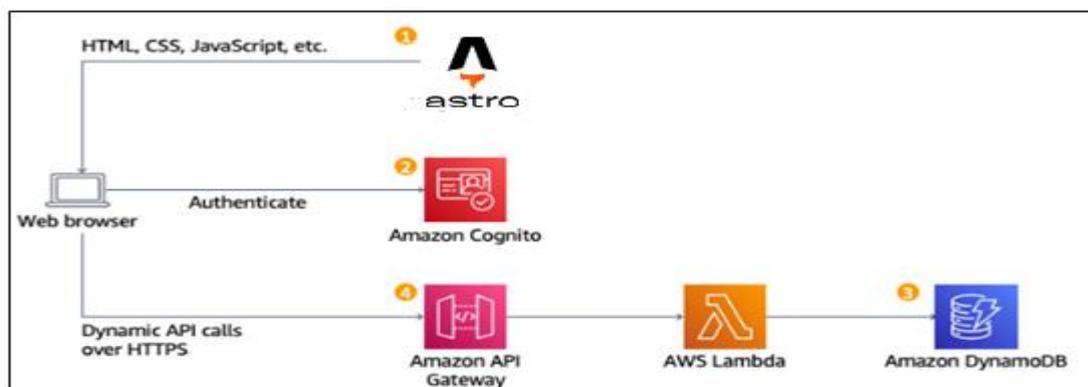


Figura 10. Arquitectura del proyecto

- **Astro:** Se utiliza para el desarrollo del front-end.
- **Amazon Cognito:** Agrega funciones de registro e inicio de sesión para los usuarios y controlar el acceso al sistema Web
- **Amazon API Gateway:** Gestiona todas las llamadas a API, también actúa como "puerta de entrada" para que las aplicaciones accedan a los datos, la lógica empresarial o la funcionalidad de sus servicios de backend.
- **AWS lambda:** se encarga de ejecutar el código como respuesta a eventos y administra de manera automática los recursos informáticos.
- **Amazon DynamoDB:** Se encarga de la gestión de los datos, es una base de datos NoSQL de clave-valor serverless.

3.6.5 Requerimientos de Hardware

- Procesador multicore, como un Intel Xeon o AMD Ryzen al menos 2.5 GHz
- Memoria (RAM) al menos 8 GB
- Almacenamiento (SSD o HDD) al menos 225GB
- Conexión a Internet

3.6.6 Requerimientos de Software

- Visual Studio Code
- AWS CLI (Command Line Interface)
- Serverless Framework
- Nodejs
- Astro
- Tailwind css
- GitHub

3.6.7 Historias de Usuario

Estas describen los requerimientos del cliente enfocadas a las funcionalidades que se implantara en el sistema Web, deben ser detalladas de forma entendible, fácil de comprender con el objetivo de obtener las necesidades del usuario final.

Tabla 12. Modelo de tabla para historia de usuario

HISTORIA DE USUARIO	
Número:	Usuario:
Nombre de la historia:	
Prioridad del negocio:	Riesgo de desarrollo:
Puntos estimados:	Iteración asignada:
Responsable:	
Descripción:	
Observaciones:	

A continuación, se detalla cada una de las especificaciones que posee las historias de usuario:

- **Número:** Identificador establecido para cada historia de usuario.
- **Usuario:** Individuo responsable de la historia de usuario.
- **Nombre Historia:** Determinación del título de la historia de usuario.
- **Prioridad en Negocio:** Medidor del nivel de necesidad del usuario (Alta, Media, Baja).
- **Riesgo en Desarrollo:** Medidor del nivel de riesgo que representa (Alta, Media, Baja).
- **Puntos Estimados:** Lapso (días) que fue implementado en el desarrollo de la actividad.
- **Iteración Asignada:** Iteración designada por el desarrollo o a la fase perteneciente.
- **Programador Responsable:** Persona responsable del desarrollo de la historia de usuario.
- **Descripción:** Comentario corto del cliente acerca de la actividad de la historia de usuario.
- **Observación:** Notas extras acerca de aclaraciones de procesos de la historia de usuario.

En las siguientes tablas se definieron las historias de usuario:

Tabla 13. Historia de usuario registro de usuarios

HISTORIA DE USUARIO	
Número: 01	Usuario: Administrador
Nombre de la historia: Inicio de sesión	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Responsable: Paul hidalgo	
Descripción: Los usuarios se autenticarán por medio del correo y contraseña y según el rol tendrán acceso a las funcionalidades del sistema.	
Observaciones: los usuarios deben estar registrados para iniciar sesión.	

Tabla 14. Historia de usuario inicio de sesión

HISTORIA DE USUARIO	
Número: 02	Usuario: Todos
Nombre de la historia: Registro de usuarios	
Prioridad del negocio: Medio	Riesgo de desarrollo: medio
Puntos estimados: 1	Iteración asignada: 1
Responsable: Paul hidalgo	
Descripción: Solo el administrador puede registrar nuevos usuarios.	
Observaciones: el usuario una vez registrado puede ingresar al sistema con el correo y contraseña.	

Tabla 15. Historia de usuario recuperación de contraseña

HISTORIA DE USUARIO	
Número: 03	Usuario: Todos
Nombre de la historia: Recuperación de contraseña	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 1
Responsable: Paul hidalgo	
Descripción: Los usuarios pueden recuperar la contraseña en caso de olvido o perdida por medio del correo electrónico	
Observaciones: solo los usuarios registrados pueden realizar la opción de recuperación de contraseña	

Tabla 16. Historia de usuario menú de opciones

HISTORIA DE USUARIO	
Número: 04	Usuario: todos
Nombre de la historia: Menú de opciones	
Prioridad del negocio: Medio	Riesgo de desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 1
Responsable: Paul hidalgo	
Descripción: Los usuarios pueden visualizar las opciones del menú principal	
Observaciones: los usuarios según el rol pueden visualizar las opciones del menú.	

Tabla 17. Historia de usuario registro de productos

HISTORIA DE USUARIO	
Número: 05	Usuario: Administrador
Nombre de la historia: Registro de productos	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 2
Responsable: Paul hidalgo	
Descripción: El administrador puede ingresar productos por medio de un formulario, el cual debe ingresar toda la información referente al producto.	
Observaciones: los productos en el inventario no se pueden repetir dos veces.	

Tabla 18. Historia de usuario información del producto

HISTORIA DE USUARIO	
Número: 06	Usuario: Administrador
Nombre de la historia: Información del producto	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 2
Responsable: Paul hidalgo	
Descripción: El administrador puede visualizar la información, editar y eliminar el producto.	
Observaciones: N/A	

Tabla 19. Historia de usuario lista de productos del inventario

HISTORIA DE USUARIO	
Número: 07	Usuario: Todos
Nombre de la historia: Lista de productos del inventario	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 2
Responsable: Paul hidalgo	
Descripción: Los usuarios pueden visualizar en una lista todos los productos existentes en el inventario con la opción de buscar y filtrar por categorías.	
Observaciones: N/A	

Tabla 20. Historia de usuario registro de compras

HISTORIA DE USUARIO	
Número: 08	Usuario: Administrador
Nombre de la historia: Registro de compras	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 3
Responsable: Paul hidalgo	
Descripción: El administrador registra cada compra por medio de un formulario donde se ingresa toda la información de la compra incluyendo el proveedor.	
Observaciones: El campo total de la compra es un campo calculado el administrador no puede ingresar manualmente.	

Tabla 21. Historia de usuario información de la compra

HISTORIA DE USUARIO	
Número: 09	Usuario: Administrador
Nombre de la historia: Información de la compra	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 3
Responsable: Paul hidalgo	
Descripción: El administrador puede visualizar la información de cada compra, además puede editar ciertos o eliminar todo el registro de la compra.	
Observaciones: Los campos principales o calculados no se pueden editar.	

Tabla 22. Historia de usuario historial de compras

HISTORIA DE USUARIO	
Número: 10	Usuario: Administrador
Nombre de la historia: Historial de compras	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 3
Responsable: Paul hidalgo	
Descripción: El administrador puede visualizar en una lista el historial de las compras ordenadas por la fecha, además tiene la opción de buscar en el historial de compras una inespecífico.	
Observaciones: Los campos principales o calculados no se pueden editar.	

Tabla 23. Historia de usuario registro de ventas

HISTORIA DE USUARIO	
Número: 11	Usuario: Todos
Nombre de la historia: Registro de ventas	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 4
Responsable: Paul hidalgo	
Descripción: Los usuarios pueden realizar ventas. Los usuarios pueden visualizar opciones de búsqueda para seleccionar el producto que se va a vender, además puede filtrar por categorías. El usuario Selecciona el producto y este se ingresa a una lista de productos seleccionados y cada producto tiene opciones de aumentar o disminuir la cantidad o de eliminar de la lista	
Observaciones: los productos que no tengan stock no podrán no podrán ser vendidos	

Tabla 24. Historia de usuario detalle de la venta

HISTORIA DE USUARIO	
Número: 12	Usuario: Todos
Nombre de la historia: Detalle de la venta	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 4
Responsable: Paul hidalgo	
Descripción: <ul style="list-style-type: none"> Los usuarios deben ingresar los datos del cliente para poder confirmar la venta. 	

<ul style="list-style-type: none"> • Si el cliente ya está registrado los datos se llenan automáticamente. • Los usuarios visualizan los detalles de la venta antes de confirmar la venta.
Observaciones: si existe errores con un producto puede regresar a la lista de productos seleccionados para corregir el error.

Tabla 25. Historia de usuario historial de ventas

HISTORIA DE USUARIO	
Número: 13	Usuario: Todos
Nombre de la historia: Historial de ventas	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 4
Responsable: Paul hidalgo	
Descripción: Los usuarios pueden visualizar el historial de las ventas ordenadas por la fecha. Cada venta tiene su información y la opción de imprimir el comprobante.	
Observaciones: las ventas solo son de lectura.	

Tabla 26. Historia de usuario informes

HISTORIA DE USUARIO	
Número: 14	Usuario: Administrador
Nombre de la historia: Informes	
Prioridad del negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 5
Responsable: Paul hidalgo	
Descripción: El administrador tiene la opción de visualizar los informes de las ventas y de las compras.	
Observaciones: las ventas solo son de lectura.	

Tabla 27. Historia de usuario finalizar sesión

HISTORIA DE USUARIO	
Número: 15	Usuario: Administrador
Nombre de la historia: Finalizar sesión	
Prioridad del negocio: Media	Riesgo de desarrollo: Media
Puntos estimados: 2	Iteración asignada: 5
Responsable: Paul hidalgo	

Descripción: El usuario tiene la opción de finalizar la sesión por motivos de seguridad.
Observaciones: N/A

3.6.8 Estimación de Historias de Usuario

Realizadas las historias de usuario se establece un tiempo máximo específico para cumplir con las actividades y requerimientos de la aplicación web.

Tabla 28. Estimación de historias de usuario

N°	Historial de usuario	tiempo estimado	
		Días	Horas
01	Inicio de sesión	2	5
02	Registro de usuarios	2	5
03	Recuperación de contraseña	4	5
04	Menú de opciones	2	3
05	Registro de productos	2	5
06	Información del producto	3	5
07	Lista de productos del inventario	2	5
08	Registro de compras	2	5
09	Información de la compra	3	5
10	Historial de compras	2	5
11	Registro de ventas	2	5
12	Detalle de la venta	3	5
13	Historial de ventas	2	5
14	Informes	3	5
15	Finalizar sesión	2	5
Tiempo total estimado		36	

3.6.9 Plan de Entrega

Después que se realizó la estimación de historias de usuarios, se elabora el cronograma, para las entregas del proyecto.

Tabla 29. Plan de entrega

N°	Historial de usuario	tiempo estimado		1	2	3	4
		Días	Horas				
01	Inicio de sesión	2	5	X			
02	Registro de usuarios	2	5	X			
03	Recuperación de contraseña	4	5	X			
04	Menú de opciones	2	3	X			
05	Registro de productos	2	5		x		
06	Información del producto	3	5		X		
07	Lista de productos del inventario	2	5		X		
08	Registro de compras	2	5		X		
09	Información de la compra	3	5			X	
10	Historial de compras	2	5			X	
11	Registro de ventas	2	5			X	
12	Detalle de la venta	3	5			X	
13	Historial de ventas	2	5				X
14	Informes	3	5				X
15	Finalizar sesión	2	5				X
Tiempo total estimado		36					

3.7 Fase II: Diseño

3.7.1 Plan de iteraciones

Una vez que se ha completado el desarrollo de las historias de usuario y se ha estimado el tiempo necesario para cada una, comienza el proceso de iteraciones para cumplir con los requisitos del sistema. Durante estas iteraciones, se lleva a cabo la implementación progresiva de las historias, paso a paso, para lograr la realización completa de los objetivos planteados en el sistema.

- **Iteración 1**

Tabla 30. Historia de usuario Iteración 1

Código	Historia de usuario	Prioridad	Riesgo
01	Inicio de sesión	Alto	Alto
02	Registro de usuarios	Medio	Medio
03	Recuperación de contraseña	Alto	Alto
04	Menú de opciones	Medio	Medio

01. Inicio de sesión

Solo el administrador puede registrar y asignar roles a los usuarios.

Tabla 31. Actividades de la historia de usuario 01

Código de historia:02	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none"> • Diseño de la interfaz de Login del sistema. • Comprueba las credenciales del usuario en AWS Cognito y retorna si el usuario existe 	

02. Registro de usuarios

Los usuarios que estén registrados en deben ingresar correo y contraseña para ingresar a la aplicación web.

Tabla 32. Actividades de la historia de usuario 02

Código de historia:01	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none"> • Diseño de la interfaz del registro de usuarios • Ingreso de la información al servicio de AWS Cognito 	

03. Recuperación de contraseña

Los usuarios que olviden la contraseña tienen la opción de recuperar por medio del correo la contraseña.

Tabla 33. Actividades de la historia de usuario 03

Código de historia:03	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none"> • Diseño de la interfaz. • AWS Cognito envía un código al correo electrónico del usuario para cambiar de contraseña 	

04. Menú de opciones

Cada usuario dependiendo el rol tendrán diferentes opciones de menú.

Tabla 34. Actividades de la historia de usuario 04

Código de historia:04	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none"> • Diseño de la interfaz de las opciones de menú para administrador y usuario ventas 	

- **Iteración 2**

Tabla 35. Historia de usuario Iteración 2

Código	Historia de usuario	Prioridad	Riesgo
05	Registro de productos	Alto	Alto
06	Información del producto	Alto	Alto
07	Lista de productos del inventario	Medio	Medio

05. Registro de productos

El administrador podrá agregar nuevos productos al inventario.

Tabla 36. Actividades de la historia de usuario 05

Código de historia:05	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none"> • Diseño de la interfaz para agregar nuevo producto • Guardar producto en la tabla de inventario. 	

06. Información del producto

El administrador puede ver la información del producto, también puede editar cierta información o eliminar completamente el registro de la lista.

Tabla 37. Actividades de la historia de usuario 06

Código de historia:06	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none">• Diseño de la interfaz de la información del producto• Editar información• Eliminar registro• Guardar cambios	

07. Lista de productos del inventario

Los usuarios pueden visualizar la lista de productos del inventario y pueden buscar y filtrar por categorías los productos.

Tabla 38. Actividades de la historia de usuario 07

Código de historia:07	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none">• Diseño de la interfaz• Listar los productos de la tabla inventarios• Ordenar por fecha	

- **Iteración 3**

Tabla 39. Historias de usuario Iteración 3

Código	Historia de usuario	Prioridad	Riesgo
08	Registro de compras	Alto	Alto
09	Información de la compra	Alto	Alto
10	Historial de compras	Medio	Medio

08. Registro de compras

El administrador puede ingresar nuevas compras y automáticamente el producto se añade o actualiza en el inventario.

Tabla 40. Actividades de la historia de usuario 08

Código de historia:08	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none"> • Diseño de la interfaz • Formulario de nueva compra • Actualizar inventario • Guardar registro de la compra 	

09. Información de la compra

El administrador puede editar cierta información de la compra o eliminar totalmente el registro.

Tabla 41. Actividades de la historia de usuario 09

Código de historia:09	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none"> • Diseño de la interfaz • Información de la compra • Actualizar la información de la compra y del inventario • Guardar cambios 	

10. Historial de compras

El administrador podrá visualizar la lista del historial de compra.

Tabla 42. Actividades de la historia de usuario 10

Código de historia:10	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none"> • Diseño de la interfaz • Búsqueda en la lista del historial • Ordenar por fechas 	

- **Iteración 4**

Tabla 43. Historia de usuario Iteración 4

Código	Historia de usuario	Prioridad	Riesgo
11	Registro de ventas	Alto	Alto
12	Detalle de la venta	Alto	Alto
13	Historial de ventas	Alto	Alto

11. Registro de ventas

Los usuarios pueden realizar una nueva venta de los productos del inventario.

Tabla 44. Actividades de la historia de usuario 11

Código de historia: 11	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none"> • Diseño de la interfaz • Seleccionar productos disponibles • Filtrar por categorías • Elegir cantidad • Agregar a la lista de productos seleccionados • Editar cantidades de cada producto o eliminar de la lista 	

12. Detalle de la venta

El usuario visualiza el detalle antes de confirmar la venta y debe ingresar los datos del cliente si ya está registrado se autocompleta los campos, caso contrario se registra un nuevo cliente

Tabla 45. Actividades de la historia de usuario 12

Código de historia: 12	Responsible: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none"> • Diseño de la interfaz • Ingreso de datos del cliente • Registrar nuevo cliente • Auto completado de los datos del cliente • Listar productos seleccionados • Cálculo del total de la venta • Guardar registro de la venta 	

13. Historial de ventas

Los usuarios pueden visualizar una lista con historial de las ventas, Estas estarán ordenadas por fecha.

Tabla 46. Actividades de la historia de usuario 13

Código de historia:13	Responsable: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none">• Diseño de la interfaz• Búsqueda de las ventas	

- **Iteración 5**

Tabla 47. Historia de usuario Iteración 4

Código	Historia de usuario	Prioridad	Riesgo
14	Informes	Medio	Medio
15	Cerrar sesión	Medio	Medio

14. Informes

El administrador puede ver el informe de las ventas o de compras conforme a la fecha que necesite.

Tabla 48. Actividades de la historia de usuario 14

Código de historia:14	Responsable: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none">• Diseño de la interfaz• Seleccionar fecha para el informe	

15. Cerrar sesión

Los usuarios que tengan iniciada sesión podrán cerrar sesión.

Tabla 49. Actividades de la historia de usuario 15

Código de historia:15	Responsable: Paul Hidalgo
Actividades:	
<ul style="list-style-type: none">• Diseño de la interfaz	

3.7.2 Tarjetas CRC

Las tarjetas CRC (Clases, Responsabilidades y Colaboradores) ayuda a tener una mejor orientación para el diseño de la aplicación, además sirven como guía durante la implementación, asegurando que cada componente tenga responsabilidades claras y bien definidas.

Para el desarrollo del proyecto se ha realizado una tarjeta CRC para cada historia de usuario.

Tabla 50. Tarjeta CRC – Inicio de sesión

Inicio de sesión	
Responsabilidades	Colaboradores
Validar Credenciales. Interacción con el Usuario. Comunicación con el Backend. Gestionar Estado.	Servicio de Autenticación Componente de Validación. Redirección.
Observaciones: Los usuarios deben estar registrados y validados para poder ingresar al sistema.	

Tabla 51. Tarjeta CRC – Registro de usuarios

Registro de usuarios	
Responsabilidades	Colaboradores
Validar Datos. Interacción con el Usuario. Comunicación con el Backend. Gestionar Estado.	Servicio de Autenticación. Componente de Validación Componente de Notificación.
Observaciones: Solo el administrador puede registrar a un nuevo usuario.	

Tabla 52. Tarjeta CRC – Recuperación de contraseña

Recuperación de contraseña	
Responsabilidades	Colaboradores
Solicitud de Restablecimiento. Validar Información de Recuperación. Comunicación con el Backend. Gestionar Estado.	Servicio de Restablecimiento de Contraseña. Validación. Notificación.
Observaciones: Los usuarios solo pueden recuperar la contraseña el correo electrónico que se registraron.	

Tabla 53. Tarjeta CRC – Menú de Opción

Menú de opciones	
Responsabilidades	Colaboradores
Mostrar Opciones. Gestionar Navegación. Personalización del Menú. Gestionar Estado.	Componente de Navegación. Servicio de Autenticación/Autorización.
Observaciones: Solo el administrador tiene acceso total de todas las opciones del menú principal.	

Tabla 54. Tarjeta CRC – Registro de productos

Registro de productos	
Responsabilidades	Colaboradores
Ingreso de Datos. Validación de Datos. Comunicación con el Backend. Gestión de Estado.	Servicio de Almacenamiento de Datos. Validación. Notificación.
Observaciones: Los productos no se pueden repetir.	

Tabla 55. Tarjeta CRC – Información del producto

Información del producto	
Responsabilidades	Colaboradores
Leer información del Producto. Actualizar Producto. Eliminar Producto. Gestionar Estado.	Servicio de Almacenamiento de Datos. Detalle de Producto. Notificación.
Observaciones: Solo ciertos campos se pueden modificar.	

Tabla 56. Tarjeta CRC – Lista de productos del inventario

Lista de productos del inventario	
Responsabilidades	Colaboradores
Mostrar Lista. Filtrar y Ordenar. Buscar en la lista. Interacción con Productos. Gestionar Estado.	Servicio de Datos del Inventario. Detalle de Producto. Filtrado, búsqueda y Ordenación.
Observaciones: se filtran por categorías.	

Tabla 57. Tarjeta CRC – Registro de compras

Registro de compras	
Responsabilidades	Colaboradores
Ingreso de Datos. Validación de Datos. Comunicación con el Backend. Gestión de Estado.	Servicio de Almacenamiento de Datos. Validación. Notificación.
Observaciones: El total se calcula con la cantidad y el precio de compra del producto.	

Tabla 58. Tarjeta CRC – Información de la compra

Información de la compra	
Responsabilidades	Colaboradores
Leer información de Compra. Actualizar Compra. Eliminar Compra. Gestionar Estado.	Servicio de Almacenamiento de Datos. Detalle de Compra. Notificación.
Observaciones: Solo ciertos campos se pueden modificar	

Tabla 59. Tarjeta CRC – Historial de compras

Historial de compras	
Responsabilidades	Colaboradores
Mostrar Historial. Ordenar. Buscar. Visualización Detallada.	Servicio de Datos de Compras. Detalle de Compra. Filtrado, búsqueda y Ordenación.
Observaciones: N/A	

Tabla 60. Tarjeta CRC – Registro de ventas

Registro de ventas	
Responsabilidades	Colaboradores
Comunicación con el Backend. Filtrado. Seleccionar Productos. Gestión de Estado.	Servicio de Datos de inventarios. Validación. Notificación.
Observaciones: Los productos se filtran por categorías y no se puede seleccionar la cantidad mayor al del inventario.	

Tabla 61. Tarjeta CRC – Detalle de la venta

Detalle de la venta	
Responsabilidades	Colaboradores
Mostrar Detalles de Venta. Visualización de Elementos. Ingreso de Datos del Cliente. Presentar el formato del comprobante en PDF. Gestión de Estado.	Servicio de Almacenamiento de Datos. Visualización de Productos. Validación. Notificación.
Observaciones: Los datos de los clientes registrados se auto completa	

Tabla 62. Tarjeta CRC – Historial de ventas

Historial de ventas	
Responsabilidades	Colaboradores
Mostrar Historial. Ordenar y buscar. Visualización Detallada.	Servicio de Datos de Ventas. Detalle de Venta. Ordenación y búsqueda.
Observaciones: se ordenan por fecha de ingreso.	

Tabla 63. Tarjeta CRC – Informes

Informes	
Responsabilidades	Colaboradores
Generar Informes. Presentación Visual. Personalización.	Servicio de Datos.
Observaciones: la personalización es por fechas.	

Tabla 64. Tarjeta CRC – Cerrar sesión

Cerrar sesión	
Responsabilidades	Colaboradores
Cerrar Sesión del Usuario. Limpiar Datos de Sesión. Redirección. Gestión de Estado.	Servicio de Autenticación. Redirección. Servicio de Almacenamiento Loca.
Observaciones: N/A	

3.7.3 Diseño de la base de datos

Se utiliza **DynamoDB** como base de datos ya que es parte del ecosistema serverless, es un servicio de base de datos NoSQL, es decir, que no se basa en un esquema fijo y predefinido, como lo hace una base de datos relacional. En lugar de eso se basa en **tablas, elementos y atributos**, lo que permite una estructura flexible.

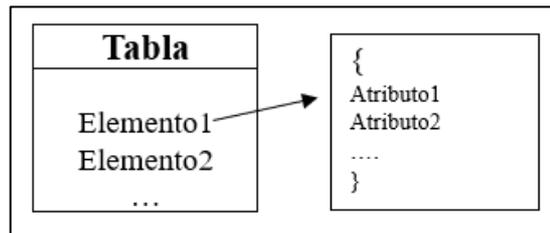


Figura 11. Estructura de DynamoDB

En el presente proyecto cuenta con ocho tablas de DynamoDB:

Tablas (8) Información			
<input type="text" value="Buscar tablas por su nombre"/>			
<input type="checkbox"/>	Nombre ▲	Estado	Clave de partición
<input type="checkbox"/>	clientestable	✓ Activo	pk (S)
<input type="checkbox"/>	comprastable	✓ Activo	pk (S)
<input type="checkbox"/>	inventario_mp	✓ Activo	pk (S)
<input type="checkbox"/>	numfactable	✓ Activo	pk (S)
<input type="checkbox"/>	provedorestable	✓ Activo	pk (S)
<input type="checkbox"/>	registrotable	✓ Activo	pk (S)
<input type="checkbox"/>	usertable	✓ Activo	pk (S)
<input type="checkbox"/>	ventastable	✓ Activo	pk (S)

Figura 12. Tablas de DynamoDB para el proyecto

Atributos de los elementos de cada tabla:

- **Tabla “clientestable”:**

Elemento:

```
{  
  "pk": "5923d0bb-cb3e-49ce-ba88-ced330a98cc3",  
  "cedula": "1804261905",  
  "direccion": "Ambato",  
  "email": "jesy27pao@hotmail.com",  
  "name": "Paola Ramos",  
  "telefono": "0998201574"  
}
```

- **Tabla “comprastable”:**

Elemento:

```
{  
  "pk": "e592a67f-3299-44cf-883d-8ae9b6b20937",  
  "cantidad": "12",  
  "categoria": "masa pan",  
  "codigo": "mspch006",  
  "descripcion": "pareja de cholitas sentadas",  
  "direccion": "Ambato",  
  "email": "pedro@gmail.com",  
  "fecha": "14/01/2024 20:33",  
  "nombre": "cholitas sentadas",  
  "preciocompra": "1.75",  
  "precioventa": "2.00",  
  "proveedor": "Pedro Sanches",  
  "telefono": "0986857456",  
  "total": "21.00"  
}
```

- **Tabla “inventario_mp”:**

Elemento:

```
{  
  "pk": "825998b0-01ab-4d54-a091-1d2c3034ce30",  
  "cantidad": 5,  
  "categoria": "masa pan",  
  "codigo": "mspva001",  
  "descripcion": "porta encienso figura vaquita",  
  "fecha": "14/01/2024 19:25",  
  "nombre": "vaquita",  
  "origen": "produccion",  
  "preciocompra": "0.50",  
  "precioventa": "1.25"  
}
```

- **Tabla “numfacttable”:**

Elemento:

```
{
  "pk": {
    "S": "zomi2024sistemcontablev1"
  },
  "numero": {
    "N": "112"
  }
}
```

- **Tabla “provedorestable”:**

Elemento:

```
{
  "pk": "062039f0-44f8-4488-96d8-cad262e16352",
  "direccion": "Ambato",
  "email": "pedro@gmail.com",
  "proveedor": "Pedro Sanches",
  "telefono": "0986857456"
}
```

- **Tabla “registrotable”:**

Elemento:

```
{
  "pk": "9d1eed7d-84a5-452b-b268-8f771564c50c",
  "cantidad": "12",
  "codigo": "mspch006",
  "fecha": "14/01/2024 20:33",
  "nombre": "cholitas sentadas",
  "origen": "compras",
  "precio": "2.00",
  "tipo": "ingreso"
}
```

- **Tabla “usertable”:**

Elemento:

```
{
  "pk": "fbbfca0f-8573-4fdc-85a8-3c27aec829f5",
  "cargo": "Ventas",
  "correo": "juperez883@gmail.com",
  "pass": "ljkh.8i9e9d6ilg5be0ljk12",
  "telefono": "0987788750",
  "usuario": "juan"
}
```

- **Tabla “ventastable”:**

Elemento:

```
{
  "0": {
    "cant": 1,
    "cod": "mspva001",
    "desc": "porta encienso figura vaquita",
    "id": "825998b0-01ab-4d54-a091-1d2c3034ce30",
    "nomb": "vaquita",
    "prec": "1.25$",
    "subt": 1.25
  },
  "1": {
    "cedula": "1804261905",
    "cliente": "Paola Ramos",
    "direccion": "Ambato",
    "email": "jesy27pao@hotmail.com",
    "fecha": "14/01/2024 19:29",
    "numfact": 105,
    "telefono": "0998201574",
    "total": "1.25"
  },
  "pk": "36620847-9dd6-4e69-add6-6b325edaf310"
}
```

3.7.4 Diseño de Iteraciones

Para el diseño se utilizó mockups que son representaciones simples de la interfaz, estas fueron realizadas en MockFlow.

A continuación, se presenta las interfaces de cada historia de usuario.

- **Inicio de sesión**

Para ingresar al sistema se necesita correo y contraseña que están registrados en AWS Cognito, además permite recuperar la contraseña.

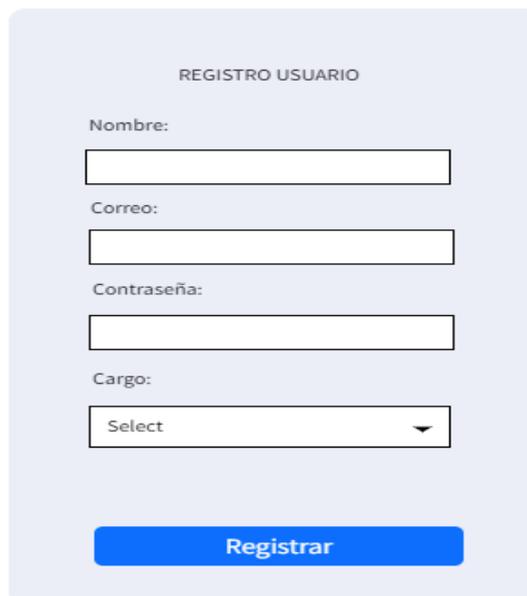


The image shows a login form titled "ZOMI". At the top, there is a rectangular box with a diagonal cross (X) inside. Below this, the text "ZOMI" is centered. The form contains two input fields: "Correo:" followed by a text box, and "Contraeña:" followed by a text box. Below the password field is a blue link that says "Recuperar Contraseña?". At the bottom of the form is a blue button with the text "Ingresar" in white.

Figura 13. Inicio de sesión.

- **Registro de usuarios**

Solo el administrador puede registrar usuarios nuevos para que puedan ingresar al sistema cada usuario debe validar su correo.

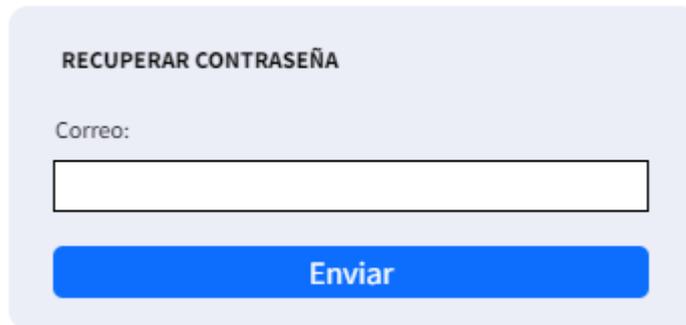


The image shows a user registration form titled "REGISTRO USUARIO". It contains four input fields: "Nombre:" followed by a text box, "Correo:" followed by a text box, "Contraeña:" followed by a text box, and "Cargo:" followed by a dropdown menu with "Select" and a downward arrow. At the bottom of the form is a blue button with the text "Registrar" in white.

Figura 14 Registro de usuario.

- **Recuperar contraseña**

Los usuarios registrados se les permite recuperar contraseña por medio del correo electrónico con el que se registraron.



RECUPERAR CONTRASEÑA

Correo:

Enviar

Figura 15. Recuperar contraseña.

- **Menú de opciones**

Los usuarios tienen diferentes roles y conforme al rol aparece las opciones de menú solo el rol de administrador puede ver todas las opciones del menú.

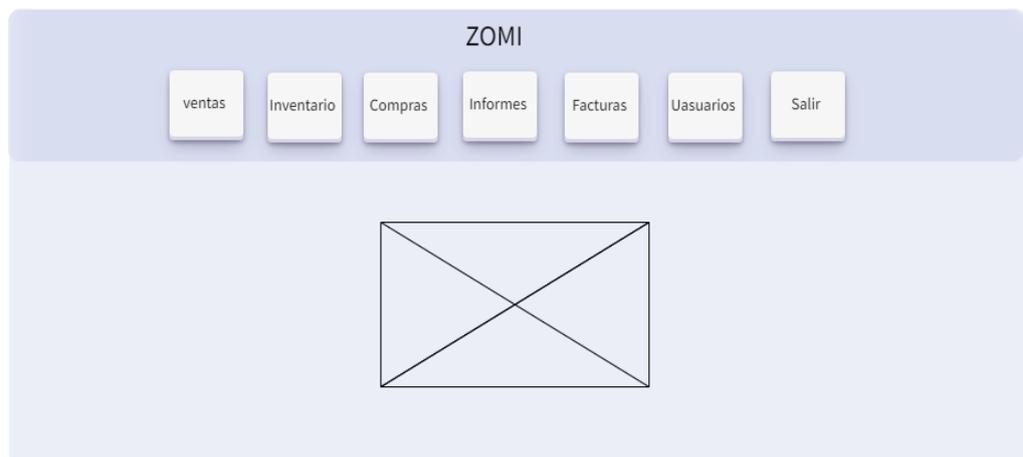


Figura 16. Menú principal

- **Registro de productos**

El administrador puede ingresar nuevos productos al inventario por medio de un formulario, los productos ingresados no se pueden repetir dos veces para evitar inconsistencias en el inventario.

Datos Producto

Codigo:

Categoria:

Nombre:

Precio Compra:

Cantidad:

Precio Venta:

Descripcion:

Agregar

Figura 17. Registro de productos.

- **Información del producto**

Cada producto tiene su propia información la cual sirve para visualizar, editar o eliminar en caso de ser necesario.

Datos Producto

Codigo:

Fecha Ingreso:

Nombre:

Cantidad Actual:

Nuevo cantidad Ingreso:

Categoria:

Tipo Ingreso:

Precio Venta:

Precio Compra:

Proveedor:

Descripcion:

Eliminar **Guardar**

Figura 18. Información del producto.

- **Lista de productos del inventario**

Los usuarios pueden visualizar la lista de los productos que existe en el inventario, además tienen la opción de filtrar por categorías y la opción de búsqueda.

The screenshot shows the ZOMI application interface. At the top, there is a navigation bar with buttons for 'ventas', 'Inventario' (highlighted in blue), 'Compras', 'Informes', 'Facturas', 'Usuarios', and 'Salir'. Below the navigation bar, there is a search and filter section with a 'Categoría:' dropdown menu set to 'Select' and a search input field with a magnifying glass icon. The main content is a table with the following data:

Fecha	Producto	Categoría
08/01/2024	jirafa	masa pan
01/01/2024	vaquita	masa pan
05/01/2024	sapito	masa pan

Figura 19. Lista de productos del inventario

- **Registro de compras**

El administrador puede ingresar las compras mediante un formulario donde incluye la información del producto y la del proveedor.

The screenshot shows the 'Datos Compra' registration form. It includes the following fields and controls:

- 'Elegir proveedor' dropdown menu
- 'Nuevo Proveedor' button
- 'Código:' text input field
- 'Producto' text input field
- 'Cantidad:' text input field
- 'Descripción:' text input field
- 'Categoría:' text input field
- 'Precio Compra:' text input field
- 'Precio Venta:' text input field
- 'Total:' text input field
- 'Agregar' button (highlighted in blue)

Figura 20. Registro de compras.

- **Información de la compra**

El administrador puede visualizar, editar ciertos campos o eliminar completamente el registro de la compra.



Formulario de Información Compra con los siguientes campos:

Código:	Producto:
<input type="text"/>	<input type="text"/>
Cantidad:	Categoría:
<input type="text"/>	<input type="text"/>
Precio Venta:	Precio Compra:
<input type="text"/>	<input type="text"/>
Descripción: <input type="text"/>	
Proveedor:	Telefono:
<input type="text"/>	<input type="text"/>
Dirección:	Correo:
<input type="text"/>	<input type="text"/>
Fecha:	Total:
<input type="text"/>	<input type="text"/>

Botones: **Eliminar** (rojo) y **Guardar** (azul).

Figura 21. Información de la compra.

- **Historial de compras**

El administrador puede visualizar la lista de todas las compras registradas en el sistema, con la opción de filtrar o buscar en la lista y las compras están ordenadas por fechas

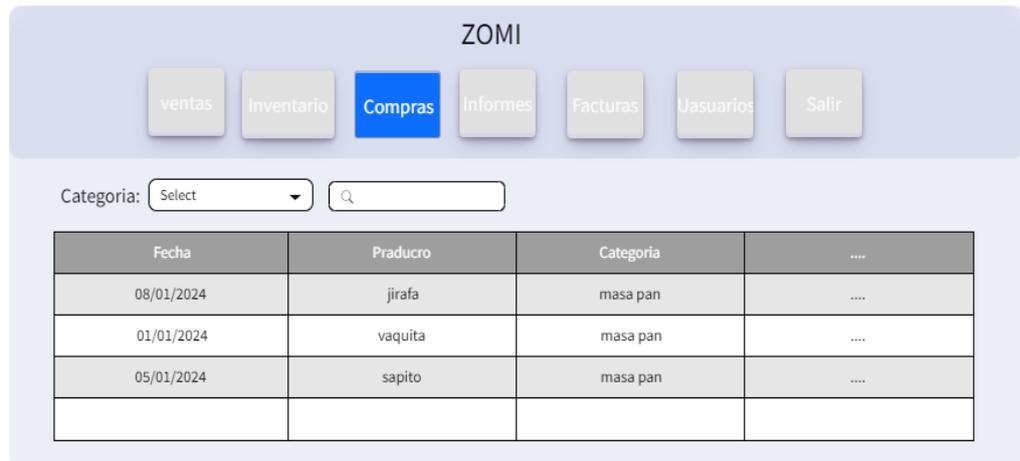


Figura 22. Historial de compras.

- **Registro de ventas**

Todos los usuarios registrados pueden realizar ventas, de los productos disponibles y se pueden filtrar por categorías, seleccionar la cantidad y agregar a una lista de productos seleccionados. Cada producto tiene la opción de aumentar, disminuir la cantidad o eliminar de la lista.

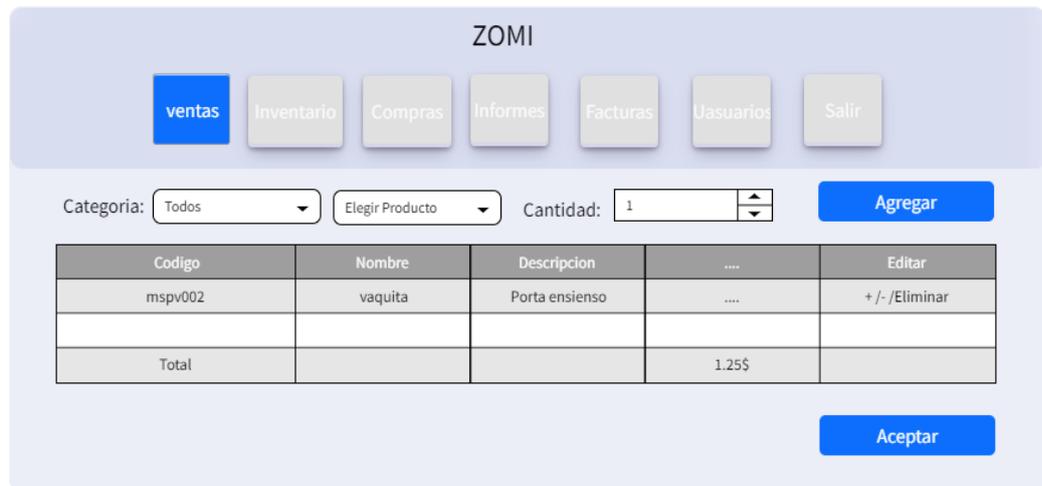


Figura 23. Registro de ventas.

- **Detalle de la venta**

Antes de confirmar la venta se visualiza el detalle de la venta, donde se ingresa los datos del cliente y verificar que los datos estén correctos.

Confirmar Venta

Cedula: Nombres:

Telefono: Correo:

Direccion:

Codigo	Descripcion	Subtotal
mospv002	Porta ensienso	1.25
Total			1.25\$

[Aceptar](#)

Figura 24. Detalle de venta.

- **Historial de ventas**

Los usuarios pueden visualizar en una lista todas las ventas realizadas por medio del sistema, las ventas están ordenadas por fechas y tiene la opción de búsqueda.

ZOMI

ventas
inventario
compras
informes
Facturas
usuarios
Salir

Fecha	Cliente	Cantidad	total
08/01/2024	Cliente	2	2.50
....

Figura 25. Historial de las ventas.

- **Informes**

El administrador tiene la opción de visualizar los informes de ventas y de compras, con la opción de elegir la fecha.

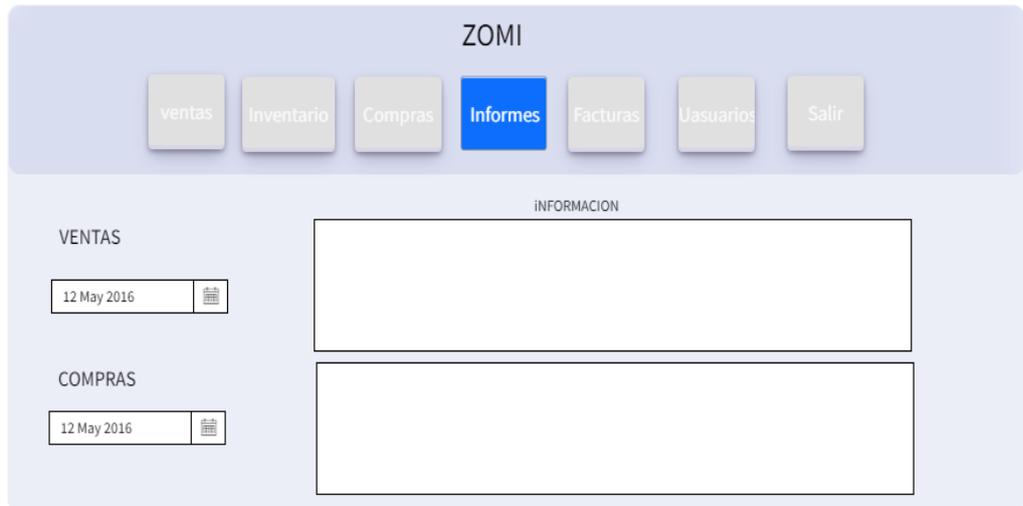


Figura 26. Informes.

- **Cerrar sesión**

Los usuarios que ingresan al sistema tienen la opción de cerrar sesión por motivos de seguridad.



Figura 27. Cerrar sesión.

3.8 Fase III: Codificación

En la fase de codificación para el back-end empezamos con la instalación de AWS CLI (Command Line Interface) en la máquina y se configura las Acces key, la región y el formato de salida en la terminal o línea de comandos, utilizando el siguiente comando: “aws configure”.

```
PS C:\Users\pauls\Desktop\serverless\CursoServerless> aws configure
AWS Access Key ID [*****Q4WI]:
AWS Secret Access Key [*****v1HE]:
Default region name [us-east-1]:
Default output format [json]:
PS C:\Users\pauls\Desktop\serverless\CursoServerless>
```

Figura 28. Configuración de AWS CLI

Las claves son del usuario que se crea en la plataforma de aws con el servicio de IAM.

3.8.1 Codificación de archivo serverless.yml

Finalizada la configuración inicia la codificación del archivo serverless.yml lo que es IaC (infraestructura como código), contiene toda la configuración de los recursos de la nube.

```
serverless.yml
1  service: crud-serverlessZomi
2
3  provider:
4    name: aws
5    runtime: nodejs14.x
6    iam:
7      role:
8        statements:
9          - Effect: Allow
10         Action: 'dynamodb:*'
11         Resource:
12           - arn:aws:dynamodb:us-east-1:490513585458:table/usertable
13           - arn:aws:dynamodb:us-east-1:490513585458:table/comprastable
14           - arn:aws:dynamodb:us-east-1:490513585458:table/ventastable
15           - arn:aws:dynamodb:us-east-1:490513585458:table/inventario_mp
16           - arn:aws:dynamodb:us-east-1:490513585458:table/registrotable
17
18  package:
19    individually: true
20    patterns:
21      - "!dynamodb"
22      - "!node_modules"
23
24  functions:
25    getallUsers:
26      handler: getallUsers/handler.getallUsers
27      events:
28        - http:
29          path: usersall
30          method: GET
```

Figura 29. Codificación del archivo serverless.yml

3.8.2 Codificación de las funciones lambdas

Cada función lambda cumple con un solo propósito por ejemplo la de insertar, actualizar, obtener o eliminar datos de las tablas de DynamoDB,

```
createVentas > JS handler.js > createVentas
1  const aws = require("aws-sdk")
2  const { randomUUID } = require("crypto")
3  let dynamoDBClientsParams = {}
4  const dynamodb = new aws.DynamoDB.DocumentClient(dynamoDBClientsParams)
5  const createVentas = async (event, context) => {
6      const id = randomUUID();
7      let userBody = JSON.parse(event.body)
8      userBody.pk = id
9
10     var params = {
11         TableName: 'ventastable',
12         Item: userBody
13     };
14
15     return dynamodb.put(params).promise().then(res => {
16         console.log(res)
17         return {
18             "statusCode": 200,
19             headers: {
20                 "Access-Control-Allow-Headers": "Content-Type",
21                 "Access-Control-Allow-Origin": "*",
22                 "Access-Control-Allow-Methods": "OPTIONS,POST,GET"
23             },
24             "body": JSON.stringify({ 'venta': params.Item })
25         }
26     })
27 }
28
29 module.exports = {
30     createVentas
31 }
32
```

Figura 30. Función lambda para insertar datos.

```
updateInvMs > JS handler.js > aws
6  const body = JSON.parse(event.body)
7
8  var params = {
9      TableName: 'inventario_mp',
10     Key: { pk: 'producId' },
11     UpdateExpression: 'set #nombre = :nombre, #codigo = :codigo, #cantidad=:cantidad, #categoria=:categoria,
12     #precioventa=:precioventa, #preciocompra=:preciocompra, #descripcion = :descripcion, #fecha = :fecha',
13     ExpressionAttributeNames: { '#nombre' : 'nombre', '#codigo': 'codigo', '#cantidad' : 'cantidad', '#categoria': 'categoria',
14     '#precioventa' : 'precioventa', '#preciocompra': 'preciocompra', '#descripcion' : 'descripcion', '#fecha' : 'fecha' },
15     ExpressionAttributeValues: { ':nombre' : body.nombre, ':codigo': body.codigo, ':cantidad' : body.cantidad, ':categoria':
16     body.categoria, ':precioventa' : body.precioventa, ':preciocompra': body.preciocompra, ':descripcion' : body.descripcion,
17     ':fecha' : body.fecha },
18     ReturnValues: 'ALL_NEW'
19 };
20
21 return dynamodb.update(params).promise().then (res => {
22     console.log(res)
23     return {
24         "statusCode": 200,
25         headers: {
26             "Access-Control-Allow-Headers" : "Content-Type",
27             "Access-Control-Allow-Origin": "*",
28             "Access-Control-Allow-Methods": "OPTIONS,POST,GET,PATCH"
29         },
30         "body": JSON.stringify({ 'invMs': res.Attributes })
31     }
32 })
33
34 module.exports = {
35     updateInvMs
36 }
37
```

Figura 31. Función lambda para actualizar datos.

```

getVenta > JS handler.js > [⌘] getVenta
1  const aws = require("aws-sdk")
2  let dynamoDBClientsParams = {}
3  const dynamodb = new aws.DynamoDB.DocumentClient(dynamoDBClientsParams)
4  const getVenta = async (event, context) => {
5
6      let userId = event.pathParameters.id
7
8      var params = {
9          ExpressionAttributeValues: {':pk': userId},
10         KeyConditionExpression: 'pk = :pk',
11         TableName: 'ventastable'
12     };
13
14     return dynamodb.query(params).promise().then (res => {
15         console.log(res)
16         return {
17             "statusCode": 200,
18             headers: {
19                 "Access-Control-Allow-Headers" : "Content-Type",
20                 "Access-Control-Allow-Origin": "*",
21                 "Access-Control-Allow-Methods": "OPTIONS,POST,GET"
22             },
23             "body": JSON.stringify({ 'venta': res})
24         }
25     })
26 }
27
28 module.exports = {
29     getVenta
30 }
31

```

Figura 32. Función lambda para obtener los datos.

```

deleteUsers > handler.py > deleteUsers
1  import boto3, json, os
2
3  client = boto3.resource('dynamodb')
4
5
6
7  table = client.Table('usertable')
8
9
10 def deleteUsers(event, context):
11     user_id = event['pathParameters']['id']
12     result = table.delete_item(Key = {'pk': user_id})
13
14     body = json.dumps( { 'message' : f"user {user_id} deleted"})
15
16     response = {
17         'statusCode': result['ResponseMetadata']['HTTPStatusCode'],
18         'body': body
19     }
20
21     return response
22

```

Figura 33. Función lambda para eliminar datos.

3.8.3 Despliegue del archivo serverless.yml

Finalizada la codificación de las funciones lambdas se despliega el código serverless.yml usando el comando “sls deploy” y al finalizar nos devuelve los endpoint o desencadenadores de cada función lambda

```
C:\Users\pauls\Desktop\serverless\CursoServerless>sls deploy
Running "serverless" from node_modules

Deploying crud-serverlessZomi to stage dev (us-east-1)

✓ Service deployed to stack crud-serverlessZomi-dev (158s)

endpoints:
GET - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/usersall
GET - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/users/{id}
POST - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/users
PATCH - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/users/{id}
PATCH - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/compra/{id}
DELETE - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/users/{id}
GET - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/compras
POST - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/compra
GET - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/invMs
POST - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/invMs
GET - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/invTs
POST - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/invTs
GET - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/ventas
POST - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/venta
GET - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/compra/{id}
GET - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/venta/{id}
GET - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/invTs/{id}
GET - https://dymh9i9rmk.execute-api.us-east-1.amazonaws.com/dev/invMs/{id}
functions:
getAllUsers: crud-serverlessZomi-dev-getallUsers (107 kB)
getUser: crud-serverlessZomi-dev-getUser (107 kB)
createUsers: crud-serverlessZomi-dev-createUsers (107 kB)
updateUsers: crud-serverlessZomi-dev-updateUsers (107 kB)
updateInvMs: crud-serverlessZomi-dev-updateInvMs (107 kB)
updateCompra: crud-serverlessZomi-dev-updateCompra (107 kB)
deleteUsers: crud-serverlessZomi-dev-deleteUsers (107 kB)
getAllCompras: crud-serverlessZomi-dev-getallCompras (107 kB)
createCompras: crud-serverlessZomi-dev-createCompras (107 kB)
```

Figura 34. Despliegue del archivo serverless.yml

3.8.4 Codificación del front-end

Para empezar, necesitamos crear un nuevo proyecto con Astro que es un framework para la construcción de interfaces de usuario en tiempo real y colaborativas.

El siguiente código se utiliza para crear un nuevo proyecto en astro con la última versión:

“npm create astro@latest”

```
PS C:\Users\pauls\Desktop\tallerServerles\BlogAstro\Proyecto_Blanco> npm create astro@latest

astro Launch sequence initiated.

dir Where should we create your new project?
./Astro

tmpl How would you like to start your new project?
✓ Project initialized!
  ■ Template copied
  ■ Dependencies installed
  ■ TypeScript customized
  ■ Git initialized

next Liftoff confirmed. Explore your project!

Enter your project directory using cd ./Astro
Run npm run dev to start the dev server. CTRL+C to stop.
Add frameworks like react or tailwind using astro add.
```

Figura 35. Nuevo proyecto con Astro.

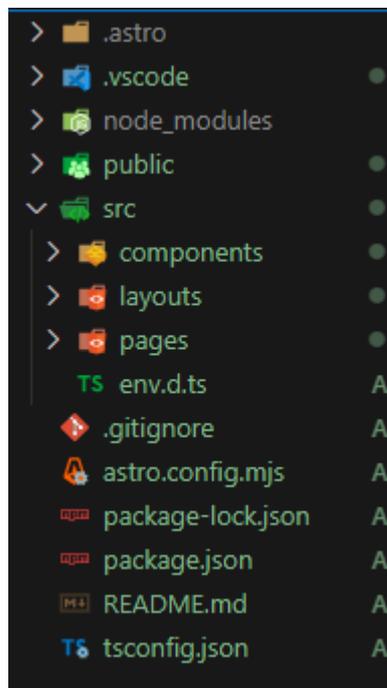


Figura 36 Carpetas del proyecto.

3.8.5 Codificación de los servicios con los endpoints

Para la codificación de los servicios se crea una carpeta de servicios en la parte de componentes y se agrega una función en JS con el método “fetch” para cada endpoint. Cada función se exporta para llamar desde otro archivo y para el desarrollo del proyecto se utiliza cuatro servicios y uno extra que es el servicio de la fecha.

```

src > components > services > JS servicesInvjs > [0] patchProduct
1  const API_URL_inms = 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/invMs';
2  const URL_GET_inms = 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/invMs';
3  const API_PATCH_prod= 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/invMs';
4  const API_POST_prod= 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/invMs';
5
6
7  export const getProduct = async () =>{
8    const res = await fetch(`${API_URL_inms}`)
9    const cardproduct = await res.json();
10   return cardproduct
11  }
12
13  export const getProductId = async ({id}) =>{
14    const res = await fetch(`${URL_GET_inms}/${id}`)
15    const product = await res.json()
16    return product
17  }
18
19  export const patchProduct = async (id, data,pg) => {
20    try {
21      const response = await fetch(`${API_PATCH_prod}/${id}`, {
22        method: 'PATCH',
23        body: JSON.stringify(data)
24      });
25
26      if (response.ok) {
27        const responseData = await response.json();
28        if (pg==='inv') {
29          window.location.href='/inventario'
30        }
31      }
32      return responseData;

```

Figura 37. Servicio de inventario.

```

const API_URL_client = 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/usersall';
const API_POST_client = 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/users';

export const getUser = async () =>{
  const res = await fetch(`${API_URL_client}`)
  const clientes = await res.json();
  return clientes
}

export const postuser = async (data) =>{
  fetch(`${API_POST_client}`, {
    method: 'POST',
    body: JSON.stringify(data)
  })
  .then(response => {
    if (response.ok) {
      return response.json();
    }
    throw new Error('Error al enviar datos');
  })
  .then(data => {
    console.log('Datos enviados correctamente:', data);
  })
  .catch(error => {
    console.error('Error:', error);
  });
}

```

Figura 38. Servicio de usuarios.

```

src > components > services > JS seviceComprasjs > [0] getCompraid > [0] res
1  const URL_GET_ID = 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/compra';
2  const URL_GET_Comp = 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/compras';
3  const URL_POST_Comp = 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/compra';
4  const URL_PATCH_Comp = 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/compra';
5
6
7  export const getCompras = async () =>{
8      const res = await fetch(`${URL_GET_Comp}`)
9      const cardcompra = await res.json();
10     return cardcompra
11 }
12
13 export const getCompraId = async ({id}) =>{
14     const res = await fetch(`${URL_GET_ID}/${id}`)
15     const compra = await res.json()
16     return compra
17 }
18
19 export const postCompra = async (data) =>{
20     try {
21         const response = await fetch(`${URL_POST_Comp}`, {
22             method: 'POST',
23             body: JSON.stringify(data)
24         });
25
26         if (response.ok) {
27             const responseData = await response.json();
28             alert('compra ingresada correctamente')
29             window.location.href='/compras'
30
31             return responseData;

```

Figura 39. Servicio de compras.

```

src > components > services > JS sevicioVentajs > ...
1  const URL_POST_Vent = 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/venta';
2  const URL_GET_Vent = 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/ventas';
3  const URL_GET_VentID = 'https://z7td10cebi.execute-api.us-east-1.amazonaws.com/dev/venta';
4
5  export const getventas = async () =>{
6      const res = await fetch(`${URL_GET_Vent}`)
7      const ventas = await res.json();
8      return ventas
9  }
10
11 export const getventatId = async ({id}) =>{
12     const res = await fetch(`${URL_GET_VentID}/${id}`)
13     const venta = await res.json()
14     return venta
15 }
16
17 export const postVenta = async (data) =>{
18     fetch(`${URL_POST_Vent}`, {
19         method: 'POST',
20         body: JSON.stringify(data)
21     })
22     .then(response => {
23         if (response.ok) {
24             return response.json();
25         }
26         throw new Error('Error al enviar datos');
27     })
28     .then(data => {
29         console.log('Datos enviados correctamente:', data);
30         window.location.href='/ventas'
31         alert('venta exitosa')
32     })

```

Figura 40. Servicio de ventas.

```

src > components > services > JS serviceFecha.js > obtenerFechaEcuador
1  export const obtenerFechaEcuador = async() =>{
2      try {
3          const respuesta = await fetch('https://worldtimeapi.org/api/timezone/America/Guayaquil');
4          if (!respuesta.ok) {
5              throw new Error('Error al obtener la fecha');
6          }
7          const datos = await respuesta.json();
8          const fecha = datos.datetime
9
10
11         return fecha; |
12     } catch (error) {
13         console.error('Error:', error);
14         return null;
15     }
16 }
17

```

Figura 41. Servicio de fecha.

3.8.6 Codificación de los componentes

Los componentes son la parte visual del sistema y para el Proyecto se utiliza los siguientes componentes:

- **Componente de ventas**

```

cardventasastro M x
src > components > cardVentas > cardventas.astro > div#ventas.py-5.bg-bgventas.disabled: > div.flex.justify-center.items-center.mx-5.mb-5.mt-5.gap-3 > input#canti
1  <div id="ventas" class="py-5 bg-bgventas disabled:">
2      <div class="flex justify-center items-center mx-5 mb-5 mt-5 gap-3">
3          <label class="font-semibold text-xl text-black"> Categoria:</label>
4          <select
5              id="categoriasVentas"
6              class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border
7          >
8              <option selected=""> Todos</option>
9          </select>
10         <select
11             id="productos"
12             class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border
13         >
14             <option disabled selected> Elegir Producto</option>
15         </select>
16         <label class="font-semibold text-xl text-black"> Cantidad:</label>
17         <input
18             type="number"
19             id="cantidad"
20             placeholder="Cantidad"
21             value="1"
22             class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border
23         />
24     <div class="text-start w-80">
25         <button
26             id="agregarDatosBtn"
27             class="bg-primary-400 hover:bg-btngreeventas text-white font-bold py-2.5 px-4 w-full rounded-lg"
28         >
29             Agregar
30         </button>
31     </div>
32 </div>

```

Figura 42. Componente de ventas.

- **Componente de compras**

```

comprasCompastro X
src > components > compras > comprasCompastro > div#ventanaCompras > div.flex.flex-row.justify-between.mb-4.mx-5 > div.mt-4 > button#nuevaCompras.text-
29 <div id="ventanaCompras">
30 <div class="flex flex-row justify-between mb-4 mx-5">
31 <div class="flex flex-row gap-1 mb-2 mt-4 w-1/2">
32 <h1 class="font-semibold mt-2">Categoria:</h1>
33 <div class="w-1/2">
34 <select
35   id="categoriase"
36   class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:outline-none"
37 >
38   <option selected="">Todos</option>
39 </select>
40 </div>
41 <div class="w-full">
42 <input
43   type="text"
44   id="busquedaCompras"
45   class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:outline-none"
46   placeholder="Buscar elemento"
47 >
48 </div>
49 </div>
50 <div class="mt-4">
51 <button
52   id="nuevaCompras"
53   type="button"
54   class="text-white bg-primary-400 hover:bg-btnagreenventas focus:ring-blue-300 font-medium rounded-lg text-sm"
55 >Nueva compra</button>
56 </div>
57 </div>
58 </div>
59

```

Figura 43.Componente de compras.

- **Componente de inventario**

```

productos.astro X
src > components > inventario > productos.astro > div.flex.flex-row.mb-4.mx-5justify-between > div.mt-4 > button#nuevoproduct.text-white.bg-primary-400.hoverbg-
49 <div class="mt-4">
50 <button
51   id="nuevoproduct"
52   type="button"
53   class="text-white bg-primary-400 hover:bg-btnagreenventas focus:ring-blue-300 font-medium rounded-lg text-sm"
54 >Nueva Producto</button>
55 </div>
56 </div>
57 </div>
58
59 <div class="border-t-2 border-black border-b-2 mb-2 px-3">
60 <div class="flex justify-around items-center">
61 <div class="w-36 text-left pl-5">Fecha</div>
62 <div class="w-48 text-left ml-2">Producto</div>
63 <div class="w-36 text-left ml-2">Categoria</div>
64 <div class="w-36 text-left ml-2">Ingreso</div>
65 <div class="w-28 text-center">Cantidad</div>
66 <div class="w-36">Precio Venta</div>
67 <div class="w-36 pr-5">Precio Comp.</div>
68 </div>
69 </div>
70
71 <div class="h-72 overflow-y-auto">
72 <div id="app" class="grid gap-x-1 sm:grid-cols-1 lg:grid-cols-1 px-3">
73 {
74   data.map(
75     ({
76       pk,
77       codigo,
78       nombre,
79       cantidad,

```

Figura 44.Componente de inventario.

- **Componente de facturas**

```

factura.astro M X
src > components > cardFacturas > factura.astro > div.h-72.overflow-y-auto
38
39 ---
40
41 <div class="p-5 mx-24 w-1/3">
42
43   <input
44     type="text"
45     id="busquedaCompras"
46     class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-bl
47     block w-full ps-2 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white
48     dark:focus:ring-blue-500 dark:focus:border-blue-500 p-2"
49     placeholder="Buscar Nota de venta"
50   />
51 </div>
52
53 <div class="border-t-2 border-black border-b-2 mb-2 mx-28 mt-2">
54   <div class="flex justify-around items-center">
55     <div class="w-36 text-left pl-5">Fecha</div>
56     <div class="w-20 text-left ml-2">Cliente</div>
57     <div class="w-16 text-right ml-10">Cantidad</div>
58     <div class="w-16 text-right">Total</div>
59   </div>
60 </div>
61
62 <div class="h-72 overflow-y-auto">
63
64 <div id="FacApp" class="grid gap-x-1 sm:grid-cols-1 lg:grid-cols-1 px-28 ">...
65 </div>
66
67 </div>
68
69 </div>

```

Figura 45. Componente facturas.

- **Componente de registro usuario**

```

cardRegistroUser.astro 2, U X
src > components > card_RegistroUsuario > cardRegistroUser.astro > div.bg-white
4
5 <div class="bg-white">
6   <section id="reg" class="flex items-center justify-center py-5">
7     <div class="w-full bg-gray-200 rounded-lg shadow dark:border md:mt-0 sm:max-w-md xl:p-0">
8       <div class="p-1 space-y-1 md:space-y-2 sm:p-8">
9         <div class="flex justify-center">
10          <h1 class="text-xl font-bold leading-tight tracking-tight text-gray-900 md:text-2xl dark:text-white">
11            Registro Usuario
12          </h1>
13        </div>
14      </div>
15    </div>
16    <div>
17      <label for="nameuser" class="block mb-2 text-sm font-medium text-gray-900 dark:text-white">Nombre</label>
18      <input type="text" name="nameuser" id="nameuser" class="bg-gray-50 border border-gray-300 text-gray-900 sm:te
19    </div>
20    <div>
21      <label for="email" class="block mb-2 text-sm font-medium text-gray-900 dark:text-white">Correo</label>
22      <input type="email" name="email" id="email" class="bg-gray-50 border border-gray-300 text-gray-900 sm:text-sm
23    </div>
24    <div>
25      <label for="password" class="block mb-2 text-sm font-medium text-gray-900 dark:text-white">Contraseña</label>
26      <input type="password" name="password" id="password" placeholder="*****" class="bg-gray-50 border border-gray
27    </div>
28    <div>
29      <label for="" class="block mb-2 text-sm font-medium text-gray-900 dark:text-white">Cargo</label>
30      <select id="miComboBox" class="block w-full px-4 py-2 border rounded-md bg-white shadow-sm focus:outline-none focu
31      <option value="">Selecciona una opción</option>
32      <option value="Admin">Admin</option>
33      <option value="Ventas">Ventas</option>
34    </div>
35  </div>

```

Figura 46. Componente de registro usuario

- **Componente Header**

```
header.astro M X
header.astro > header. > nav.grid.grid-cols-1.sm:grid-cols-3.md:grid-cols-3.lg:grid-cols-7.items-center.justify-between.w-full.text-center.px-4.mx-auto.max-w-xl > Headerboton
1 ---
2 import Headerboton from "../components/home/headerboton.astro";
3 ---
4
5 <div class="flex flex-row items-center justify-center text-center"> ...
16 </div>
17 <header transition:persist class="">
18   <nav
19     class="grid grid-cols-1 sm:grid-cols-3 md:grid-cols-3 lg:grid-cols-7 items-center justify-between w-full text-center px-4
20   >
21     <Headerboton ruta="/ventas"> ...
41   </Headerboton>
42     <Headerboton ruta="/inventario"> ...
61   </Headerboton>
62     <Headerboton ruta="/compras"> ...
80   </Headerboton>
81     <Headerboton ruta="/informes"> ...
103  </Headerboton>
104     <Headerboton ruta="/facturas"> ...
122  </Headerboton>
123     <Headerboton ruta="/auth/registroUser"> ...
144  </Headerboton>
145     <a ...
167   </a>
168 </nav>
169 </header>
---
```

Figura 47. Componente header.

3.8.7 Codificación de los Layouts

Los layouts o capas su utilizaron en el menú principal y otro se utilizó para el archivo index, astro.

En todas las páginas del proyecto excepto en el índice, se visualizará el layout de home el que contiene el menú de opciones

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="description" content="Astro description" />
    <meta name="viewport" content="width=device-width" />
    <link rel="icon" type="image/svg+xml" href="/favicon.svg" />
    <meta name="generator" />
    <title>{title}</title>
  </head>
  <body >
    <div class="bg-login">
      <Header />
    </div>
    <main >
      <slot />
    </main>
  </body>
  <script src="../components/auth/logout.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/amazon-cognito-identity-js/dist/amazon-cognito-identity.min.js"></script>
</html>
```

Figura 48. Layout de la página principal

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="description" content="Astro description" />
    <meta name="viewport" content="width=device-width" />
    <link rel="icon" type="image/svg+xml" href="/favicon.svg" />
    <meta name="generator" content={Astro.generator} />
    <title>{title}</title>
  </head>
  <body>
    <slot />
  </body>

  <script src="../../components/auth/login.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/amazon-cognito-identity-js/dist/amazon-cognito-identity.min.js"></script>
</html>
```

Figura 49. Layout para el archivo index, astro.

3.8.8 Codificación de las páginas

La ventaja de astro al momento de codificar las páginas es que se crea un archivo en la carpeta “pages” automáticamente se genera una ruta con el nombre del archivo.

En el proyecto se utilizan páginas con rutas dinámicas es decir que se genera una página por cada elemento de la lista y la ruta es el id del producto y estáticas que no cambia de nombre en la ruta. Además, en la mayoría de la página se importa el Layout y el componente.

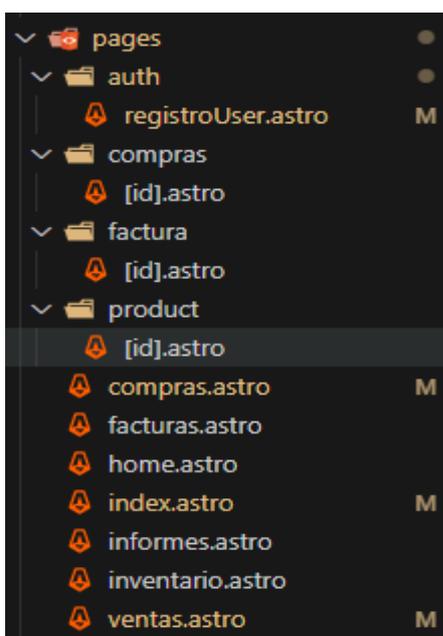


Figura 50. Páginas del proyecto.

- **auth/registroUser.astro**

```

---
import LayoutHome from "../../layouts/Layout_home.astro";
import CardRegistroUser from '../../components/card_RegistroUsuario/cardRegistroUser.astro'
---

<LayoutHome title="registro">

  <CardRegistroUser></CardRegistroUser>

</LayoutHome>

<script src="../../components/auth/seguridadRutas.js"> </script>

```

Figura 51. Página de registro de usuarios.

- **compras/ [id]. astro (página dinámica)**

```

[id].astro M X
src > pages > compras > [id].astro > LayoutHome
1 ---
2 import {getCompraId} from '../../components/services/seviceCompras'
3 import LayoutHome from "../../layouts/Layout_home.astro";
4 const { id } = Astro.params;
5 let compra;
6 if (id) {
7   compra = await getCompraId({ id });
8 }
9 ---
10 <LayoutHome title="compras">
11
12   <div class="flex justify-center items-center pt-1 rounded-lg">
13     <div class="w-1/2 bg-slate-200">
14       <div class="text-center mb-3 flex justify-between">
15         <h1 class="font-semibold text-lg w-full">Información Compra</h1>
16         <button
17           id="cerrarVentanaComp" ...
18           >X</button>
19       </div>
20       <div id="dataproduc" class="px-8" >
21         <div>...
22       </div>
228
229
230       <button ...
231         >Enviar</button>
232     </div>
233   </div>
234
235   </div>
236 </div>
237 </div>
238 <script src="../../components/funciones/compras/ActualizarCompra.js"></script>
239 </LayoutHome>

```

Figura 52. Página de la información de la compra.

- **factura/[id]. astro (página dinámica)**

```
[id].astro M X
src > pages > factura > [id].astro > ...
1 ---
2 import Layout_Home from "../../layouts/Layout_home.astro";
3
4 import { getventatId } from "../../components/services/sevicioVenta";
5
6 const { id } = Astro.params;
7
8 let venta;
9
10 if (id) {
11   |   venta = await getventatId({ id });
12   }
13
14 const keys = Object.keys(venta.venta.Items[0]);
15 let ultimaClave = keys[keys.length - 2];
16 ---
17
18 > <input ...
24 </>
25 > <Layout_Home title="Factura">...
132 </Layout_Home>
133 <script src="../../components/funciones/facturas/llenado.js"></script>
134
```

Figura 53. Página de la información de la factura

- **producto/[id]. astro (página dinámica)**

```
[id].astro M X
src > pages > product > [id].astro > LayoutHome
1 ---
2 import { getProductId , patchProduct } from "../../components/services/servicesInv";
3 import LayoutHome from "../../layouts/Layout_home.astro";
4 const { id } = Astro.params;
5
6 let product;
7 if (id) {
8   |   product = await getProductId({ id });
9   }
10
11 ---
12
13 > <LayoutHome title="productos">...
199 <LayoutHome>
```

Figura 54. Página de la información del producto

- **compras.astro**

```
compras.astro M X
src > pages > compras.astro > LayoutHome > div#comprasp.bg-white
1 ---
2 import LayoutHome from "../layouts/Layout_home.astro";
3 import ComprasComp from '../components/compras/comprasComp.astro'
4 ---
5
6 <LayoutHome title="compras">
7
8   <div id="comprasp" class="bg-white">
9
10    <ComprasComp></ComprasComp>
11
12   </div>
13 <script src="../components/auth/seguridadRutas.js" > </script>
14 <script src="../components/funciones/compras/buquedaCompras.js"></script>
15 <script src="../components/funciones/compras/nuevaCompra.js"></script>
16 <script src="../components/funciones/compras/proveedoresLista.js"></script>
17 <script src="../components/funciones/compras/llevarCompraInventario.js"></script>
18 <script src="../components/funciones/compras/autocompletado.js"></script>
19 </LayoutHome>
```

Figura 55.página de compras

- **facturas.astro**

```
facturas.astro M X
src > pages > facturas.astro > LayoutHome > div.bg-white
1 ---
2 import LayoutHome from "../layouts/Layout_home.astro";
3 import Factura from '../components/cardFacturas/factura.astro'
4 ---
5
6 <script src="../components/auth/seguridadRutas.js"> </script>
7 <LayoutHome title="facturas">
8
9
10 <div class="bg-white">
11
12   <Factura></Factura>
13
14 </div>
15
16
17 </LayoutHome>
```

Figura 56. Página de facturas

- **home.astro**

```
home.astro X
src > pages > home.astro > ...
1 ---
2 import Layout_home from "../layouts/Layout_home.astro";
3 ---
4
5
6 <script src="../components/auth/seguridadRutas.js"> </script>
7 <Layout_home title="Home">
8   <div class="">
9     <div class="bg-white">
10      <div class="flex items-center justify-center">
11        <div></div>
12        
13        <div></div>
14      </div>
15    </div>
16  </div>
17 </Layout_home>
18
```

Figura 57. Página principal

- **index.astro**

```
index.astro M X
src > pages > index.astro > ...
1 ---
2 import Layout from "../layouts/Layout.astro";
3 ---
4
5 <Layout title="Login">
6   <div class="flex justify-between">...
72 </div>
73 </Layout>
74
75 <script>
76   const name = localStorage.getItem(
77     "CognitoIdentityServiceProvider.77jn0p12i0vpe4r3e28m7rnfia.LastAuthUser",
78   );
79   if (
80     localStorage.getItem(
81       `CognitoIdentityServiceProvider.77jn0p12i0vpe4r3e28m7rnfia.${name}.accessTo
82     )
83   ) {
84     location.replace("/home");
85   }
86 </script>
87
```

Figura 58. Página de inicio de sesión

- **informes.astro**

```
informes.astro X
src > pages > informes.astro > LayoutHome > div.bg-bgventas
1 ---
2 import LayoutHome from "../layouts/Layout_home.astro";
3 import CardInforme from '../components/cardInformes/cardInforme.astro'
4 ---
5
6 <script src="../components/auth/seguridadRutas.js"> </script>
7 <LayoutHome title="informes">
8
9 ⚠ <div class="bg-bgventas">
10
11     <CardInforme></CardInforme>
12
13 </div>
14
15
16 <script src="../components/funciones/informes/ventas.js"></script>
17 <script src="../components/funciones/informes/ventaProductos.js"></script>
18 <script src="../components/funciones/informes/compras.js"></script>
19 </LayoutHome>
20
```

Figura 59. Página de informes

- **inventarios.astro**

```
inventario.astro X
src > pages > inventario.astro > LayoutHome > div.bg-white
1 ---
2 import Productos from "../components/inventario/productos.astro";
3 import LayoutHome from "../layouts/Layout_home.astro";
4 ---
5
6
7 <script src="../components/auth/seguridadRutas.js"> </script>
8
9 <LayoutHome title="inventario">
10 ⚠ <div class="bg-white">
11     <div>
12         <Productos></Productos>
13     </div>
14 </div>
15 <script src="../components/funciones/inventario/busquedaInventario.js"></script>
16 <script src="../components/funciones/inventario/crearProducto.js"></script>
17 <script src="../components/funciones/inventario/historialIngreso.js"></script>
18 </LayoutHome>
19
```

Figura 60. Página de inventario

- **ventas.astro**

```
ventas.astro M X
src > pages > ventas.astro > LayoutHome
1 ---
2 import LayoutHome from "../layouts/Layout_home.astro";
3 import Cardventas from "../components/cardVentas/cardventas.astro"
4 ---
5 <script src="../components/auth/seguridadRutas.js"> </script>
6 <LayoutHome transition:persist title="ventas">
7
8   <Cardventas ></Cardventas>
9
10
11 <script src="../components/funciones/ventas/llenadoSelect.js"></script>
12 <script src="../components/funciones/ventas/guardarVenta.js"></script>
13 <script src="../components/funciones/ventas/clientes.js"></script>
14 <script src="../components/services/servicioCliente.js"></script>
15 </LayoutHome>
```

Figura 61. Página de ventas

3.8.9 Codificación de las funciones

En las funciones incluyen las validaciones que aseguran que los datos ingresados cumplan con los criterios esperados, evitando problemas o errores inesperados. Además, incluye los estados para verificar si se realizó correctamente un proceso.



Figura 62. Funciones del proyecto.

3.8.9.1 Funciones de autenticación

- Configuración.js

```
JS configuracion.js U X
src > components > funciones > auth > JS configuracion.js > ...
1 export var poolData = {
2   UserPoolId: 'us-east-1-...', // Rea
3   ClientId: '...', //
4 };
```

Figura 63. Función configuración AWS Cognito

- forgetpass.js

```
JS forgetpass.js U X
src > components > funciones > auth > JS forgetpass.js > validar
1 import { poolData } from './configuracion';
2
3 const userPool = new AmazonCognitoIdentity.CognitoUserPool(poolData)
4 const buttons = document.querySelectorAll('#forgotpass');
5 const mailverify = document.getElementById('correoverify');
6 const userEmail = document.getElementById('email');
7
8 var clases = ['bg-red-50', 'border-red-500', 'text-red-900'];
9
10 buttons.forEach((button) => {
11   button.addEventListener('click', () => {
12     if( validar()){
13       mailverify.classList.add('hidden');
14       enviarcorreo();
15     }else{
16       console.log('ingrese el correo o verifique que este bien escrito')
17     }
18   });
19 });
20 });
21
22 > function validar(){...
36 }
37
38 > function validarCorreo(correo) {...
42 }
43
44 > function enviarcorreo(){...
75 }
```

Figura 64. Función recuperar contraseña.

- **login.js**

```
JS login.js U X
src > components > funciones > auth > JS login.js > buttons.forEach() callback
1 import { poolData } from './configuracion';
2 //console.log(localStorage)
3
4 const buttons = document.querySelectorAll('#login');
5
6
7 buttons.forEach((button) => {
8 > button.addEventListener('click', () => { ...
49 });
50
51 });
52
```

Figura 65. Función ingresar al sistema

- **logout.js**

```
JS logout.js U X
src > components > funciones > auth > JS logout.js > ...
1 import { poolData } from './configuracion';
2 > //console.log(localStorage); ...
8 // document.getElementById('cargo').innerText = localStorage.getItem(
9
10 const userPool = new AmazonCognitoIdentity.CognitoUserPool(poolData)
11 const buttons2 = document.querySelectorAll('#logout');
12
13 const cognitoUser = userPool.getCurrentUser();
14
15 buttons2.forEach((button) => {
16 > button.addEventListener('click', () => {
17 > if (cognitoUser != null) {
18 > cognitoUser.signOut();
19 > console.log("se cerro la session")
20 > localStorage.clear();
21 > console.log(localStorage);
22 > window.location.href='/';
23 > }else{
24 > console.log('no esta logeado')
25 > }
26 > });
27 > });
28
```

Figura 66. Función para salir del sistema

- **registerUser.js**

```
JS registerUser.js U X
src > components > funciones > auth > JS registerUser.js > ...
1  import { poolData } from './configuracion';
2
3  const userPool = new AmazonCognitoIdentity.CognitoUserPool(poolData)
4  const buttons = document.querySelectorAll('#register');
5  const buttonv = document.querySelectorAll('#veficar');
6  const verify = document.getElementById('check');
7  const registro = document.getElementById('reg');
8  > { ...
19 }
20
21 buttons.forEach((button) => {
22 > button.addEventListener('click', () => { ...
31 });
32 });
33
34 > function verificar_campos() { ...
63 }
64
65 > function validarCorreo(correo) { ...
69 }
70
71 > cargo.addEventListener('change', function() { ...
73 });
74
75 > function registrar(){ ...
97 }
98
99 > buttonv.forEach((button) => { ...
110 });
111
112 > function verificar(){ ...
133 }
```

Figura 67. Función para registrar usuario

- **seguridadRutas.js**

```
JS seguridadRutas.js U X
src > components > funciones > auth > JS seguridadRutas.js > ...
1
2  let nombre = localStorage.getItem('CognitoIdentityServiceProvider.
77jn0pl2i0vpe4r3e28m7rnfia.LastAuthUser')
3  > if(!localStorage.getItem(`CognitoIdentityServiceProvider.
77jn0pl2i0vpe4r3e28m7rnfia.${nombre}.accessToken`)) {
4  |   console.log('no estas logeado')
5  |   location.replace("/");
6  | }
```

Figura 68. Función verificar si estas logeado.

3.8.9.2 Funciones para módulo de compras

- ActualizarCompra.js

```
JS ActualizarCompra.js X
src > components > funciones > compras > JS ActualizarCompra.js > ...
1 import {patchCompra} from '../services/seviceCompras'
2 import {patchProduct, getproduct} from '../services/servicesInv'
3 import {obtenerFechaEcuador} from '../services/serviceFecha'
4 const fecha = await obtenerFechaEcuador();
5 let fechahoraEnEcuador
6 const fechaLocal = new Date(fecha);
7 fechaLocal.setHours(fechaLocal.getHours() - 5);
8 const fechaActualISO = fechaLocal.toISOString();
9 const [fechaActual, horaActual] = fechaActualISO.split('T');
10 const partesFecha = fechaActual.split('-');
11 const fechaEnEcuador = `${partesFecha[2]}/${partesFecha[1]}/${partesFecha[0]}`;
12 const [hora, minutos] = horaActual.split(':');
13 fechahoraEnEcuador = `(${fechaEnEcuador}) ${hora}:${minutos}`;
14 const dataproductos: any
15 const dataproductos = await getproduct()
16 let items = {};
17 let producto={};
18 const btnCerrar = document.getElementById('cerrarVentanaComp')
19
20 btnCerrar.addEventListener('click', () => {
21 | window.location.href='/compras'
22 | });
23
24 const ids = document.getElementById('id').value;
25 const codigo = document.getElementById('codigo');
26 const nombre = document.getElementById('nombre');
27 const cantidad = document.getElementById('cantidad');
28 const categoria = document.getElementById('categoria');
29 const precioVenta = document.getElementById('preciov');
30 const precioCompra = document.getElementById('precioc');
31 const descripcion = document.getElementById('descripcion');
32
```

Figura 69. Función actualizar compra

- autocompletado.js

```
JS autocompletado.js X
src > components > funciones > compras > JS autocompletado.js > ...
1 import { getCompras } from '../services/seviceCompras'
2
3 const datoscompras = await getCompras();
4
5 const infoUnicaPorNombre = datoscompras.reduce((acc, compra) => {
6 |   if (!acc[compra.nombre]) {
7 |     acc[compra.nombre] = compra;
8 |   }
9 |   return acc;
10 | }, {});
11
12 const resultado = Object.values(infoUnicaPorNombre);
13
14 const listaproductos = document.getElementById('listaProducto');
15 const codigoc = document.getElementById('inputCodigo');
16 const nombrec = document.getElementById('inputproductoComp');
17 const categoriac = document.getElementById('inputCategoria');
18 const descripcionc = document.getElementById('inputDescrip');
19 const inputproductoComp = document.getElementById('inputproductoComp');
20
21
22 // Función para llenar la lista de clientes
23 function llenarListaproductos() {
24 |   resultado.forEach(producto => {
25 |
26 |     const li = document.createElement('li');
27 |     li.textContent = producto.nombre;
28 |     li.setAttribute('data-codigo', producto.codigo);
29 |     li.setAttribute('data-precio', producto.precio);
30 |     li.setAttribute('data-categoria', producto.categoria);
31 |     li.setAttribute('data-descripcion', producto.descripcion);
32 |     listaproductos.appendChild(li);

```

Figura 70. Función auto completado proveedor.

- **buquedaCompras.js**

```

JS buquedaCompras.js M X
src > components > funciones > compras > JS buquedaCompras.js > ...
22
23 import { getCompras } from '../services/seviceCompras'
24 const compras = await getCompras();
25 const selectc = document.getElementById('categoriase');
26
27
28     const categoriase = compras.map((objeto) => {
29         return objeto.categoria.trim();
30     });
31     const categoriaseUnicas = [...new Set(categoriase)];
32     categoriaseUnicas.forEach(item => {
33         const option = new Option(item);
34         selectc.appendChild(option);
35     });
36
37 let criterioBusqueda
38 const miSelect = document.getElementById('categoriase');
39 miSelect.addEventListener('change', function () {
40     const opcionSeleccionada = this.value;
41
42     const listaProductos = document.querySelector('#compraApp');
43     const productos = Array.from(listaProductos.children);
44     if (opcionSeleccionada === 'Todos') {
45         criterioBusqueda = '';
46
47     }else{
48         criterioBusqueda = opcionSeleccionada.trim().toLowerCase();
49     }
50     const productosFiltrados = productos.filter(producto => {
51         const nombreProducto = producto.textContent.toLowerCase();
52         return nombreProducto.includes(criterioBusqueda);
53     });

```

Figura 71. función búsqueda compra

- **llenarCompraInventario.js**

```

JS llenarCompraInventario.js M X
src > componet let datap: {} compras > JS llenarCompraInventario.js > ...
27
28 let datap = {}
29 let historial = {}
30 let producto = {}
31 let res ;
32
33 const agregar = document.getElementById('agregarCompra');
34
35 const compraInventario = () => {
36
37     datap.codigo = inputCodigo.value
38     datap.nombre = inputProductoComp.value
39     if (inputCodigo.value !== '') {
40         res = buscarProducto(datap)
41
42         if (res.length !== 0 ) {
43             const suma = parseInt(inputCantidad.value) + parseInt(res[0].cantidad)
44             console.log(res)
45             producto.pk = res[0].pk
46             producto.codigo = res[0].codigo
47             producto.nombre = res[0].nombre
48             producto.cantidad = suma
49             producto.precioventa = inputPrecioV.value
50             producto.precio compra = inputPrecioC.value
51             producto.descripcion = res[0].descripcion
52             producto.categoria = res[0].categoria
53             producto.proveedor = res[0].proveedor
54             producto.fecha = fechaHoraEnEcuador
55             producto.origen = 'compras'
56
57             patchProduct(res[0].pk , producto , '' )
58

```

Figura 72. Función llenar inventario por compras

- nuevaCompra.js

```

JS nuevaCompra.js M X
src > components > funciones > compras > JS nuevaCompra.js > ...
1
2 import {postCompra} from '../services/seviceCompras'
3 import {getUser,postuser} from '../services/servicioCliente'
4 import { obtenerFechaEcuador } from '../services/serviceFecha'
5 const fecha = await obtenerFechaEcuador();
6 const dataprovedor = await getUser();
7
8 let item = {
9 }
10
11 > let dataprov = { ...
13 }
14 > { ...
26 > { ...
27 > { ...
41 }
42 > const nuevacompra = ()=>{ ...
81 }
82
83 //-----BOTON DE AGREGAR NUEVA COMPRA-----
84
85 const btnMostrar = document.getElementById('nuevaCompras')
86 const btnCerrar = document.getElementById('cerrarVentanaCompra')
87 const ventanaProduct = document.getElementById('divCompra')
88 const ventanaCompra = document.getElementById('ventanaCompras')
89 const agregar = document.getElementById('agregarCompra')
90
91 btnMostrar.addEventListener('click', () => {
92     ventanaProduct.classList.remove('hidden');
93     ventanaCompra.classList.add('opacity-50', 'pointer-events-none');
94 });

```

Figura 73. Función nueva compra

- proveedoresLista.js

```

JS proveedoresLista.js X
src > components > funciones > compras > JS proveedoresLista.js > ...
1
2 import {getUser} from '../services/servicioCliente'
3
4 const datosprovedor = await getUser();
5
6 // Datos simulados de clientes (esto puede ser una respuesta de una
7
8 const listaprov = document.getElementById('listaProvedor');
9 const inputNombre = document.getElementById('inputProvedor');
10 const inputDireccion = document.getElementById('inputDireccion');
11 const inputNumero = document.getElementById('inputTelefono');
12 const inputEmail = document.getElementById('inpuCorreo');
13
14
15 // Función para llenar la lista de clientes
16 function llenarListaClientes() {
17     datosprovedor.forEach(provedor => {
18
19         if (provedor.tipo === 'provedor') {
20             const li = document.createElement('li');
21             li.textContent = provedor.name;
22             li.setAttribute('data-nombre', provedor.name);
23             li.setAttribute('data-numero', provedor.telefono);
24             li.setAttribute('data-email', provedor.email);
25             li.setAttribute('data-direc', provedor.direccion);
26             listaprov.appendChild(li);
27         }
28     });
29 }
30

```

Figura 74. función lista de proveedores

3.8.9.3 Funciones para facturas

- llenado.js

```
JS llenado.js M X
src > components > funciones > facturas > JS llenado.js > ...
1 import {getventatId} from '../services/sevicioVenta'
2
3 const inputpk = document.getElementById('inputpk')
4 const cerraventana = document.getElementById('cerrarVentanafact')
5 const id=inputpk.value
6 const data = await getventatId({id})
7 const keys = Object.keys(data.venta.Items[0]);
8 const ultimaClave = keys[keys.length - 2];
9
10 > const mostrardatos = () => { ...
36 }
37
38 mostrardatos();
39
40
41 cerraventana.addEventListener('click', ()=>{
42     window.location.href= '/facturas'
43 });
```

Figura 75. Función llenar datos factura.

3.8.9.4 Funciones para informes

- compras.js

```
JS compras.js M X
src > components > funciones > informes > JS compras.js > ...
31
32 let sumcompras = 0;
33 let TotaldecomprasMes = 0;
34 let contadProductosmes = 0;
35 let su const calculoCantidadesTotalCompras: () => void
36 const calculoCantidadesTotalCompras = () => {
37     for (let i = 0; i < compras.length; i++) {
38         const total = compras[i].total;
39         sumcompras = sumcompras + parseFloat(total)
40     }
41     document.getElementById('totalcompras').textContent = sumcompras
42 }
43 calculoCantidadesTotalCompras();
44
45 let summescompria = 0;
46 > const calculoMesAnioCompras = (mesarg) => { ...
110 }
111 calculoMesAnioCompras(fechaconvertida);
112 // ----- INPUT DE LAS FECHAS -----
113 const inputFechaMesAniocomp = document.getElementById('mesAniocompras');
114 > inputFechaMesAniocomp.addEventListener('change', function (event) { ...
120 });
121
122 const inputFecha = document.getElementById('fechacompletaCompra');
123 > inputFecha.addEventListener('change', function (event) { ...
131 });
132
```

Figura 76. función informes compras.

- **ventas.js**

```

JS ventas.js M X
src > components > funciones > informes > JS ventas.js > ...
30
31 let suma = 0;
32 let sumaFechas = 0;
33 let TotaldeVentas = 0;
34 let TotaldeVentasmes = 0;
35 let sumames = 0;
36
37 const calculoCantidades = (fechaselect) => {
38   for (let i = 0; i < ventas.length; i++) {
39     const keys = Object.keys(ventas[i]);
40     const ultimaClave = keys[keys.length - 2];
41     const fecha = ventas[i][ultimaClave].fecha.split(' ');
42     if (fecha[0] === fechaselect) {
43       const total = ventas[i][ultimaClave].total;
44       sumaFechas = sumaFechas + parseFloat(total)
45       TotaldeVentas++
46     }
47   }
48
49   totalFec.textContent = ` ${sumaFechas.toFixed(2)}$`
50   document.getElementById('numVentas').textContent = TotaldeVentas
51   document.getElementById('totalventasfinal').textContent = ventas.length
52
53   sumaFechas = 0;
54   TotaldeVentas = 0;
55 }
56
57 const calculoCantidadesTotal = () => {
58   for (let i = 0; i < ventas.length; i++) {
59     const keys = Object.keys(ventas[i]);
60     const ultimaClave = keys[keys.length - 2];
61     const total = ventas[i][ultimaClave].total;

```

Figura 77. Función informe ventas

3.8.9.5 Funciones para el inventario

- **ActualizarProducto.js**

```

JS ActualizarProducto.js M X
src > components > funciones > inventario > JS ActualizarProducto.js > ...
13 fechahoraEnEcuador = ` ${fechaEnEcuador} ${hora}:${minutos}`;
14
15 let items = {};
16 let historial = {}
17
18 const btnCerrar = document.getElementById('cerrarVentanaproducto')
19
20 btnCerrar.addEventListener('click', () => {
21   window.location.href = '/inventario'
22 });
23
24 const inputcantidad = document.getElementById('inputcantidad').value;
25 const inputingreso = document.getElementById('ingreso').value;
26 let suma
27 suma = parseInt(inputcantidad) + parseInt(inputingreso)
28 let cantanterior = inputcantidad
29 console.log(suma)
30
31 const btnenviar = document.getElementById('enviardatos');
32
33 > btnenviar.addEventListener('click', () => { ...
76 });
77
78 const origen = document.getElementById('origen').value;
79 const ventana = document.getElementById('dataproducto');
80 const proveedor = document.getElementById('proveedor');
81
82 if(origen === 'produccion'){
83   ventana.classList.remove('opacity-50', 'pointer-events-none')
84   btnenviar.classList.remove('hidden')
85   proveedor.classList.add('hidden')

```

Figura 78. Función actualizar producto inventario

- **busquedaInventario.js**

```

JS busquedaInventario.js X
src > components > funciones > inventario > JS busquedaInventario.js > ...
1
2 let criterioBusqueda
3 const miSelectcateg = document.getElementById('categorias');
4 miSelectcateg.addEventListener('change', function () {
5     const opcionSeleccionada = this.value;
6
7     const listaProductos = document.querySelector('#app');
8     const productos = Array.from(listaProductos.children);
9     if (opcionSeleccionada === 'Todos') {
10        criterioBusqueda = '';
11
12    } else {
13        criterioBusqueda = opcionSeleccionada.trim().toLowerCase();
14    }
15    const productosFiltrados = productos.filter(producto => {
16        const nombreProducto = producto.textContent.toLowerCase();
17        return nombreProducto.includes(criterioBusqueda);
18    });
19    productos.forEach(producto => {
20        if (productosFiltrados.includes(producto)) {
21            producto.style.display = 'block';
22        } else {
23            producto.style.display = 'none';
24        }
25    });
26    productosVisibles = productosFiltrados.filter(producto => producto.style.
27    console.log(productosVisibles)
28 });
29

```

Figura 79. Función búsqueda inventario.

- **crearProducto.js**

```

JS crearProducto.js X
src > components > funciones > inventario > JS crearProducto.js > ...
43
44 document.getElementById('agregarProducto').addEventListener('click', (
45
46     ventanaProduct.classList.remove('hidden');
47     const codigo = document.getElementById('inputCodigo').value;
48     const nombre = document.getElementById('inputNombreP').value;
49     const cantidad = document.getElementById('inputCantidad').value;
50     const categoria = document.getElementById('inputCategoria').value;
51     const precioVenta = document.getElementById('inputPrecioV').value;
52     const precioCompra = document.getElementById('inputPrecioC').value;
53     const descripcion = document.getElementById('inputDescrip').value;
54
55     items.codigo=codigo
56     items.nombre = nombre
57     items.cantidad =cantidad
58     items.categoria = categoria
59     items.precioventa = precioVenta
60     items.preciocompra =precioCompra
61     items.descripcion =descripcion
62     items.fecha =fechahoraEnEcuador
63     items.origen = 'produccion'
64
65     historial.codigo = codigo
66     historial.nombre =nombre
67     historial.cantidad =cantidad
68     historial.fecha =fechahoraEnEcuador
69     historial.origen = 'produccion'
70
71     postProductoHist(historial)
72     postProducto(items , 'inventario');
73
74 });

```

Figura 80. Función nuevo producto inventario.

- `llenadoCategoriainv.js`

```

JS llenadoCategoriainv.js X
src > components > funciones > inventario > JS llenadoCategoriainv.js > ...
1  import { getproduct } from '../services/servicesInv'
2
3
4  const data = await getproduct();
5  const select = document.getElementById('categorias');
6
7  const categorias = data.map((objeto) => {
8      return objeto.categoria.trim();
9  });
10 const categoriasUnicas = [...new Set(categorias)];
11
12 categoriasUnicas.forEach(item => {
13     const option = new Option(item);
14     select.appendChild(option);
15 });

```

Figura 81. Función llenado categorías inventario

3.8.9.6 Funciones para ventas

- `clientes.js`

```

JS clientes.js X
src > components > funciones > ventas > JS clientes.js > ...
54 // Evento click en la lista para autocompletar los campos
55 listaClientes.addEventListener('click', function(event) {
56     if (event.target.tagName === 'LI') {
57         inputCedula.value = event.target.getAttribute('data-cedula');
58         inputNombre.value = event.target.getAttribute('data-nombre');
59         inputNumero.value = event.target.getAttribute('data-numero');
60         inputEmail.value = event.target.getAttribute('data-email');
61         inputDirec.value = event.target.getAttribute('data-direc');
62         listaClientes.style.display = 'none';
63     }
64 });
65
66 // Llenar la lista de clientes al cargar la página
67 llenarListaClientes();
68
69 import { getproduct } from '../services/servicesInv'
70
71
72 const data = await getproduct();
73 const select = document.getElementById('categoriasVentas');
74
75 const categorias = data.map((objeto) => {
76     return objeto.categoria.trim();
77 });
78 const categoriasUnicas = [...new Set(categorias)];
79
80 categoriasUnicas.forEach(item => {
81     const option = new Option(item);
82     select.appendChild(option);
83 });
84

```

Figura 82. Función auto completado cliente

- guardarVenta.js

```

JS guardarVenta.js X
src > components > funciones > ventas > JS guardarVenta.js > ...
98 });
99
100 botonCerrar.addEventListener('click', () => {
101   if (inputEmail.value !== '' && inputNombre.value !== '') {
102     ventasp.classList.remove('opacity-50', 'pointer-events-none');
103     divFlotante.classList.add('hidden');
104     const valorNombre = inputNombre.value;
105     const valorTelefo = inputTelef.value;
106     const valorEmail = inputEmail.value;
107     const valorDirec = inputDirec.value;
108     const valorCedula = inputCedula.value;
109     let cliente = {
110       cedula: valorCedula,
111       name: valorNombre,
112       telefono: valorTelefo,
113       email: valorEmail,
114       direccion: valorDirec,
115       tipo: 'cliente',
116     };
117     const resultClient = buscarCliente(cliente)
118     if (resultClient.length === 0) {
119       postuser(cliente) // METODO FECH CLIENTE
120       console.log('clienite nuevo');
121     }
122     listaproductos.push({total: totalGeneral.toFixed(2) , fecha: fechaha
123     postVenta(listaproductos); //METODO FECH VENTA
124     actualizardatos(listaproductosActualizar);
125     limpiar();
126     limpiarDetalle();
127     limpiardatosCliente();
128

```

Figura 83. Función guardar vena

- llenadoSelect.js

```

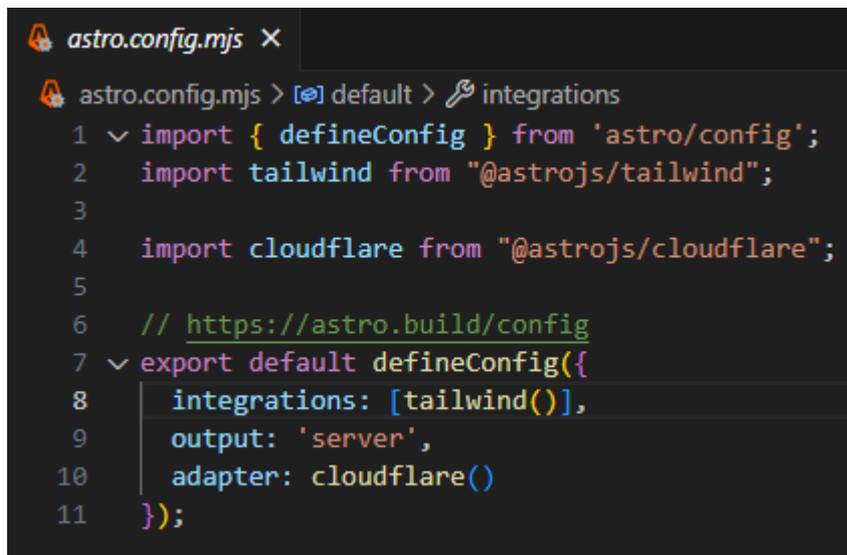
JS llenadoSelect.js X
src > components > funciones > ventas > JS llenadoSelect.js > ...
1
2 import { getproduct } from "../services/servicesInv";
3 const data = await getproduct();
4
5 let elementosguardado = [];
6 let nombreguardado = [];
7
8 let itemseleccionado = '';
9 let datosseleccionados = '';
10
11 export const buscarDatos = (term) => {
12
13   const searchTerm = term.toLowerCase();
14   const resultados = data.filter(item =>
15     item.nombre.toLowerCase().includes(searchTerm)
16   );
17
18   return resultados
19 };
20 let dataFiltrada;
21
22 let selecCategoria = document.getElementById('categoriasVentas');
23 let selecProducto = document.getElementById('productos');
24
25
26 > const itemsCategoria = () => { ...
54 }
55
56
57 > if (selecCategoria.value === 'Todos') { ...
63 }
64

```

Figura 84. Función llenar categorías

3.8.10 Codificación de las integraciones

En el proyecto se utilizaron dos integraciones: la primera es tailwind css para dar estilos a las páginas, y la segunda es Cloudflare, que es el adaptador SSR que refiere a un Adaptador de Renderizado del Lado del Servidor.



```
astro.config.mjs X
astro.config.mjs > default > integrations
1 import { defineConfig } from 'astro/config';
2 import tailwind from '@astrojs/tailwind';
3
4 import cloudflare from '@astrojs/cloudflare';
5
6 // https://astro.build/config
7 export default defineConfig({
8   integrations: [tailwind()],
9   output: 'server',
10  adapter: cloudflare()
11 });
```

Figura 85. Integraciones del proyecto

3.9 Fase IV: Pruebas

3.9.1 Pruebas de aceptación

En la fase de pruebas se verifica la funcionalidad de la aplicación web con el propósito de determinar la validez, basándose en las iteraciones basados en las historias de usuario

Tabla 65. Prueba de aceptación número 1.

Prueba de aceptación	
Número: 1	Historia de usuario: 1
Nombre: Inicio de sesión	
Descripción: Verificar la funcionalidad de inicio de sesión al sistema.	
Condición de ejecución: El sistema está desplegado, accesible y el usuario debe constar en el sistema.	
Entrada: Correo de usuario y contraseña válidos.	
Resultado esperado: Acceso exitoso al panel principal del sistema.	
Evaluación de prueba: Satisfactoria.	

Tabla 66. Prueba de aceptación numero 2

Prueba de aceptación	
Número: 2	Historia de usuario: 2
Nombre: Registro de usuarios	
Descripción: Verificar que solo el administrador pueda registrar nuevos usuarios en el sistema.	
Condición de ejecución: El sistema está disponible y solo permite que el rol de administrador acceda al formulario de registro de usuarios.	
Entrada: Datos válidos del nuevo usuario	
Resultado esperado: <ul style="list-style-type: none"> • Cuando un usuario con rol diferente al de administrador intente acceder al formulario de registro, se debe negar el acceso • Registro exitoso 	
Evaluación de prueba: Satisfactoria.	

Tabla 67. Prueba de aceptación numero 3

Prueba de aceptación	
Número: 3	Historia de usuario: 3
Nombre: Recuperación de Contraseña	
Descripción: Validar el proceso de recuperación de contraseña a través del correo electrónico.	
Condición de ejecución: El sistema permite la recuperación de contraseña únicamente mediante el correo electrónico asociado a la cuenta	
Entrada: Dirección de correo electrónico asociada a la cuenta del usuario.	
Resultado esperado: Se envía un código para restablecer la contraseña al correo electrónico proporcionado.	
Evaluación de prueba: Satisfactorio	

Tabla 68. Prueba de aceptación numero 4

Prueba de aceptación	
Número: 4	Historia de usuario: 4
Nombre: Menú principal	
Descripción: Verificar que el menú principal muestre opciones diferentes según el rol del usuario.	
Condición de ejecución: El sistema muestra opciones específicas en el menú principal según el rol del usuario.	
Entrada: iniciar sesión con diferentes roles de usuario (por ejemplo, administrador o usuario ventas).	
Resultado esperado: El menú principal reflejar las opciones correspondientes al rol del usuario.	
Evaluación de prueba: Satisfactorio.	

Tabla 69. Prueba de aceptación numero 5

Prueba de aceptación	
Número: 5	Historia de usuario: 5
Nombre: Registro de productos.	
Descripción: El administrador debe ingresar los datos en el formulario y validar el proceso de registro de productos en el sistema.	
Condición de ejecución: El sistema permite registrar productos con detalles específicos.	
Entrada: Datos válidos del producto (nombre, descripción, precio, etc.).	
Resultado esperado: El sistema confirma el registro exitoso del producto y almacena los detalles correctamente.	
Evaluación de prueba: Satisfactoria	

Tabla 70. Prueba de aceptación numero 6

Prueba de aceptación	
Número: 6	Historia de usuario: 6
Nombre: Información del producto	
Descripción: El administrador tiene la capacidad de visualizar, modificar y eliminar la información de un producto.	
Condición de ejecución: El sistema permite al administrador acceder y realizar acciones en la información de los productos.	
Entrada: Iniciar sesión como administrador y seleccionar un producto para ver, modificar o eliminar.	
Resultado esperado: El sistema permite al administrador visualizar, editar o eliminar la información del producto seleccionado.	
Evaluación de prueba: Satisfactoria.	

Tabla 71. Prueba de aceptación numero 7

Prueba de aceptación	
Número: 7	Historia de usuario: 7
Nombre: Lista de productos del inventario.	
Descripción: los usuarios pueden acceder y visualizar la lista de productos en inventario.	
Condición de ejecución: El sistema permite a cualquier usuario acceder a la lista de productos disponibles en el inventario.	
Entrada: Iniciar sesión como usuario del sistema.	
Resultado esperado: El sistema muestra una lista completa y detallada de los productos disponibles en el inventario.	
Evaluación de prueba: Satisfactoria.	

Tabla 72. Prueba de aceptación numero 8

Prueba de aceptación	
Número: 8	Historia de usuario: 8
Nombre: Registro de compras.	
Descripción: Solo el administrador tendrá la opción para que pueda ingresar nuevas compras en el sistema.	
Condición de ejecución: El sistema solo permite que el rol de administrador registre nuevas compras.	
Entrada: Iniciar sesión como administrador y acceder a la sección de registro de compras.	
Resultado esperado: El sistema permite al administrador ingresar y registrar nuevas compras de productos.	
Evaluación de prueba: Satisfactorio.	

Tabla 73. Prueba de aceptación numero 9

Prueba de aceptación	
Número: 9	Historia de usuario: 9
Nombre: Información de las compras.	
Descripción: <ul style="list-style-type: none"> • Validar que únicamente el administrador tenga acceso de Leer, Actualizar y Eliminar un registro de compra. • Solo ciertos campos son permitidos para modificar. 	
Condición de ejecución: El rol del usuario debe ser administrador	
Entrada: <ul style="list-style-type: none"> • Iniciar sesión como administrador y acceder al módulo de gestión de compras. • El administrador cambia los datos o elimina de la información de la compra • Validar los cambios utilizando los botones 	
Resultado esperado: El administrador puede leer, actualizar y eliminar registros de compras.	
Evaluación de prueba: Satisfactorio.	

Tabla 74. Prueba de aceptación numero 10

Prueba de aceptación	
Número: 10	Historia de usuario: 10
Nombre: Historial de compras	
Descripción: El administrador tienen acceso al historial completo de compras realizadas en el sistema web. Esto permite visualizar y revisar todas las transacciones previas	
Condición de ejecución: El rol del usuario debe ser administrador	
Entrada: <ul style="list-style-type: none"> • Acceder a la sección de historial de compras. • Visualizar la lista completa de compras realizadas con detalles relevantes (productos, fechas, cantidades, etc.). 	
Resultado esperado: Acceso efectivo al historial de compras con información detallada y precisa.	
Evaluación de prueba: Satisfactoria.	

Tabla 75. Prueba de aceptación numero 11

Prueba de aceptación	
Número: 11	Historia de usuario: 11
Nombre: Registro de ventas	
Descripción: Todos los usuarios del sistema web pueden realizar ventas registrando los detalles de la transacción. Además, pueden filtrar por categorías y seleccionar la cantidad e ingresar a una lista de productos seleccionados.	
Condición de ejecución: La funcionalidad de registro de ventas está disponible para todos los usuarios del sistema.	
Entrada: <ul style="list-style-type: none"> • Acceder al formulario de registro de ventas. • Seleccionar los productos requeridos con los detalles (productos, cantidades, precios, etc.). • Botón para agregar a la lista de productos seleccionados 	
Resultado esperado: se visualiza el listado de los productos seleccionados con los detalles de cada producto y con la opción de aumentar o disminuir las cantidades o eliminar el producto de la lista.	
Evaluación de prueba: satisfactoria.	

Tabla 76. Prueba de aceptación numero 12

Prueba de aceptación	
Número: 12	Historia de usuario: 12
Nombre: Detalle de la venta	
Descripción: Esta función permite al usuario ingresar los datos del cliente, visualizar los detalles de los productos seleccionados para la venta y confirmar la compra final.	
Condición de ejecución: Esta función está disponible para todos los usuarios	
Entrada: <ul style="list-style-type: none"> • Acceder al formulario de generación de detalle de ventas. • Ingresar los datos del cliente (nombre, dirección, contacto, etc.). • Seleccionar los productos deseados para la venta. • Visualizar los detalles de los productos seleccionados. • Confirmar la compra. 	
Resultado esperado: ventas realizadas exitosamente y lista para generar un comprobante.	
Evaluación de prueba: Satisfactoria.	

Tabla 77. Prueba de aceptación numero 3

Prueba de aceptación	
Número: 13	Historia de usuario: 13
Nombre: Historial de ventas	
Descripción: Los usuarios pueden acceder a un historial completo de ventas, ordenado por fecha, con la capacidad de interactuar para ver detalles de la venta y generar un comprobante de manera opcional.	
Condición de ejecución: Esta función está disponible para todos los usuarios del sistema.	
Entrada: <ul style="list-style-type: none"> • Acceder a la sección de historial de ventas. • Visualizar una lista de todas las ventas ordenadas por fecha. • Interactuar para ver detalles específicos de una venta seleccionada. • Opcionalmente, generar un comprobante de la venta. 	
Resultado esperado: visualización la lista ordenada por fecha, ver detalles específicos de las ventas y generar un comprobante en caso necesario.	
Evaluación de prueba: Satisfactoria.	

Tabla 78. Prueba de aceptación numero 14

Prueba de aceptación	
Número: 14	Historia de usuario: 14
Nombre: Informes	
Descripción: El administrador puede generar informes filtrados por fecha que muestran los resultados de ventas y compras durante un período específico.	
Condición de ejecución: Esta funcionalidad está disponible únicamente para el administrador del sistema.	
Entrada <ul style="list-style-type: none"> • Acceder al módulo de generación de informes. • Seleccionar un rango de fechas para filtrar las ventas y compras. • Visualizar informe. 	
Resultado esperado: Capacidad para filtrar y generar informes que muestren los resultados de ventas y compras durante un período específico.	
Evaluación de prueba: Satisfactoria.	

Tabla 79. Prueba de aceptación numero 15

Prueba de aceptación	
Número: 15	Historia de usuario: 15
Nombre: Finalizar sesión	
Descripción: Todos los usuarios tienen la opción de cerrar su sesión en el sistema, eliminando los datos de la sesión actual.	
Condición de ejecución: Esta funcionalidad está disponible para todos los usuarios del sistema.	
Entrada: Localizar y hacer clic en la opción "Salir".	
Resultado esperado: <ul style="list-style-type: none"> • La capacidad del sistema para cerrar la sesión actual y eliminar los datos de esta. • Redireccionar a la página de Login. 	
Evaluación de prueba: Satisfactoria.	

3.9.2 Pruebas de usabilidad

Se realizaron pruebas de uso con el propósito de evaluar la experiencia y facilidad del uso del sistema web por parte del administrador y el usuario de ventas. Se utilizó una escala de Likert para medir el nivel de satisfacción en términos de usabilidad y experiencia del usuario en la aplicación.

- **Totalmente de Acuerdo**
- **De acuerdo**
- **Neutral**
- **En desacuerdo**
- **Totalmente en desacuerdo**

A continuación, se presentan los resultados obtenidos de las pruebas, realizadas a los usuarios de la microempresa “ZOMI”

Tabla 80. Resultados prueba usabilidad usuario administrador.

Resultados prueba de usabilidad	
Tipo/Rol de usuario: Administrador	Evaluación
El sistema permite el registro de ventas adecuadamente	Totalmente de Acuerdo
El sistema permite el registro de compras y visualizar el historial de cada registro	Totalmente de Acuerdo
El sistema permite la gestión del inventario y la edición de la información	Totalmente de Acuerdo
EL sistema permite visualizar informes Filtrando por fechas	Totalmente de Acuerdo
El manejo y uso en general del sistema es fácil	Totalmente de Acuerdo
La interfaz está diseñada adecuadamente para su uso, tanto en términos de colores y estructura.	De acuerdo
Los distintos módulos están organizados y estructurados	De acuerdo

Tabla 81. Resultados prueba usabilidad usuario ventas

Resultados prueba de usabilidad	
Tipo/Rol de usuario: Ventas	Evaluación
El sistema permite el registro de ventas adecuadamente	Totalmente de Acuerdo
El sistema permite el registro de compras y visualizar el historial de cada registro	Totalmente de Acuerdo
El sistema permite la gestión del inventario y la edición de la información	Totalmente de Acuerdo
EL sistema permite visualizar informes Filtrando por fechas	Totalmente de Acuerdo

El manejo y uso en general del sistema es fácil	De acuerdo
La interfaz está diseñada adecuadamente para su uso, tanto en términos de colores y estructura.	De acuerdo
Los distintos módulos están organizados y estructurados	De acuerdo

Finalizada la evaluación de usabilidad con los usuarios principales de la microempresa ZOMI, se puede concluir que la aplicación web cumple con las funciones propuestas, además, presenta una estructura y organización correcta esto para evitar confusiones. Además, su diseño es agradable para los usuarios. Sin embargo, en cuanto al manejo, se tiene una leve dificultad, ya que los usuarios pueden no estar completamente familiarizados con el uso de sistemas.

3.10 Fase V: Implantación

Para la implementación de la aplicación web, solo se debe subir la parte del front-end, ya que el back-end se encuentra en la nube del proveedor de AWS teniendo la facilidad de realizar pocas configuraciones para la implementación, esto se llevó a cabo mediante la plataforma de cloudflare.

3.10.1 Proceso de Configuración

Figura 86. Creación de la cuenta en Cloudflare.com

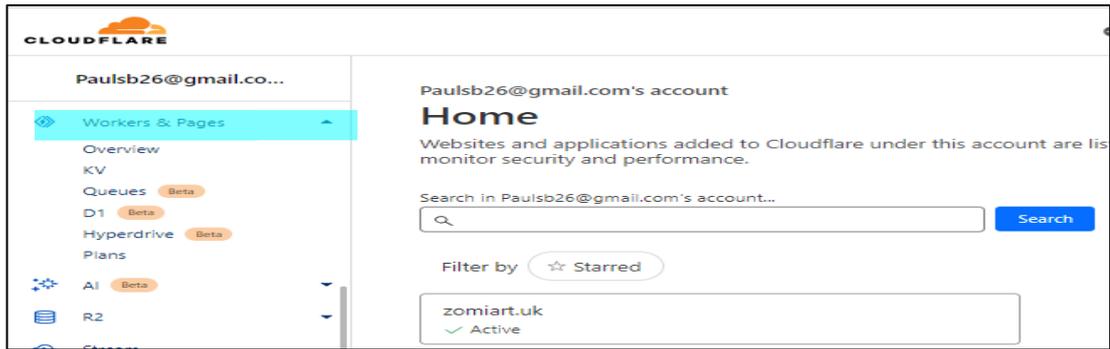


Figura 87. Menú principal de Cloudflare

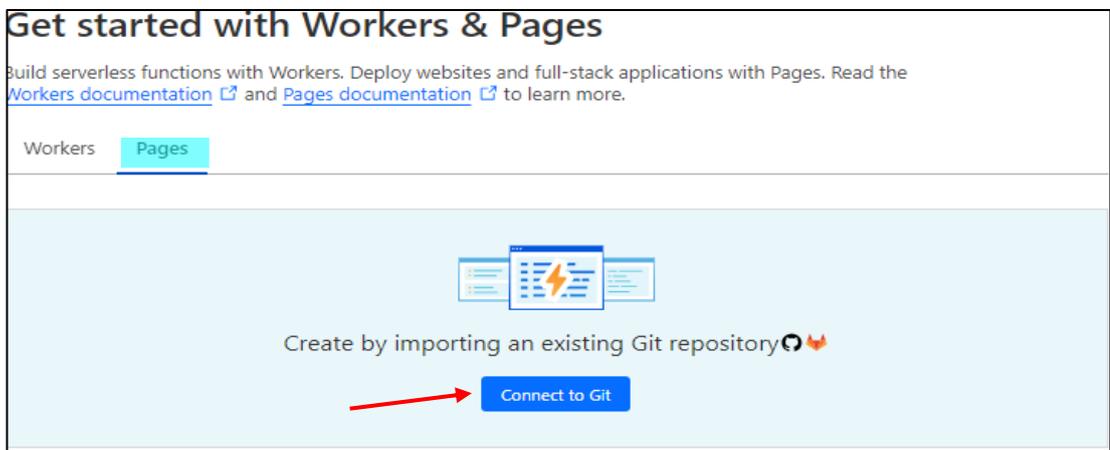


Figura 88. Nueva página con Git

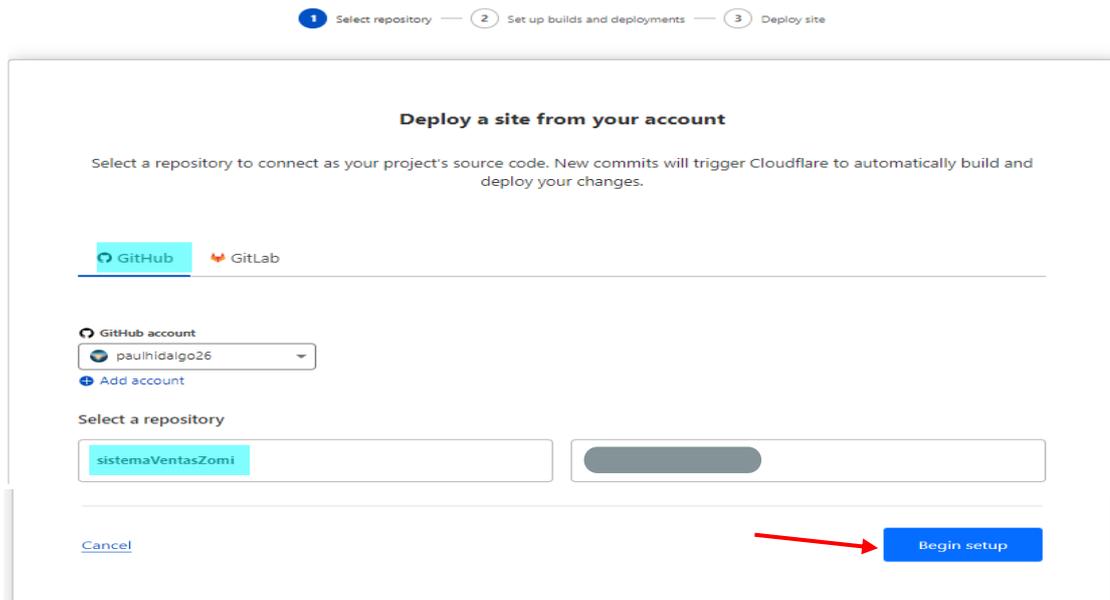


Figura 89. Selección de repositorio Git

Set up builds and deployments

Configure automatic builds and deployments for [paulhidalgo26/sistemaVentasZomi](#)

Project name

Your project will be deployed to [sistemaventaszomi.pages.dev](#).

Production branch

Pushes to this branch automatically trigger deployments to the Production environment. Pushes to all other branches will trigger deployments within the Preview environment.

Build settings [Configuring builds](#)

If your project uses a static site generator or build tool, set the build instructions for Cloudflare.

Framework preset

Select a framework to prefill recommended settings.

Build command ⓘ

e.g. npm run build

Build output directory ⓘ

e.g. dist

> [Root directory \(advanced\)](#)

> [Environment variables \(advanced\)](#)

[← Change repository](#) [Save and Deploy](#)

Figura 90. Selección de framework y despliegue.

✓ Select repository — ✓ Set up builds and deployments — **3** Deploy site

Building and deploying

✓ Initializing build environment	2s
. Cloning git repository	3s
<i>Building application</i>	—
<i>Deploying to Cloudflare's global network</i>	—
<i>Deploying to Cloudflare's global network</i>	—

[Download log](#) [Copy log](#)

```

20:48:03.877 Cloning repository...
20:48:04.959 From https://github.com/paulhidalgo26/sistemaVentasZomi
20:48:04.960 * branch      8343e6d safe95294d2bb973d1def40402916101f -> FETCH_HEAD
20:48:04.960
20:48:05.010 HEAD is now at 8343e6d cambio de endpoints
20:48:05.010
20:48:05.107
20:48:05.136 Success: Finished cloning repository files
20:48:07.510 Detected the following tools from environment: npm@9.6.7, nodejs@18.17.1
20:48:07.510 Installing project dependencies: npm clean-install --progress=false
          
```

Build and deployment settings

Repository: [paulhidalgo26/sistemaVentasZomi](#)

Branch: [main](#) [8343e6d](#) cambio de endpoints

Status: ✓ Success 8:50PM January 9, 2024

Duration: 2m 28s

[> Build settings](#)

[← Change settings](#)
Continue to project

Figura 91. Fases del despliegue.

[Deployments](#) [Functions metrics](#) [Custom domains](#) [Settings](#)

Production [Visit site](#)

[Automatic deployments enabled](#)

Domains: [sistemaventaszomi.pages.dev](#)

Production	main 8343e6d fee20e7e.sistemaventaszomi.pages.dev	✓ 18 minutes ago View details
--	---	--

Figura 92. Información del despliegue finalizado

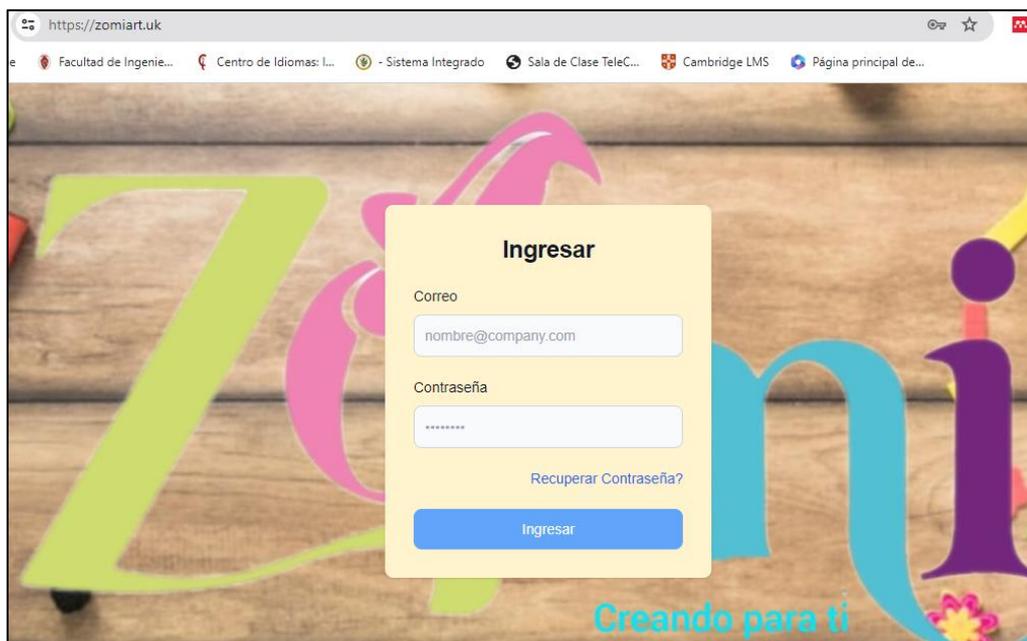


Figura 93. Aplicación implementada.

3.10.2 Capacitaciones

Una vez finalizada la implementación del sistema, se realizó una capacitación al personal de la microempresa ZOMI

Tabla 82. Capacitaciones.

	Encargada	Dirigida A	Fecha inicio	Fecha final	6/1/2024
realizar ventas	Paul Hidalgo	Vendedores /Administrador	6/1/2024	6/1/2024	
lista de inventario	Paul Hidalgo	Vendedores /Administrador	6/1/2024	6/1/2024	
nueva compra	Paul Hidalgo	Administrador	6/1/2024	6/1/2024	
modificar producto	Paul Hidalgo	Administrador	6/1/2024	6/1/2024	
modificar compra	Paul Hidalgo	Administrador	6/1/2024	6/1/2024	
informes	Paul Hidalgo	Administrador	6/1/2024	6/1/2024	
historial de ventas	Paul Hidalgo	Vendedores /Administrado	6/1/2024	6/1/2024	

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- A través de la información recolectada por las entrevistas y fichas de observación en la microempresa ZOMI, se determinó que los procesos internos relacionados con la gestión financiera no son adecuados, esto teniendo en cuenta que no manejaban ningún sistema digital para gestionar los procesos. Dando como resultado la obtención de los requisitos y funcionalidades para el desarrollo del sistema web.
- La arquitectura serverless representa un avance significativo en el diseño y desarrollo de sistemas de software, ofreciendo una serie de beneficios sustanciales que impactan positivamente en la eficiencia operativa y la escalabilidad.
- Uno de los aspectos más destacados de la arquitectura serverless es su capacidad para liberar a los desarrolladores y equipos de operaciones la responsabilidad directa de administrar la infraestructura subyacente. Además, es un modelo de pago por uso, la microempresa puede optimizar sus gastos alineándolos con la demanda real, evitando los costos asociados con la infraestructura tradicional que permanece activa incluso durante períodos de baja utilización.
- La metodología XP fue elegida para gestionar las actividades de este proyecto permitiendo obtener una mejor organización durante los procesos de codificación al dar un rol a cada participante, además esta metodología permite el uso de técnicas para desarrollar interacciones de manera progresiva de tal manera que disminuyan errores en cada módulo.
- La implementación del sistema web ha permitido mejorar la organización y almacenamiento de los datos de compras, ventas e inventarios esto con el fin de mejorar la calidad de sus servicios lo cual permite que su información sea más precisa y ayude con la toma decisiones.

4.2 Recomendaciones

- Se recomienda utilizar la aplicación web en cada proceso para poder almacenar los datos de manera organizada dentro de la microempresa ZOMI, esto con el fin de obtener información real en su gestión financiera y evitar que sus registros se realicen en hojas de papel como lo venían haciendo, lo cual no ha tenido un buen resultado.
- Se recomienda la adopción de la arquitectura serverless en el desarrollo de páginas web especialmente cuando se busca optimizar la eficiencia operativa, mejorar la escalabilidad y reducir la complejidad en la gestión de la infraestructura.
- Es recomendable la integración de la arquitectura serverless en las microempresas ya que permite a los desarrolladores y equipos de operaciones mejorar el desarrollo web, y obtener beneficios de agilidad, rentabilidad y mantenimiento simplificado, además es un modelo de pago por uso es decir que optimizaría sus gastos.
- Se aconseja el uso de la metodología XP, cuando se trate de desarrollar un sistema web ya que este le permite gestionar las actividades de manera más ordenada y eficiente con la finalidad de tener una mejor visión de cada proceso, es decir que cada participante sabe que debe hacer en el sistema web.
- Es recomendable brindar capacitación al personal que va a utilizar el sistema web con la finalidad de facilitar el uso y evitar confusiones al momento de utilizar el sistema web.

REFERENCIAS BIBLIOGRÁFICAS

- [1] M. A. Cruz Pérez, M. A. Pozo Vinueza, A. F. Andino Jaramillo, and A. D. Arias Parra, “Las Tecnologías de la Información y la Comunicación (TIC) como forma investigativa interdisciplinaria con un enfoque intercultural para el proceso de formación de los estudiantes,” *e-Ciencias de la Información*, Nov. 2018, doi: 10.15517/eci.v1i1.33052.
- [2] C. Económicas Empresariales Artículo de Investigación, C. I. Hernán Mero-Suárez, and J. Juleidy Muñiz-Tomalá III, “Use of Technological Tools in the Development of Small and Medium Enterprises in Manabí Uso de ferramentas tecnológicas no desenvolvimento de pequenas e médias empresas em Manabí,” *Febrero Especial*, vol. 8, no. 1, pp. 330–345, 2022, doi: 10.23857/dc.v8i1.2574.
- [3] E. M. Capuz Balladares and X. A. Morales Urrutia, “GESTIÓN ADMINISTRATIVA FINANCIERA Y SU INCIDENCIA EN LA RENTABILIDAD DE FERRETERÍA AYALCA,” *Universidad Técnica de Ambato. Dirección de Posgrado. Maestría en Dirección de Empresas*, 2022, [Online]. Available: <https://repositorio.uta.edu.ec/jspui/handle/123456789/36670>
- [4] C. DE Tecnologías La Información Proyecto De Titulación Previo A La Obtención Del Título De and I. Martha Romero Castro, “UNIVERSIDAD ESTATAL DEL SUR DE MANABÍ FACULTAD DE CIENCIAS TÉCNICAS TUTORA,” 2022.
- [5] F. Esneider and M. Romero, “Diseño e implementación de sistema de inventarios para el almacén de pinturas y ferretería Ferrecolor,” 2019. [Online]. Available: <http://hdl.handle.net/20.500.12494/8557>
- [6] M. A. Maisanche Montachana and H. Y. Chicaiza Recalde, “La tecnología como factor determinante para el mejoramiento de la calidad en empresas de servicios de la provincia de Tungurahua,” *Universidad Técnica de Ambato. Facultad de Ciencias Administrativas. Carrera de Organización de Empresas*, 2020, [Online]. Available: <https://repositorio.uta.edu.ec/jspui/handle/123456789/31476>
- [7] E. Bahit, “Arquitectura de Sistemas Informáticos,” Mar. 2021, doi: 10.13140/RG.2.2.23508.32643.
- [8] E. Politecnica Superior, S. Sánchez Prado Tutora, and A. María González Marcos, “UNIVERSIDAD AUTONOMA DE MADRID”.
- [9] Luiza Petrosyan, “Arquitecturas y Modelos Economicos de Aplicaciones Serverless,” *Universitat Politècnica de València*, 2020, [Online]. Available: <http://hdl.handle.net/10251/153763>
- [10] Samu Uunonen, “Backend as a service in web development,” 2023.

- [11] Javier Encinas Cortés, “UNIVERSIDAD AUTONOMA DE MADRID TRABAJO FIN DE GRADO,” 2020.
- [12] S. Bernardi, “Sistemas de información para la dirección: Un enfoque guiado por un caso de estudio”, Accessed: Feb. 04, 2024. [Online]. Available: <http://www.cud.unizar.es>
- [13] M. J. Suárez-Espinar, “Gestión Empresarial: una paradigma del siglo XXI,” *Revista Científica FIPCAEC (Fomento de la investigación y publicación en Ciencias Administrativas, Económicas y Contables)*. ISSN : 2588-090X . Polo de Capacitación, Investigación y Publicación (POCAIP), vol. 3, no. 8, pp. 44–64, Jul. 2018, doi: 10.23857/fipcaec.v3i8.57.
- [14] M. Fajardo Ortiz and C. Soto González, “Gestión Financiera Empresarial,” *Machala: Universidad Técnica de Machala*, 2018, [Online]. Available: <http://repositorio.utmachala.edu.ec/handle/48000/12487>
- [15] Paul Beynon-Davies, “Sistemas de información: Introducción a la informática en las organizaciones,” Editorial Reverte.
- [16] J. Alejandro, E. Cano, A. Yesid, O. González Marín, and J. C. Farfán Rodríguez, “SISTEMA DE CONTROL DE INVENTARIOS PARA LA EMPRESA COMERCIALIZADORA LITURGICA Y EDITORIAL LTDA,” 2018.
- [17] “AWS | Lambda - Gestión de recursos informáticos.” Accessed: Nov. 15, 2023. [Online]. Available: https://aws.amazon.com/es/lambda/?did=ft_card&trk=ft_card
- [18] “AWS | Servicio de base de datos gestionada NoSQL (DynamoDB).” Accessed: Nov. 15, 2023. [Online]. Available: https://aws.amazon.com/es/dynamodb/?did=ft_card&trk=ft_card
- [19] “Amazon API Gateway | API Management | Amazon Web Services.” Accessed: Nov. 15, 2023. [Online]. Available: https://aws.amazon.com/es/api-gateway/?did=ft_card&trk=ft_card
- [20] “¿Por qué Astro? | Docs.” Accessed: Jan. 09, 2024. [Online]. Available: <https://docs.astro.build/es/concepts/why-astro/>
- [21] “Getting Started – React.” Accessed: Jan. 09, 2024. [Online]. Available: <https://legacy.reactjs.org/docs/getting-started.html>
- [22] “Angular - Introducción a la Documentación de Angular.” Accessed: Jan. 09, 2024. [Online]. Available: <https://docs.angular.lat/docs>
- [23] J. Camilo Salazar, Á. Tovar, J. Carlos Linares, A. Lozano, and L. Valbuena, “Scrum contra XP: similitudes y diferencias,” *TIA*, vol. 6, no. 2, pp. 29–37, 2018, [Online]. Available: <https://revistas.udistrital.edu.co/ojs/index.php/tia>
- [24] M. C. Saúl González Campos, M. C. Luis, and F. Fernández Martínez, “Culcyt//Software Programación Extrema: Prácticas, Aceptación y Controversia.”

- [25] L. Castellano Lendínez, “Kanban. Metodología para aumentar la eficiencia de los procesos. 3C Tecnología. Glosas de innovación aplicadas a la pyme,” vol. 8, no. 1, pp. 30–41, 2019, doi: 10.17993/3ctecno/2019.
- [26] K. Schwaber, J. Sutherland, and L. G. Definitiva, “La Guía Scrum,” 2020.
- [27] D. Óscar, F. Aranda, D. César, and V. Rodríguez, “Aplicación de metodología Scrum para el desarrollo de una API web de gestión médica con MEAN Stack,” 2019.
- [28] J. L. Arciniegas, V. Fernández, A. Hormiga, A. Tulande, F. A. Urbano, and C. A. Collazos, “Proceso de requerimiento y análisis para la definición de la arquitectura desde la perspectiva de usabilidad para el desarrollo de aplicaciones en la Web,” *Revista Avances en Sistemas e Informática*, vol. 6, no. 2, pp. 205–210, 2009, Accessed: Nov. 18, 2023. [Online]. Available: <https://www.redalyc.org/articulo.oa?id=133113598023>

ANEXOS

Anexo A. Entrevistas y Matriz de observación

Guía 1 orientada al gerente.

ENTREVISTA DIRIGIDA AL GERENTE PROPIETARIO DE LA MICROEMPRESA "ZOMI"

Sistema de gestión financiera

Fecha: __/__/____

Entrevistador: Paul Hidalgo

Entrevistado: Michelle Hidalgo

Cargo: Gerente

Objetivo:

Analizar los procesos de control de compra, venta e inventario de productos la Microempresa "ZOMI".

Introducción:

Entrevistador: Buen día, agradecemos tu participación en esta entrevista que tiene como objetivo obtener información valiosa para la implementación exitosa de nuestro nuevo sistema web de gestión financiera en "ZOMI". Queremos escuchar tus opiniones y experiencias para garantizar que el sistema se adapte de la mejor manera a nuestras operaciones diarias. ¿Estás listo para comenzar?

Preguntas:

Pregunta 1: ¿Se lleva un registro detallado de las compras realizadas dentro de la microempresa "ZOMI"?

Respuesta:

Observación:

Pregunta 2: ¿Cuáles son las principales dificultades que enfrenta la microempresa "ZOMI" en relación con la gestión de inventarios?

Respuesta:

Observación:

Pregunta 3: ¿Cuáles son los principales desafíos que enfrenta actualmente en el proceso de ventas de sus productos?

Respuesta:

Observación:

Pregunta 4: ¿Podría describir el proceso que sigue la microempresa para evaluar y analizar las ganancias generadas durante un mes?

Respuesta:

Observación:

Pregunta 5: ¿Considera usted que al aplicar un sistema web dentro de la microempresa "ZOMI" mejoraría la gestión financiera? Si es así, ¿podría mencionar algunos aspectos específicos?

Respuesta:

Observación:

Pregunta 6: ¿Qué características y funcionalidades específicas le gustaría que tuviera un sistema web de gestión financiera dentro microempresa "ZOMI"?

Por ejemplo:

- Que permita registrar fácilmente todos los ingresos y gastos de la empresa.
- Tenga la capacidad de generar informes financieros precisos y detallados.
- Que permita llevar un seguimiento del inventario con alertas de reabastecimiento

Respuesta:

Observación:

Pregunta 7: ¿Cuál cree que son los posibles desafíos que podrían surgir durante la implementación de un sistema web para la gestión financiera en la microempresa "ZOMI"?

Respuesta:

Observación:

Pregunta 8: ¿A trabajado alguna vez con un sistema WEB? Si es así, ¿puede mencionar uno de ellos?

Respuesta:

Observación:

Pregunta 9: "Si se implantara un sistema web para la gestión financiera en la microempresa 'ZOMI', ¿cómo crees que este cambio afectaría tu día a día y tu forma de realizar las tareas relacionadas con las ventas y la gestión de productos?"

Respuesta:

Observación:

Conclusión

Guía 2 orientada al personal de ventas.

ENTREVISTA DIRIGIDA AL PERSONAL VENTAS DE LA MICROEMPRESA "ZOMI"

Sistema de gestión financiera

Fecha: __/__/____

Entrevistador: Paul Hidalgo

Entrevistado: Evelyn Barreno

Cargo: ventas

Objetivo:

Analizar los procesos de control de compra, venta e inventario de productos la Microempresa "ZOMI".

Introducción:

Entrevistador: Buen día, agradecemos tu participación en esta entrevista que tiene como objetivo obtener información valiosa para la implementación exitosa de nuestro nuevo sistema web de gestión financiera en "ZOMI". Queremos escuchar tus opiniones y experiencias para garantizar que el sistema se adapte de la mejor manera a nuestras operaciones diarias. ¿Estás listo para comenzar?

Preguntas:

Pregunta 1: ¿Cuál es el proceso actual de venta de la microempresa "ZOMI", desde la recepción de un pedido hasta la entrega del producto al cliente?

Respuesta:

Observación:

Pregunta 2: ¿Has enfrentado desafíos en la consulta de cantidades y/o precios de los productos durante la realización de una venta? Si es así, ¿podrías describir algunos ejemplos específicos?

Respuesta:

Observación:

Pregunta 3: ¿De qué manera se lleva a cabo el registro de inventario de productos actualmente en la microempresa "ZOMI"? Respuesta:

Observación:

Pregunta 4: ¿cuáles son las expectativas o necesidades que crees que este sistema debería cumplir para mejorar el desempeño financiero de la microempresa "ZOMI"?

Respuesta:

Observación:

Pregunta 5: En su opinión, ¿cuáles son los posibles beneficios que este sistema puede aportar a la eficiencia y productividad en ventas de la microempresa “ZOMI”?

Respuesta:

Observación:

Pregunta 6: ¿Qué tipo de información financiera consideras crucial y que debería estar fácilmente accesible para el personal de ventas en este sistema?

Respuesta:

Observación:

Pregunta 7: Desde su rol en ventas, ¿visualiza algún desafío potencial que podríamos enfrentar durante la implementación y adaptación a este nuevo sistema?

Respuesta:

Observación:

Pregunta 8: ¿En qué áreas cree que este sistema podría automatizar tareas manuales en ventas y ayudar a ahorrar tiempo en relación con las operaciones financieras?

Respuesta:

Observación:

Conclusión

Guía 3 entrevista orientada al personal de producción.

ENTREVISTA DIRIGIDA AL PERSONAL DE PRODUCCIÓN DE LA
MICROEMPRESA “ZOMI”

Sistema de gestión financiera

Fecha: __/__/____

Entrevistador: Paul Hidalgo

Entrevistado: Lida Pérez

Cargo: producción

Objetivo:

Analizar la gestión de la producción de la Microempresa “ZOMI” con el fin de identificar áreas de mejora

Introducción:

Entrevistador: Buen día. Agradecemos tu participación en esta entrevista que tiene como objetivo obtener información valiosa para la implementación exitosa de nuestro nuevo sistema web de gestión financiera en "ZOMI". Queremos escuchar tus opiniones y experiencias para garantizar que el sistema se adapte de la mejor manera a nuestras operaciones diarias. ¿Estás listo para comenzar?

Preguntas:

1. ¿Puede describir brevemente su papel y responsabilidades como persona encargada de producción en ZOMI?

Respuesta:

Observación:

2. ¿Cuál es el criterio que utilizan actualmente para determinar las cantidades de productos a producir?

Respuesta:

Observación:

3. ¿Cuáles son los principales desafíos que enfrenta el área de producción en ZOMI en la actualidad?

Respuesta:

Observación:

4. ¿Actualmente el registro de la producción se realiza por medio de un software o manualmente?

Respuesta:

Observación:

5. ¿Qué tipo de información sería importante conocer en el área de producción para mejorar la gestión de productos?

Respuesta:

Observación:

6. ¿Cómo cree que la implementación de un sistema web podría beneficiar a ZOMI en términos de producción?

Respuesta:

Observación:

7. Desde tu rol en producción, ¿visualizas algún desafío al implantar este nuevo sistema?

Respuesta:

Observación:

Conclusión:

Matriz 1: Orientado al proceso de ventas

La matriz de observación fue realizada en conjunto con la persona de ventas para la recolección de información sobre las actividades de ventas.

Escalas:

1 (Insatisfactorio): Indica que la interacción o el servicio fue de muy baja calidad y no cumplió con las expectativas del cliente ni de la empresa.

2 (Necesita Mejora): Sugiere que hubo áreas de mejora evidentes en la interacción o el servicio y no fue satisfactorio en su totalidad.

3 (Aceptable): Significa que el servicio o la interacción fueron razonablemente aceptables, pero hay margen para mejorar.

4 (Bueno): Indica que el servicio o la interacción fueron buenos y cumplieron con las expectativas en la mayoría de los aspectos.

5 (Excelente): Representa un servicio o una interacción excepcionales que superaron las expectativas y dejaron al cliente satisfecho

Tabla 83. Matriz de observación de ventas

NO	CATEGORÍA DE OBSERVACIÓN	DESCRIPCIÓN DE LA SITUACIÓN	ACCIONES Y PROCESOS OBSERVADOS	ESCALA (1-5)	COMENTARIOS /NOTAS
1	Toma de pedidos	Proceso de toma de pedidos corto plazo			
		Proceso de toma de pedidos al instante			
2	Disponibilidad de productos	disponibilidad de productos			
		Cantidad de productos			
		No hay suficientes productos en stock.			
3	Manejo de Pago	Proceso de pago en ventas bajo pedido.			
		Proceso de pago de ventas al instante.			
4	Facturación y Documentación	Generación de factura			
		Emisión del Documento			
CONCLUSIÓN:					

Matriz 2: Orientado al proceso de adquisición de productos

La matriz de observación fue realizada en conjunto con la gerente de la microempresa para la recolección de información sobre las actividades de adquisición de productos.

Tabla 84. Matriz de observación de compras

NO	DETALLES DE LA ADQUISICIÓN	PROCESO DE ADQUISICIÓN	COMENTARIOS /NOTAS
1	Revisión de productos	Selección de productos faltantes	
		Cantidad de productos	
2	Proveedor(es)	Selección de Proveedor	
		Tipo de proveedor	
3	Productos Adquiridos	Negociación y Acuerdo de Compra	
4	Manejo de pago	Proceso de Pago	
5	Facturación	Recibo de la compra	
CONCLUSIÓN:			

Anexo B. Manual de usuario

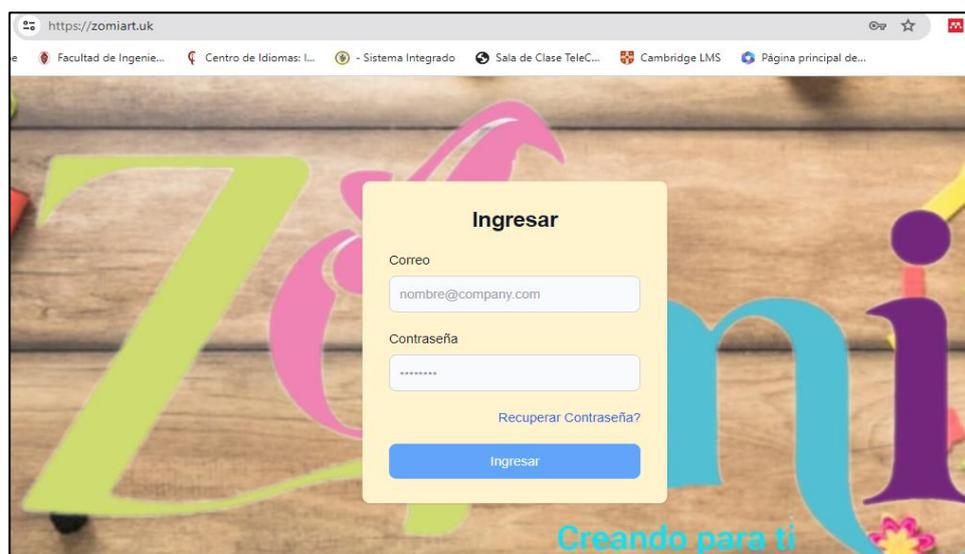
Enlace: <https://zomiart.uk/>

El manual se encuentra estructurado de la siguiente forma:

1. Ingreso al Sistema
2. Restablecer Contraseña
3. Página Principal
4. Módulo de ventas
5. Módulo de inventarios
6. Módulo de Compras
7. Informes
8. Historias de facturas
9. Usuarios
10. Finalizar Sesión

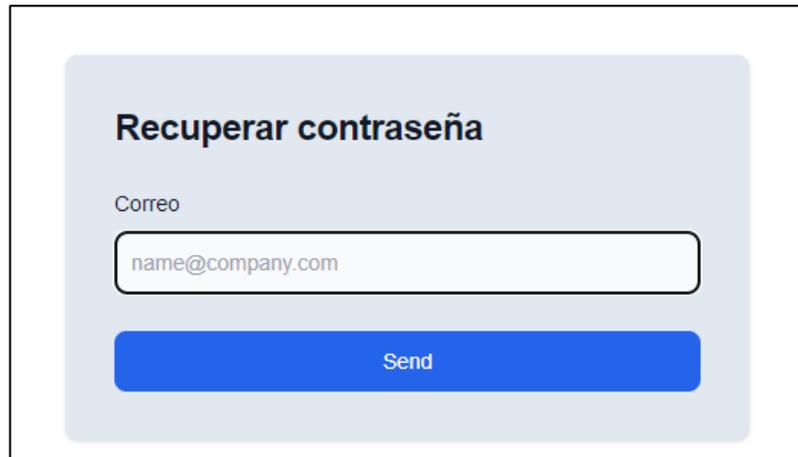
- ***Ingreso al sistema***

Para ingresar al sistema, en la interfaz de inicio de sesión solicita el correo y clave del usuario registrado.



- **Restablecer Contraseña**

En caso de que el usuario olvide la contraseña, tiene la opción de recuperar por medio del correo, en la interfaz de inicio aparece con letra azules y con un click redirige a la página de recuperar contraseña.



- **Página principal**

El sistema web tiene dos paginas principales. La primera es para usuarios con el rol de administrador y la segunda el para usuarios con el rol de vendedor.

- **Administrador**



El usuario administrador tiene acceso a todas las opciones del sistemas web.

- Ventas



El usuario de ventas solo tiene acceso a tres módulos: ventas, inventarios y facturas.

- **Módulo de ventas (todos)**
 - **Primera sección**

El usuario tiene la opción de filtrar los productos por categorías, siguiente escoger el producto de una lista, escoger la cantidad y ultimo el botón agregar.
 - **Segunda sección**

Una que se agrega el producto se visualiza la información del producto seleccionado, además, el usuario tiene la opción de incrementar, disminuir la cantidad o eliminar de la lista el producto
 - **Tercero Botón guardar**

Finalizada la selección de productos se guarda los productos para ser visualizados en otra sección.

Zomi
Usuario: (Ventas)

Ventas
Inventario
Facturas
Salir

1 **Categoria:** Todos **Cantidad:** 1 Agregar

CODIGO	NOMBRE	DESCRIPCION	CATEGORIA	PRECIO C/U	CANTIDAD	SUBTOTAL			
mSPI007	jirafa	porta encienso figura jirafa	masa pan	1.25	1	1.25	+	-	🗑️
mSpto002	torito	porta encienso figura torito	masa pan	1.25	1	1.25	+	-	🗑️
TOTAL						2.50\$			

3 Guardar

- **Ingreso datos usuarios y confirmar la venta**

Finalmente se visualiza la pantalla de confirmar venta, donde el usuario ingresa los datos del cliente verifica que todo este correcto y acepta la venta

Confirnar la venta X

Identificacion:
#Venta:

Nombres:
Telefono:

Correo:
Direccion:

Nombre	Descripcion	Precio	Cantidad	SubTotal
burrito	porta encienso figura burrito	1.00	1	1.00
jirafa	porta encienso figura jirafa	1.25	1	1.25
perrito	porta encienso figura perrito	1.25	1	1.25
sapito	porta encienso figura sapito	1.25	1	1.25

Total 4.75\$
Aceptar

- **Módulo de inventarios (administrador)**

La opción de inventarios el usuario visualiza la lista de productos con la opción de seleccionar el producto y se despliega la información completa del mismo, además tiene las opciones de: realizar búsqueda filtradas por categorías y la opción de ingresar un nuevo producto.

Fecha	Producto	Categoría	Ingreso	Cantidad	Precio Venta	Precio Comp.
15/01/2024	burrito	masa pan	produccion	16	1.00\$	0.50\$
15/01/2024	perrito	masa pan	produccion	15	1.25\$	0.50\$
15/01/2024	unicornio	masa pan	produccion	14	1.25\$	0.50\$
15/01/2024	sapito	masa pan	produccion	14	1.25\$	0.50\$
15/01/2024	jirafa	masa pan	produccion	12	1.25\$	0.50\$
14/01/2024	cholitas sentadas	masa pan	compras	12	2.00\$	1.75\$
14/01/2024	diablo huma	llaveros	compras	8	1.25\$	0.75\$
14/01/2024	torito	masa pan	produccion	6	1.25\$	0.50\$
14/01/2024	vaquita	masa pan	produccion	5	1.25\$	0.50\$

- **Formulario nuevo producto**

En la opción de nuevo producto, se visualiza el formulario para el ingreso de nuevos productos y al finalizar el ingreso de los datos se confirma el ingreso nuevo y actualiza la lista de productos.



Formulario de Datos Producto con los siguientes campos:

- Codigo:
- Categoria:
- Nombre:
- Precio Compra:
- Catidad:
- Precio Venta:
- Descripcion:

Botón:

- **Información completa del producto**

En la información completa del producto el usuario tiene la opción de modificar los campos excepto campos importantes, en el campo de “cantidad de ingreso” el usuario puede ingresar nuevas cantidades para para abastecer el inventario si la necesidad de ingresar todo el producto nuevamente, finalmente la opción de eliminar el producto

Información del producto X

Código:
mspbu0012

Fecha Ingreso:
15/01/2024 00:46

Nombre:
burrito

Cantidad actual: 16 **Cantidad de ingreso:** 0

Categoría: masa pan **tipo Ingreso:** producción

Precio venta: 1.00 **Precio compra:** 0.50

Proveedor:

Descripción:
porta encienso figura burrito

Eliminar
Aceptar

- **Módulo de Compras (administrador)**

En la opción de compras el usuario visualiza una lista del historial de todas las compras realizadas, cada una tiene la opción de seleccionar y se despliega su información completa, además, tiene las opciones de: búsquedas filtradas por categorías y la opción de ingresar nueva compra.

Zomi
Usuario: paul (admin)

Ventas Inventario Compras Informes Facturas Usuarios Salir

Categoría: Todos ▼ Nueva compra

Fecha	Producto	Cantidad	Precio Venta	Precio Comp.	Categoría	Proveedor	Total
14/01/2024	cholitas sentadas	12	2.00\$	1.75\$	masa pan	Pedro Sanches	21.00
14/01/2024	diablo huma	12	1.25\$	0.75\$	llaveros	Mario Tores	9.00

- **Nueva Compra**

La opción de nueva compra se despliega un formulario, el usuario tiene la opción de elegir proveedor de los que ya se encuentren registrados o puede ingresar uno nuevo, finalmente , completado los campos con la opción agregar, el sistema notifica del nuevo ingreso y se actualiza la lista de compras.

The screenshot shows a form titled "Datos Compra" with a close button (X) in the top right corner. The form contains the following elements:

- A dropdown menu labeled "Elegir Proveedor" with a downward arrow.
- A button labeled "Nuevo Proveedor" in the top right.
- Input fields for "Codigo:" and "Categoria:".
- Input fields for "Producto:" and "Precio Compra c/u:".
- Input fields for "Cantidad:" and "Precio Venta:".
- A large input field for "Descripcion:".
- A "Total:" label followed by a small input field.
- A blue button labeled "Agregar" in the bottom right corner.

- **Nuevo proveedor**

La opción nuevo proveedor se despliega el formulario para ingresar los datos.

The screenshot shows a form titled "Datos Proveedor" with a close button (X) in the top right corner. The form contains the following elements:

- An input field for "Nombre Prov.:".
- An input field for "Direccion:".
- An input field for "Telefono:".
- An input field for "Correo:" with the text "Opcional" below it.
- A blue button labeled "Aceptar" at the bottom center.

- **Información completa de la compra**

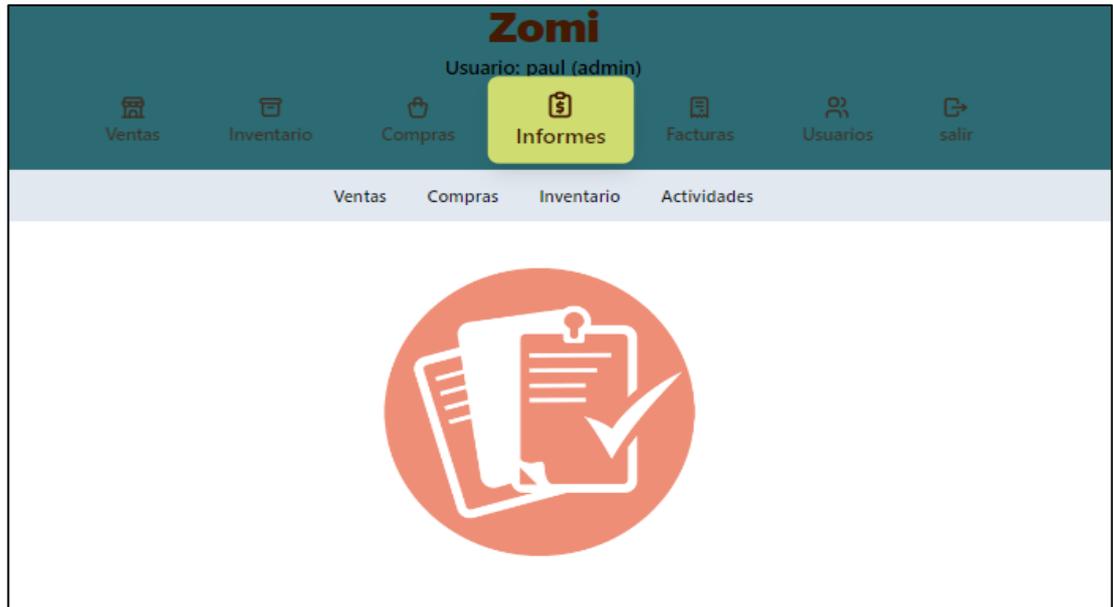
Cada compra de la lista tiene su información completa, el usuario tiene la opción de modificar ciertos campos y eliminar la compra.

Información Compra

Código:	Producto:
<input type="text" value="mspch006"/>	<input type="text" value="cholitas sentadas"/>
Cantidad:	Categoría:
<input type="text" value="12"/>	<input type="text" value="masa pan"/>
Precio venta c/u:	Precio Compra c/u:
<input type="text" value="2.00"/>	<input type="text" value="1.75"/>
Descripción:	
<input type="text" value="pareja de cholitas sentadas"/>	
Proveedor:	Teléfono:
<input type="text" value="Pedro Sanches"/>	<input type="text" value="0986857456"/>
Dirección:	Correo:
<input type="text" value="Ambato"/>	<input type="text" value="pedro@gmail.com"/>
Fecha:	Total:
<input type="text" value="14/01/2024 20:33"/>	<input type="text" value="21.00"/>
<input type="button" value="Eliminar"/> <input type="button" value="Enviar"/>	

- **Informes (administrador)**

En la opción de compras el usuario visualiza las opciones de: “ventas”, “compras”, “inventario” y “actividades”



- **Ventas**

La opción de ventas el usuario visualiza toda la información referente a las ventas, con la opción de buscar por fechas.

Ventas Compras Inventario Actividades				
VENTAS		Total todas las ventas: 27.00\$		#ventas: 11
	informacion de las ventas por Dia	Cantidad de productos por mes	Cantidad de productos Vendidos	
Seleccione un mes : enero de 2024 <input type="checkbox"/> Total venta del mes: 0.00\$ Numero ventas del mes: 0	unicomico: 1	torito: 7 diablo huma: 7 vaquita: 6 jirafa: 2 unicomio: 1	torito: 7 diablo huma: 7 vaquita: 6 jirafa: 2 unicomio: 1	
Seleccione una fecha: 01/15/2024 <input type="checkbox"/> Total Diario: 1.25\$ Numero de ventas del dia: 1				
	Total : 1	Total : 23	Total : 23	

- **Compras**

La opción de compras de igual manera se visualiza toda la información de las compras, con la opción de buscar por fechas

Ventas Compras Inventario Actividades				
COMPRAS		Total todas las compras: 30		#compras: 2
Selecciona un mes : enero de 2024 <input type="text"/> Total compras del mes: 30.00\$ Numero compras del mes: 2	informacion compras por Dia	Cantidad de compras por mes	Cantidad de prodctos comprados	
		chollitas sentadas: 12	chollitas sentadas:	12
		diablo huma: 12	diablo huma:	12
Selecciona una fecha: 01/15/2024 <input type="text"/> Total Diario: 0.00\$ Numero de ventas del dia: 0				
	Total :	0	Total :	24
			Total :	24

- **Inventario**

La opción de inventario el usuario visualiza total de productos con sus cantidades y precio, además, están divididas en productos ingresados de producción y productos ingresados por compras.

Ventas Compras Inventario Actividades						
Inventario		Total productos: 9		Cantidad Total: 102	Total: 132.50\$	
Inventario - Producción			Inventario - Compras			
Nombre	Precio (USD)	Cantidad	Nombre	Precio (USD)	Cantidad	
burrito	1.00	16	chollitas sentadas	2.00	12	
jirafa	1.25	12	diablo huma	1.25	8	
vaquita	1.25	5				
perrito	1.25	15				
torito	1.25	6				
sapito	1.25	14				
unicornio	1.25	14				
Total Cantidad:		82	Total Cantidad:		20	
Total Precio:		98.50	Total Precio:		34.00	

- **Archivo “pdf” de informe de inventario**

El usuario tiene la opción de guardar o imprimir un archivo “pdf” la información del inventario.

Informe Inventario					
Total productos: 9					
Cantidad Total: 102					
Total: 132.50\$					
Producción			Compras		
Nombre	Precio (USD)	Cantidad	Nombre	Precio (USD)	Cantidad
burrito	1.00	16	cholitas sentadas	2.00	12
jirafa	1.25	12	diablo huma	1.25	8
vaquita	1.25	5			
perrito	1.25	15			
torito	1.25	6			
sapito	1.25	14			
unicornio	1.25	14			
Total Cantidad: 82			Total Cantidad: 20		
Total Precio: 98.50			Total Precio: 34.00		

- **Actividades**

La opción de actividades el usuario visualiza un registro de todas las actividades por ejemplo cuando un usuario realiza una venta o se ingresa un nuevo producto así mismo si se ingresa una compra, etc.

	Ventas	Compras	Inventario	Actividades		
Actividades						
Nombre	Tipo	Código	Origen	Fecha	Precio c/u	Cantidad
sapito	salida	m��a008	ventas	15/01/2024 04:46	1.25	1
burrito	salida	m��bu0012	ventas	15/01/2024 04:46	1.00	1
perrito	salida	m��pe009	ventas	15/01/2024 04:46	1.25	1
jirafa	salida	m��ji007	ventas	15/01/2024 04:46	1.25	1
perrito	salida	m��pe009	ventas	15/01/2024 04:46	1.25	1
torito	salida	m��to002	ventas	15/01/2024 01:35	1.25	1
unicornio	salida	m��ni0010	ventas	15/01/2024 00:50	1.25	1
burrito	ingreso	m��bu0012	produccion	15/01/2024 00:46	1.00	16
unicornio	ingreso	m��ni0010	produccion	15/01/2024 00:39	1.25	15
perrito	ingreso	m��pe009	produccion	15/01/2024 00:39	1.25	15
sapito	ingreso	m��sa008	produccion	15/01/2024 00:33	1.25	14
jirafa	ingreso	m��ji007	produccion	15/01/2024 00:32	1.25	12
vaquita	salida	m��va001	ventas	14/01/2024 22:19	1.25	3
diablo huma	salida	llvdh005	ventas	14/01/2024 20:45	1.25	3

- **Historial de facturas (todos)**

La opción de “facturas” los usuarios visualizan una lista de todas las ventas realizadas con la opción de seleccionar para visualizar los detalles de la venta, además , tiene la opción de buscar.

Zomi
Usuario: paul (admin)

Ventas
Inventario
Compras
Informes
Facturas
Usuarios
Salir

Buscar Nota de venta

Fecha	Cliente	Num. Factura	Productos	Total
15/01/2024 00:50	Paola Ramos	111	1	1.25
14/01/2024 22:19	Paul Hidalgo	110	1	3.75
14/01/2024 20:45	Paola Ramos	109	1	3.75
14/01/2024 20:17	Paola Ramos	108	1	6.25
14/01/2024 19:36	Paola Ramos	107	2	2.50
14/01/2024 19:35	Paul Hidalgo	106	1	1.25
14/01/2024 19:29	Paola Ramos	105	1	1.25
14/01/2024 19:21	Paola Ramos	104	1	1.00
14/01/2024 18:46	Paola Ramos	103	1	1.00
14/01/2024 18:46	Paola Ramos	102	2	2.00
14/01/2024 18:43	Paola Ramos	101	3	3.00

- **Información de la venta**

Los usuarios tienen la opción de imprimir la factura si el cliente los desea o de eliminar en caso sea necesario.

Información Venta 🔒 X

Identificación: # 0000109

Nombres: Telefono:

Correo: Direccion:

Codigo	Descripcion	Precio	Cantidad	SubTotal
llvdh005	llavero de diablo huma	1.25	3	3.75

Total 3.75

Eliminar

- **Factura**

Formato de la factura con el logo de la microempresa.



Factura:	# 0000109
HIDALGO SILVA MICHELLE MARISOL	
Dirección Matriz : Calle: ESPEJO Número: 89 Intersección: JUAN BENIGNO VELA	
OBLIGADO A LLEVAR CONTABILIDAD : NO	
CONTRIBUYENTE NEGOCIO POPULAR - RÉGIMEN RIMPE	

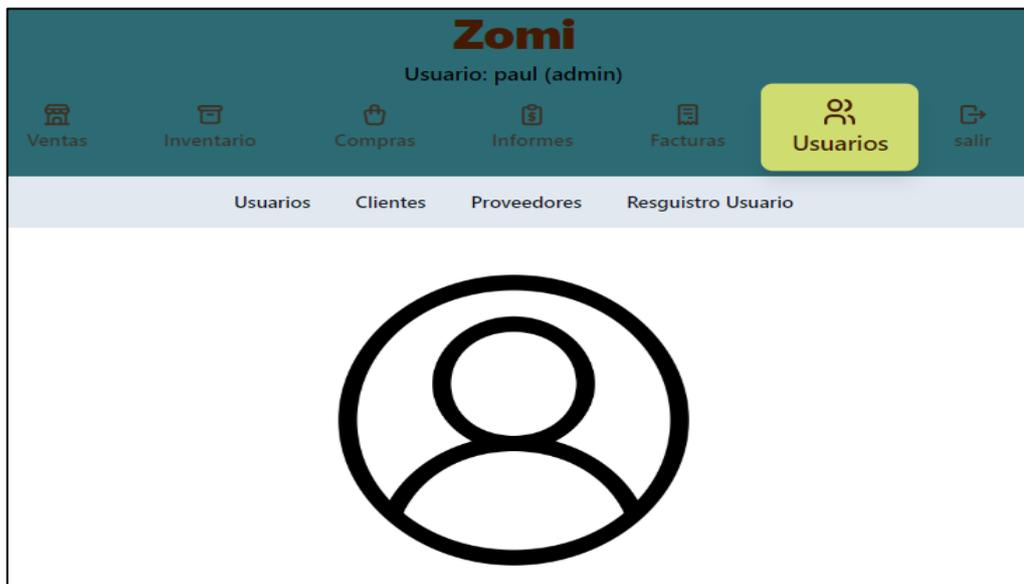
Nombres y Apellidos: Paola Ramos	
Identificación: 1804261905	
Fecha: 14/01/2024	Telefono: 0998201574
Dirección: Ambato	Email: jesy27pao@hotmail.com

Codigo	Descripcion	Precio	Cantidad	SubTotal
llvdh005	llavero de diablo huma	1.25	3	3.75

TOTAL : 3.75

- **Usuarios (administrador)**

La opción de usuarios, se visualiza cuatro opciones una para registrar y las demás para visualizar una lista de casa tipo de usuario.



- **Usuarios del sistema**

Esta sección es para los usuarios que utilizan el sistema cada uno con su información.

Usuarios Clientes Proveedores Resgistro Usuario			
Usuario	Telefono	Cargo	Correo
juan	0987788750	Ventas	juperez883@gmail.com
Paola	0987785750	Ventas	gabriel94m26@gmail.com

- **Información de usuario**

El usuario tiene la opción de modificar nombre o teléfono, además, puede eliminar el usuario del sistema.

X

Información Usuario

Nombre: <input style="width: 90%; border: 1px solid gray;" type="text" value="juan"/>	Correo: <input style="width: 90%; border: 1px solid gray;" type="text" value="juperez883@gmail.com"/>
Telefono: <input style="width: 90%; border: 1px solid gray;" type="text" value="0987788750"/>	Cargo: <input style="width: 90%; border: 1px solid gray;" type="text" value="Ventas"/>

Eliminar

Aceptar

- **Clientes**

Esta sección es la lista de clientes registrados en el sistema cada uno con su información.

Usuarios <u>Clientes</u> Proveedores Resgistro Usuario				
Cedula	Cliente	direccion	Correo	Telefono
1804261905	Paola Ramos	Ambato	jesy27pao@hotmail.com	0998201574
0105747323	Paul Hidalgo	Cuenca	paulsb26@gmail.com	0984247560
999999999999	Consumidor Final	N/A	N/A	999999999

- **Información del cliente**

El usuario tiene la opción de modificar los datos del cliente excepto la cedula, además, puede eliminar cliente del sistema.

X

Información Cliente

Identificación:	Nombres:
<input type="text" value="1804261905"/>	<input type="text" value="Paola Ramos"/>
Telefono:	Correo:
<input type="text" value="0998201574"/>	<input type="text" value="jesy27pao@hotmail.com"/>
Direccion:	
<input type="text" value="Ambato"/>	

Eliminar

Aceptar

- **Proveedores**

Esta sección es la lista de los proveedores registrados en el sistema cada uno con su información.

Usuarios Clientes Proveedores Resgistro Usuario			
Proveedor	Direccion	Correo	Telefono
Pedro Sanches	Ambato	pedro@gmail.com	0986857456
Mario Tores	Quito	mario@gmail.com	0987788750

- **Información del proveedor**

Igualmente, el usuario tiene la opción de modificar y eliminar la información

X

Información Proveedor

Nombre:	Direccion:
<input type="text" value="Mario Tores"/>	<input type="text" value="Quito"/>
Telefono:	Correo:
<input type="text" value="0987788750"/>	<input type="text" value="mario@gmail.com"/>

Eliminar

Aceptar

- **Registro usuario**

La opción de registro usuario se despliega un formulario para ingresar los datos del nuevo usuario que usara el sistema .

Usuarios Clientes Proveedores Resgistro Usuario

Registro Usuario

Nombre
Nombre de usuario

Correo
nombre@company.com

Telefono
Telefono

Contraseña
.....

Cargo
Selecciona una opción

Registrar

- **Finalizar Sesión (todos)**

Finalmente, el usuario tiene la opción de cerrar sesión.

