

**DESARROLLO DE UNA APLICACIÓN WEB PARA EL MANEJO Y GESTIÓN DEL
INVENTARIO A TRAVES DEL FRAMEWORK VUEJS PARA LA TIENDA
“YALMECRIS”**

**Autor:
DIMAR DAVID GALVAN BAYONA**



Directora: Nini Johana Arias Herrera

Universidad de Pamplona

Facultad de Ingenierías y Arquitectura

Departamento de Ingenierías Electrónica, Eléctrica, Sistemas y Telecomunicaciones, EEST

Ingeniería de Sistemas

Villa del Rosario, Colombia.

2022

DEDICATORIA

Quiero dedicar el siguiente trabajo por encima de todo a Dios, a mis padres que sin su incondicional apoyo la realización de este logro seria solo más que un sueño, a mis amigos, compañeros y eventualmente colegas que han creído en mi incluso más de lo que yo lo he hecho, a aquellos docentes que en mi edad temprana me ayudaron a formar mi carácter y mi persona, pero en especial a mi pequeño hermano menor que sea en él una meta referente la cual desee seguir.

AGRADECIMIENTOS

Agradezco a Dios, que ha estado presente durante esos días de desfallecimiento, a mi padre Dimar Galvan Galvan por su esfuerzo que ha sido mayor que sus noches de descanso, por su confianza y apoyo que parece infinito, a mi madre Nancy Bayona Castro la persona más temperamental que conociera jamás, pero con un amor más grande que su genio, por su esfuerzo incondicional y su compañía cada mañana que me regalaba un desayuno y el café de su mirada, a mi hermano Dylan Jhoel Galvan Bayona quien pareciera ser los más diferente a mí pero es la persona que recompensa todo mi esfuerzo, un agradecimiento en especial a mi amigo y colega Luis David Celis Caicedo quien creyó más en mí en el momento en que menos confianza me tenía y a quien le debo incluso mucho más que un agradecimiento, a mis mejores amigos Freiban Sánchez, Samir Álvarez y Andrés Noriega a quienes su compañía ha sido un alivio a muchas de las malas situaciones de la vida, a mi primo John Maikol Rueda Bayona quien no pareciera ser un primo sino un hermano, consejero y un hombre en cual apoyarme, finalmente y no menos importante a todos aquellas personas que en menor o en gran medida han compartido conmigo un trocito del tiempo que caminamos juntos en esta vida.

RESUMEN

En el presente trabajo de grado se realizó una aplicación web con el objetivo de mejorar los procesos que se llevan de manera manual en el negocio YALMECRIS, este negocio está enfocado al abastecimiento de víveres, productos de aseo personal, productos básicos de la canasta familiar entre otros, en el que se busca implementar una solución tecnológica que permita mayor eficiencia en sus procesos de gestión y manejo de inventario, además del registro de las salidas de los productos. La realización de este proyecto se consiguió bajo la metodología ágil de desarrollo Scrum, en el cual inicialmente se realizó un diagnóstico del negocio para luego determinar los requisitos de la aplicación web y así continuar con los diagramas necesarios para la arquitectura del sistema, posterior se realizó la codificación de un frontend en cual fue desarrollado mediante la herramienta Vue.js con la cual se consumió los servicios que fueron expuestos por la Api del backend y así darle interactividad y funcionalidad al sistema, la realización del backend se hizo con Node.js, y además de esto, se utilizó de una base de datos relacional en PostgreSQL finalmente se le aplicaron pruebas unitarias para corroborar que la aplicación funcionaba correctamente .

Palabras claves: Aplicación web, Manejo de inventario, Metodología Scrum, Vue.js, Node.js.

ABSTRACT

In the present degree work, a web application was made with the objective of improving the processes that are carried out manually in the YALMECRIS business, this business is focused on the supply of food, personal hygiene products, basic products of the family basket among others, in which it seeks to implement a technological solution that allows greater efficiency in its management processes and inventory management, in addition to the registration of product outputs. The realization of this project was achieved under the agile Scrum development methodology, in which initially a business diagnosis was made to later determine the requirements of the web application and thus continue with the necessary diagrams for the system architecture, later it was carried out the coding of a frontend in which it was developed using the Vue.js tool with which the services that were exposed by the backend API were consumed and thus give interactivity and functionality to the system, the backend was made with Node.js, and in addition to this, a relational database in PostgreSQL was used, finally unit tests were applied to verify that the application worked correctly.

Keywords: Web application, Inventory management, Scrum Methodology, Vue.js, Node.js.

TABLA DE CONTENIDO

Contenido

CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA	26
1.1. PLANTEAMIENTO DEL PROBLEMA	26
1.2. FORMULACIÓN DEL PROBLEMA	27
1.3. JUSTIFICACIÓN	27
1.4. OBJETIVOS	28
1.4.1. Objetivo General	28
1.4.2. Objetivos Específicos	28
1.4.3. Acotaciones	28
CAPÍTULO 2. MARCO TEÓRICO	28
2.1. ANTECEDENTES	28
2.1.1. Antecedentes Internacional	29
2.1.2. Antecedentes Nacional	30
2.1.3. Antecedentes Regional	31
2.2. MARCO TEÓRICO	33
2.2.1. APLICACIÓN WEB	33
2.2.2. FRONTEND	33
2.2.3. HTML	34

2.2.12. API (Application Programming Interface)	41
2.2.13. POSTGRESQL	41
2.2.14. PRUEBAS UNITARIAS	42
2.2.15. METODOLOGÍAS DE DESARROLLO DE SOFTWARE	43
2.2.15.1 METODOLOGÍA	43
2.2.15.2 METODOLOGÍA TRADICIONAL	43
2.2.15.3 METODOLOGÍA AGILES	44
2.2.15.4 METODOLOGÍA ÁGIL SCRUM	45
2.3. MARCO LEGAL	47
2.3.1. Ley Estatutaria 1581 de 2012	47
2.3.2. Ley 1341 TIC en Colombia	47
2.3.3. Ley 1314 de 2009	48
2.3.4. Artículo 63 del Decreto de 2649 de 1993	48
CAPITULO 3. METODOLOGÍA PARA DESARROLLO AGIL	49
3.1.1. Product Backlog	49
3.1.2. Roles	50
3.1.3. Sprint Backlog	50
3.1.3.1 Sprint 0	51
3.1.3.2 Sprint 1	51
3.1.3.3 Sprint 2	52

3.1.3.4 Sprint 3	52
3.1.3.5 Sprint 4	53
3.1.3.6 Sprint 5	53
CAPITULO 4. DESARROLLO DEL SISTEMA	54
4.1. Diagnosticar el proceso actual que se tiene en la tienda, para el control de inventario.	54
4.2. Establecer los requisitos funcionales y no funcionales en base a los requerimientos que necesita el cliente para el desarrollo de la aplicación web.	55
4.2.1. Requisitos Funcionales	55
4.2.1.1 Requisito funcional 1	55
4.2.1.2 Requisito funcional 2	55
4.2.1.3 Requisito funcional 3	56
4.2.1.4 Requisito funcional 4	56
4.2.1.5 Requisito funcional 5	57
4.2.1.6 Requisito funcional 7	57
4.2.1.7 Requisito funcional 8	58
4.2.1.8 Requisito funcional 9	58
4.2.1.9 Requisito funcional 10	59
4.2.1.9 Requisito funcional 11	59
4.2.1.10 Requisito funcional 12	60
4.2.1.11 Requisito funcional 13	60

4.2.1.12 Requisito funcional 14 -----	61
4.2.1.13 Requisito funcional 15 -----	61
4.2.2 Requisitos no funcionales -----	62
4.2.2.1 Requisito no funcionales 1 -----	62
4.2.2.2 Requisito no funcionales 2 -----	62
4.2.2.3 Requisito no funcionales 3 -----	63
4.2.2.4 Requisito no funcionales 4 -----	63
4.3. Diseñar la aplicación web para la gestión de inventario de la tienda. -----	64
4.3.1. Diagrama de Casos de Uso -----	64
4.3.1.1 Diagrama de caso uso administrador -----	64
4.3.1.2 Diagrama de caso uso operador -----	65
4.2.2. Diagrama de Actividades -----	66
4.2.2.1 Diagrama de actividades inicio de sesión -----	66
4.2.2.2 Diagrama de actividades registrar operador -----	67
4.2.2.3 Diagrama de actividades registrar cliente -----	68
4.2.2.4 Diagrama de actividades registrar proveedor -----	68
4.2.2.5 Diagrama de actividades registrar producto -----	70
4.2.2.5 Diagrama de actividades registrar ingreso -----	70
4.2.2.6 Diagrama de actividades registrar venta -----	71
4.2.3 Diagrama entidad relación -----	74

4.2.4 Diccionario de la base de datos	74
4.2.5. Diseños en Figma	83
4.2.5.1 Wireframe Inicio de sesión	83
4.2.5.2 Wireframe Recuperación de usuario	84
4.2.5.3 Wireframe Recuperación de contraseña	84
4.2.5.3 Wireframe Validación de código	85
4.2.5.4 Wireframe Cambio de contraseña	85
4.2.5.2 Wireframe Menú de opciones administrador	86
4.2.5.3 Wireframe Menú de opciones operador	87
4.2.5.4 Wireframe registrar cliente	88
4.2.5.5 Wireframe registrar operador	89
4.2.5.6 Wireframe registrar proveedor	90
4.2.5.7 Wireframe registrar producto	91
4.2.5.8 Wireframe registrar factura de compra	92
4.2.5.9 Wireframe realizar venta	93
4.2.5.10 Wireframe añadir saldo	93
4.2.5.10 Wireframe ver clientes	94
4.2.5.11 Wireframe ver proveedores	94
4.2.5.12 Wireframe ver inventario	95
4.2.5.13 Wireframe historial de ventas	95

4.2.5.13 Wireframe historial de saldos	96
4.2.5.14 Wireframe historial de saldos	96
5.1. Implementar la aplicación web que optimice la gestión y manejo de inventario de la tienda.	97
5.1.1. Desarrollo del backend	97
5.1.1.1 Codificación del proyecto	98
5.1.2. Desarrollo del frontend	103
5.1.2.1 Codificación del frontend	104
5.2. Validar el correcto funcionamiento de la aplicación web con respecto a los requerimientos solicitados por el cliente mediante pruebas unitarias.	106
5.2.1 Instalación y configuración de las librerías para los test	106
5.2.2 Realización de pruebas	107
5.2.2.1 Pruebas al Backend	107
5.2.3 Ejecución de pruebas	108
5.2.3.1 prueba del servicio cliente	108
5.2.3.2 prueba del servicio lote	108
5.2.3.3 prueba del servicio producto	108
5.2.3.4 prueba del servicio proveedor	109
5.2.3.5 prueba del servicio usuario	109
5.2.3.6 prueba del servicio venta	109
5.2.3.7 prueba del servicio saldo	109

5.2.2.2 Pruebas al frontend-----	110
5.2.2.2.1 Pruebas en Login-----	110
5.2.2.2.2 Recuperar Usuario -----	111
5.2.2.2.3 Recuperar Contraseña-----	112
5.2.2.2.4 Registrar proveedores -----	115
5.2.2.2.5 Registrar Cliente-----	116
5.2.2.2.6 Registrar Operador -----	117
5.2.2.2.6 Registrar Producto -----	119
5.2.2.2.7 Registrar Factura Ingreso-----	120
5.2.2.2.8 Registrar Venta -----	122
6. CONCLUSIONES -----	124
7. REFERENCIAS BIBLIOGRÁFICAS -----	125
ANEXOS-----	127
ANEXO A. MANUAL DEL USUARIO -----	127
1. Inicio de sesión -----	127
2. Menú de opciones (Administrador)-----	132
3. Menú de opciones (Operador) -----	133
4. Registrar proveedor-----	133
5. Registrar cliente-----	134
6. Registrar operador-----	135

7.	Registrar producto	136
8.	Registrar Factura Compra	137
9.	Registrar venta	138
10.	Abrir saldo	139
11.	Ver Inventario	140
12.	Ver Clientes	141
13.	Ver proveedores	142
14.	Historial de ventas	143
15.	Historial de saldos	145

Lista de Ilustraciones

Ilustración 1 Aplicación Web tomado de https://www.muycomputer.com/2019/09/26/aplicaciones-web-progresivas/	33
Ilustración 2 Frontend tomado de https://anincubator.com/diferencias-entre-frontend-y-backend-en-el-desarrollo-web/	34
Ilustración 3. HTML tomado de https://borjaarandavaquero.com/que-es/html/	35
Ilustración 4 CSS tomado de https://play.google.com/store/apps/details?id=com.codeliber.csspro&hl=es&gl=US	35
Ilustración 5 JAVASCRIPT tomado de https://soyhorizonte.com/blog/que-es-javascript-y-como-funciona/	36
Ilustración 6 Backend tomado de https://geekflare.com/es/backend-solutions-for-web-and-mobile-apps/	37
Ilustración 7 FRAMEWORKS tomado de https://javifont.medium.com/comparativa-de-frameworks-front-end-con-pruebas-de-rendimiento-d400484fd9c6	38
Ilustración 8 Vue.js tomado de https://hotmart.com/es/marketplace/productos/curso-completo-de-vue-js/T66126685E	38
Ilustración 9 Node.js tomado de https://damiandeluca.com.ar/como-usar-node-en-nuestros-proyectos	39
Ilustración 10 Patrones de Diseño tomado de https://es.ccm.net/contents/224-patrones-de-diseno	40
Ilustración 11 MVC tomado de https://achetronic.medium.com/modelo-vista-controlador-el-incomprendido-controlador-1017cd1ed6d4	40
Ilustración 12 API tomado de https://www.zenvia.com/es/blog/que-es-api/	41

ilustración 13 PostgreSQL tomado de https://www.muylinux.com/2018/07/09/configurar-postgresql-ruby-on-rails/	42
ilustración 14 unit testing tomado de https://jconsoftwares.com/blog/article/14	42
Ilustración 15 Metodología tomado de: https://ayvisa.es/wp-content/uploads/2021/04/generacion-de-ideas.jpg	43
Ilustración 16 Metodología tradicional tomado de: https://sacpma.com/wp-content/uploads/2022/11/image-7.png	44
Ilustración 17 Metodología Agiles tomado de: https://cdnb.ganttpro.com/uploads/2021/11/ques-metodologia-agil.png	45
Ilustración 18 Scrum tomado de https://chet.com.ar/blog/wp-content/uploads/2019/08/metodologia-agile.png	46
Ilustración 19 Diagrama de caso de uso administrador	64
Ilustración 20 Diagrama de caso de uso operador	65
Ilustración 21 Diagrama de actividades Inicio de Sesión	66
Ilustración 22 Diagrama de actividades Registrar operador	67
Ilustración 23 Diagrama de actividades Registrar Cliente	68
Ilustración 24 Diagrama de actividades Registrar proveedor	69
Ilustración 25 Diagrama de actividades Registrar producto	70
Ilustración 26 Diagrama de actividades Registrar Factura de compra	71
Ilustración 27 Diagrama de actividades Registrar venta	72
Ilustración 28 Diagrama entidad relación	74
Ilustración 29 Wireframe Inicio de Sesión	83
Ilustración 30 Wireframe Recuperar Usuario	84
Ilustración 31 Wireframe Recuperar contraseña	84

Ilustración 32 Wireframe Validación de código.....	85
Ilustración 33 Wireframe Cambio de contraseña	85
Ilustración 34 Wireframe Menú de opciones administrador	86
Ilustración 35 Wireframe Menú de opciones operador	87
Ilustración 36 Wireframe Registrar cliente	88
Ilustración 37 Wireframe Registrar operador	89
Ilustración 38 Wireframe Registrar proveedor	90
Ilustración 39 Wireframe Registrar producto	91
Ilustración 40 Wireframe Registrar factura de compra	92
Ilustración 41 Realizar Venta	93
Ilustración 42 Añadir Saldo.....	93
Ilustración 43 Ver clientes.....	94
Ilustración 44 Ver proveedores.....	94
Ilustración 45 Wireframe Ver Inventario	95
Ilustración 46 Wireframe Historial de Ventas.....	95
Ilustración 47 Wireframe Historial saldos.....	96
Ilustración 48 Abonar Saldo	96
Ilustración 49. Estructura del proyecto en NodeJS - Elaboración propia.....	98
Ilustración 50 Server.js	98
Ilustración 51 App.js.....	99
Ilustración 52 Rutas de los endpoints	100
Ilustración 53 cliente.js.....	101
Ilustración 54 Controlador del cliente	102
Ilustración 55 Configuración Vue.js.....	103

Ilustración 56 Estructura del proyecto en Vue.js.....	103
Ilustración 57 Componente Cliente	104
Ilustración 58 View Cliente.....	105
Ilustración 59 index.js.....	105
Ilustración 60 carpeta unit_testing.....	106
Ilustración 61 package.json	106
Ilustración 62 Prueba Unitaria a los endpoint de producto	107
Ilustración 63 Resultado testing servicio cliente	108
Ilustración 64 Resultado testing servicio lote	108
Ilustración 65 Resultado testing servicio producto.....	108
Ilustración 66 Resultado testing servicio proveedor.....	109
Ilustración 67 Resultado testing servicio usuario	109
Ilustración 68 Resultado testing servicio venta	109
Ilustración 69 Resultado testing servicio saldo	110
Ilustración 70 Credenciales Invalidas en Login	110
Ilustración 71 Campos incompletos en Login	111
Ilustración 72 Campo incompleto en Recuperar Usuario.....	111
Ilustración 73 Correo no existe.....	112
Ilustración 74 Campo incompleto en Recuperar contraseña	112
Ilustración 75 Correo no existe.....	113
Ilustración 76 Campo incompleto en validar código.....	113
Ilustración 77 Código incorrecto	114
Ilustración 78 campos incompletos en cambiar contraseña.....	114
Ilustración 79 No coinciden las contraseñas.....	115

Ilustración 80 campos incompletos formulario	115
Ilustración 81 Proveedor registrado.....	116
Ilustración 82 Campos incompletos	116
Ilustración 83 Cliente Registrado	117
Ilustración 84 Correo usuario ya registrado	117
Ilustración 85 Nick ya registrado.....	118
Ilustración 86 Cedula usuario ya registrada	118
Ilustración 87 Formulario producto incompleto.....	119
Ilustración 88 producto ya registrado	119
Ilustración 89 productos sin agregar a Factura	120
Ilustración 90 Proveedor sin asociar.....	120
Ilustración 91 empresa sin agregar	121
Ilustración 92 Id factura sin agregar	121
Ilustración 93 producto sin agregar a venta.....	122
Ilustración 94 Cliente sin agregar a Venta.....	123

Lista de Tablas

Tabla 1 Product Backlog	49
Tabla 2 Roles scrum	50
Tabla 3 Sprint 0	51
Tabla 4 Sprint 1	51
Tabla 5 Sprint 2	52
Tabla 6 Sprint 4	53
Tabla 7 Sprint 5	53
Tabla 8 Requisito funcional 1.....	55
Tabla 9 Requisito funcional 2.....	55
Tabla 10 Requisito funcional 3.....	56
Tabla 10 Requisito funcional 4.....	56
Tabla 11 Requisito funcional 5.....	57
Tabla 12 Requisito funcional 7.....	57
Tabla 13 Requisito funcional 8.....	58
Tabla 14 Requisito funcional 9.....	58
Tabla 15 Requisito funcional 10.....	59
Tabla 16 Requisito funcional 11.....	59
Tabla 17 Requisito funcional 12.....	60
Tabla 18 Requisito funcional 13.....	60
Tabla 19 Requisito funcional 14.....	61
Tabla 20 Requisito funcional 15.....	61
Tabla 21 Requisito no funcional 1.....	62

Tabla 22 Requisito no funcional 2.....	62
Tabla 23 Requisito no funcional 3.....	63
Tabla 24 Requisito no funcional 4.....	63
Tabla 25 Tabla entidad usuario	75
Tabla 26 Tabla entidad cliente.....	76
Tabla 27 Tabla entidad proveedor	76
Tabla 28 Tabla entidad producto	77
Tabla 29 tabla entidad lote.....	78
Tabla 30 Tabla entidad lote_producto	79
Tabla 31 Tabla entidad venta.....	80
Tabla 32 Tabla entidad venta_producto	80
Tabla 33 Tabla entidad saldo.....	81
Tabla 34 Tabla entidad saldo_abono	82

INTRODUCCIÓN

El presente trabajo de grado tiene como finalidad el desarrollo de una aplicación web para el manejo y gestión del inventario de la tienda Yalmecris la cual se enfoca en el aprovisionamiento y abastecimiento de diferentes productos para la comunidad de la localidad, con productos necesarios para la canasta básica familiar, abarrotes, productos de aseo personal entre otros, mediante la aplicación se espera poder tener el control sobre los productos que se presentan en el inventario, un registro de clientes y de proveedores además de registro de ventas y la opción de facilitar abrir un saldo en base a una venta realizada.

El constante avance tecnológico de los últimos años ha permitido contar con una gran cantidad de herramientas tecnológicas que permiten a los negocios crecer de gran manera, facilitando en una alta medida actividades que se hacían de manera manual y presentado soluciones que resuelven muchos problemas en la eficiencia de estas, consiguiendo así ahorrar tiempo y ofreciendo un mayor control o monitorización de las actividades relacionadas con el negocio, una de las herramientas tecnológicas que ha presentado un enorme incremento son las soluciones mediante aplicaciones web, la cual presenta un gran número de alternativas de desarrollo, un amplio abanico de posibilidades, enorme cantidad de información que constantemente es actualizada por la comunidad, dentro de las alternativas de desarrollo de aplicaciones web encontramos tecnologías con Vue.js herramienta con la cual se ha hecho el desarrollo de la frontend de la aplicación, Vue es un framework de desarrollo enfocado a la construcción de interfaces de usuario mediante la creación de componentes facilitando la reutilización de código y con la simplicidad que ofrece la web para la realización del diseño y elaboración de las vista de la capa de presentación, consiguiendo así tener una interfaz amigable con el usuario y con buen acabo final. Por otra parte dentro de la lógica del negocio y la

separación de la capa de presentación encontramos la realización de un backend mediante Node.js siendo este un entorno de ejecución del lado del servidor el cual permite crear y aislar la lógica del negocio de la capa de presentación y además de que es el medio para la construcción de la API la cual se encargara de hacer la comunicación con la base de datos relacional que se desarrolló con PostgreSQL y así exponer los servicios y dar solución a las funcionalidades necesarias para el correcto funcionamiento de la aplicación cumpliendo con la lógica del modelo de negocio, finalmente para la validación final se hace importante la realización de pruebas , las cuales fueron aplicadas a los endpoints de los servicios realizados en el backend.

CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

En los últimos años el crecimiento y desarrollo tecnológico se encuentra en constante incremento, cada día son más las empresas, negocios y emprendimientos que utilizan las herramientas que ofrece la tecnología para dar soporte a sus negocios, esto se debe a que los sistemas de información permiten contemplar los diferentes modelos de negocios y ofrecer una solución alternativa más eficiente y en algunos casos una más competitiva.

Pese al incremento tecnológico aún existe un gran número de negocios que no optan por soluciones más modernas, dentro del grupo de negocios que aún no han implementado herramientas tecnológicas para dar soporte a su modelo de negocios se encuentra una tienda comercial denominada “YALMECRIS”, ubicada en la localidad de “las Américas”. Dicha tienda está enfocada en el abastecimiento de víveres, abarrotes, productos de aseo personal, entre otros. Actualmente, la forma como se está llevando el manejo y gestión del inventario es manual, ya que se registra la información de los productos existentes, proveedores, ventas y saldos en un cuaderno, siendo esto una manera poco efectiva a la hora de conocer la totalidad de los productos que poseen y productos vendidos, dificultando también el proceso de realizar el pedido de los productos faltantes, al mismo tiempo, para conocer los precios de los productos se necesita aprenderlos por memorización o buscar las facturas, sumado a lo anterior, en cuanto al control de precios con los proveedores se puede apreciar que es muy ambiguo y no hay forma de verificar y constatar de una manera breve y acertada los nuevos y antiguos precios teniendo en cuenta, además, como se mueve el mercado y sus posibles adaptaciones a nuestra necesidad, siendo esto un gran problema para la propietaria del negocio que no ocurriría si contara con una solución eficiente a través de un sistema de información que se encargue de la supervisión del stock y gestión del inventario.

1.2. FORMULACIÓN DEL PROBLEMA

¿Mediante que herramienta tecnológica se puede volver más eficiente el proceso de gestión y manejo del inventario de la tienda “Yalmecris”?

1.3. JUSTIFICACIÓN

El progreso y desarrollo que ha tenido la tecnología en la última década ha permitido que muchas de las tareas que anteriormente eran realizadas por personas de manera manual, se conviertan en tareas más automatizadas, eficientes y efectivas, el avance tecnológico supone un aumento en la productividad en gran medida este efecto ha generado que un alto número de empresas opten por soluciones tecnológicas más efectivas acorde a su modelo de negocio, ya que mediante sistemas de información se puede manejar de manera correcta el flujo de datos y almacenarlos para un fácil acceso y su seguridad. Es por este motivo que nace la necesidad de implementar una aplicación web en el negocio “Yalmecris” ya que esta solución les permitirá conocer mejor la totalidad de su inventario, ver los productos faltantes para su posterior pedido aumentando el flujo y control de trabajo que el negocio actualmente posee.

El objetivo es desarrollar una aplicación web ya que las tecnologías de desarrollo actuales ofrecen soluciones muy efectivas, además que con el crecimiento de la web ha resultado en un amplio abanico de tecnologías de desarrollo como lo llega a ser Vue.js que es un framework de desarrollo enfocado a la utilización y separación del código mediante componentes, permitiendo un flujo de trabajo más ágil y mediante un patrón de diseño para hacer más ordenado el código que se desarrolla, además que permite el desarrollo mediante herramientas como lo son HTML, CSS y JavaScript siendo estas las tecnologías predominantes en el diseño y desarrollo web.

1.4. OBJETIVOS

1.4.1. Objetivo General

- Desarrollar una aplicación web para el manejo y gestión del inventario a través del framework Vue.js para la tienda “YALMECRIS”

1.4.2. Objetivos Específicos

- Diagnosticar el proceso actual que se tiene en la tienda, para el control de inventario.
- Establecer los requisitos funcionales y no funcionales en base a los requerimientos que necesita el cliente para el desarrollo de la aplicación web.
- Diseñar la aplicación web para la gestión de inventario de la tienda.
- Implementar la aplicación web que optimice la gestión y manejo de inventario de la tienda.
- Validar el correcto funcionamiento de la aplicación web con respecto a los requerimientos solicitados por el cliente mediante pruebas unitarias.

1.4.3. Acotaciones

- Se utilizará la plataforma Figma para realizar los diseños del sistema de información.
- El frontend de la aplicación web de escritorio se desarrollará en Vue.js.
- El backend de la aplicación web de escritorio se desarrollará con NodeJS y Express.
- Creación de la base de datos con PostgreSQL para el almacenamiento de la información.

CAPÍTULO 2. MARCO TEÓRICO

2.1. ANTECEDENTES

Se presenta los trabajos de investigación, los artículos que son importantes referentes para la propuesta.

2.1.1. Antecedentes Internacional

En la tesis de (Sánchez Villalva, 2018) titulada “Diseño e implementación de una aplicación web para el control de inventario de la empresa “DGC Jeans” el autor realizó el desarrollo, diseño e implementación de una aplicación web para el control de inventario de la empresa “Dgc Jeans”, utilizando Spring y Oracle, se realizó un estudio de los procesos administrativos para conocer cómo se llevan a cabo las actividades de cada una de las áreas involucradas en la confección de los productos, además se efectuó el análisis del framework Spring y la base de datos Oracle para el desarrollo del sistema, se realizó la evaluación de la funcionalidad del sistema CISBlue aplicando el estándar de calidad ISO/IEC 9126, para el desarrollo del sistema se utilizó la metodología ágil SCRUM, lo que permitió realizar pequeños entregables del producto al final de cada sprint, los componentes se integraron con la librería de primeface y bootstrap, para garantizar el almacenamiento de los datos se utilizó Oracle 11g XE como gestor de base de datos. Finalmente se realizó la evaluación de las métricas de calidad de la funcionalidad del software propuestas por estándar ISO/IEC 9126 y se determinó que el sistema CISBlue es 83.6% funcional, debido al análisis de calidad funcional de la aplicación obtenida se recomienda ser implantada en otras instituciones que se dediquen a la misma actividad económica de la empresa Dgc Jeans. Este trabajo de grado se realizó bajo la metodología ágil SCRUM, la cual es reconocida como una de las metodologías ágiles más utilizadas actualmente, en base a este hecho, la investigación aporta información útil que puede ser utilizada dentro de la comparación del conjunto de diferentes metodologías de desarrollo.

En la tesis de (Lema Sagbaycela, 2018) titulada “Desarrollo del sistema web para el control de inventarios, ventas, facturación y publicidad del Taller de Aluminio y Vidrio “López” aplicando la metodología Lean Software Development” el autor desarrolla un sistema web para el control de inventarios, ventas, facturación y publicidad del taller de aluminio y vidrio “López” en el que se utiliza la metodología ágil LEAN SOFTWARE DEVELOPMENT esta metodología proporciona procesos en los cuales se establecen normas y reglas que ayudan a que el proyecto emprendido tenga mayor probabilidad de éxito, se basa en principios en los cuales establecen directrices óptimas para el desarrollo del software. Uno de los aspectos importantes para el desarrollo de software son las tecnologías de desarrollo a utilizar, en este proyecto se ha

utilizó JAVA, EDGRID, JavaScript, HTML, CSS, MYSQL gestor de base de datos, IDE NETBEANS 8.2 entorno de desarrollo integrado. Al aplicar la metodología ágil, en el desarrollo se obtuvo 41 requerimientos funcionales y no funcionales, se construyeron 45 diagramas que ayudaron a visualizar los procesos que el usuario realiza en el sistema web.

En el trabajo de grado se presenta el desarrollo de un sistema web para el control inventarios el cual se desarrolla bajo herramientas similares a las mencionadas en esta investigación permitiendo ver el funcionamiento efectivo de dichas herramientas para dar solución a una problemática muy similar a la presentada, sirviendo, así como un marco de referencia para el desarrollo actual.

2.1.2. Antecedentes Nacional

En la tesis de (Tamayo Flórez, 2019) titulada “Desarrollo de sistema de gestión de activos del área de ingeniería clínica de la fundación colombiana de cancerología Clínica Vida” es este trabajo el autor diseñó una aplicación que cumpliera con requerimientos previamente establecidos con el personal de Ingeniería Clínica, esta se dividió en dos partes: (1) una aplicación web y (2) una aplicación de escritorio. La primera ayuda al manejo de inventario y los datos relacionados a un equipo mientras que la segunda es una versión de la aplicación para manejo de Excel y diligenciamiento de formatos de hoja de vida propios de la Clínica Vida. Ambas ayudan a automatizar y controlar los procesos asociados a los activos de dispositivos biomédicos de la institución cumpliendo con los objetivos y los requisitos establecidos durante este trabajo. Al igual que en la presente investigación este trabajo de grado presenta una solución a la manera en cómo se realizaban los procesos manuales de la clínica de cancerología Vida por medio de una aplicación web consiguiendo una mejora en la eficiencia y control de dichos procesos.

En la tesis de (Córdoba Lemus, 2021) titulada “Diseño e implementación de un sistema de gestión integral aplicado al mantenimiento, seguimiento y control de los equipos de rayos x”, en este trabajo el autor planteó diseñar e implementar un sistema de gestión por medio de una aplicación web, la cual permite la captura de información y gestión de los procesos relacionados a los equipos. La aplicación se desarrolló bajo el framework de desarrollo Angular,

NodeJS como entorno de desarrollo de JavaScript, y conexión a la base de datos implementada en MongoDB. Para esto se establecieron 5 etapas de desarrollo en la metodología las cuales van desde una revisión, análisis y diagnóstico de los archivos e inventarios existentes en la empresa, hasta la capacitación al personal de la aplicación implementada. Este proyecto trae como resultado la mejora de los procesos de mantenimiento, seguimiento y control de los equipos biomédicos en la empresa, logrando de esta manera eficiencia en los servicios prestados y calidad operativa. Por otro lado, mediante la implementación de la Aplicación Web se llevarán todos los registros e información asociada a los dispositivos de una manera sencilla, en tiempo real, flexible y de fácil acceso, ya que se puede realizar desde cualquier computador con conexión a internet. Además, el aplicativo permitir tener una experiencia de usuario simple y genera notificaciones para mejorar la toma de decisiones. La contribución que realiza este trabajo de grado a la investigación actual se basa en la mejora de los procesos del negocio a través de la implementación de una aplicación web en la cual se aumenta la eficiencia de los servicios prestados.

2.1.3. Antecedentes Regional

En la tesis de (Reyes Getial, 2017) titulada “Análisis y diseño de un sistema de información, basado en web móvil para la asignación de citas médicas de la IPS centro clínico integral Tibú Ltda.” El autor diseña un sistema de información para la asignación de citas de la IPS Centro Clínico Integral Tibú Ltda. Haciendo uso de la ingeniería de requerimientos y la metodología MOBILE-D, este estudio se centra en el área de la salud el fin de este proyecto es contar con una herramienta informática eficiente que permita la obtención de información verídica y actualizada para un funcionamiento acorde a las necesidades tanto del personal obrero como de los pacientes que interactúan en el centro médico. La importancia de este trabajo de grado radica en que permite observar el análisis sobre las diferentes metodologías con las cuales se puede desarrollar un sistema de información siendo esto uno de los objetivos de la presente investigación.

En la tesis de (Betancur Rosas, 2018) titulada “Aplicación web para la gestión de

inventario de reactivos del laboratorio de química de la universidad de Pamplona sede Villa del Rosario” el autor desarrolla e implementa una aplicación web para la gestión de inventario de reactivos del laboratorio de química de la Universidad de Pamplona sede Villa del Rosario. Para realizar este desarrollo el autor determinó el diseño metodológico, estableciendo la investigación descriptiva, que permitió conocer la realidad de la población objeto de estudio conformada por los auxiliares, docentes y coordinador del laboratorio, a los cuales se aplicó el instrumento de recolección, es decir, la entrevista y la observación directa, en donde la metodología de analizarla cuantitativa y cualitativamente mediante tablas y gráficas que permitieron conocer el nivel de aceptación de los Docentes y Administrativos, en poder tener una herramienta que facilite la gestión de solicitud de reactivos que se encuentran disponible en la página web de la Universidad de Pamplona y así poder diligenciar el formato en forma virtual fla-04 v 02 solicitud de reactivos y medios. El aporte que este trabajo de grado presenta es referente a la problemática del manejo de inventario por medio de métodos físicos y como a través de una aplicación Web se puede llevar un correcto orden en la gestión del inventario.

2.2. MARCO TEÓRICO

2.2.1. APLICACIÓN WEB

En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. (Wictea, 2022)



Ilustración 1 Aplicación Web tomado de <https://www.muycomputer.com/2019/09/26/aplicaciones-web-progresivas/>

2.2.2. FRONTEND

Front End es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando accedemos a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas (RWD), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales... y otros elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario. Como hemos dicho, el desarrollador frontend se encarga de la experiencia del usuario, es decir, en el momento en el que este entra a una página web, debe ser capaz de navegar por ella, por lo

que el usuario verá una interface sencilla de usar, atractiva y funcional. (descubre comunicación, 2019)



Ilustración 2 Frontend tomado de <https://anincubator.com/diferencias-entre-frontend-y-backend-en-el-desarrollo-web/>

2.2.3. HTML

HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript).

"Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a las páginas creadas por otras personas, te conviertes en un participante activo en la «World Wide Web» (mdn web docs, 2022)



Ilustración 3. HTML tomado de <https://borjaarandavaquero.com/que-es/html/>

2.2.4. CSS

Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (en-US) (incluyendo varios languages basados en XML como SVG, MathML o XHTML). CSS describe como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios. (mdn web docs, 2021)



Ilustración 4 CSS tomado de <https://play.google.com/store/apps/details?id=com.codeliber.csspro&hl=es&gl=US>

2.2.5. JAVASCRIPT

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (mdn web docs, 2022).

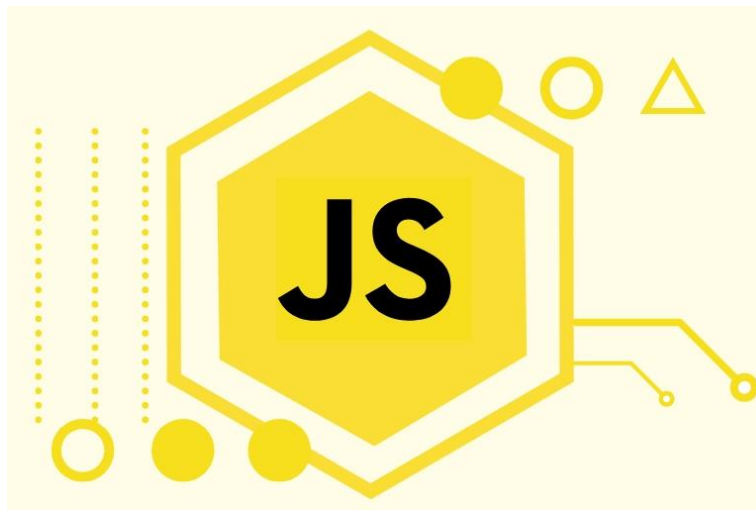


Ilustración 5 JAVASCRIPT tomado de <https://soyhorizonte.com/blog/que-es-javascript-y-como-funciona/>

2.2.6. BACKEND

Como hemos dicho antes, Frontend es todo con lo que el usuario se encuentra directamente en la web o aplicación, entonces cuando hablamos de “Backend” nos referimos al interior de las aplicaciones que viven en el servidor y al que a menudo se le denomina “el lado del servidor”.

El backend del sitio web consiste en un servidor, una aplicación y una base de datos. Se toman los datos, se procesa la información y se envía al usuario. (descubre comunicación, 2019).

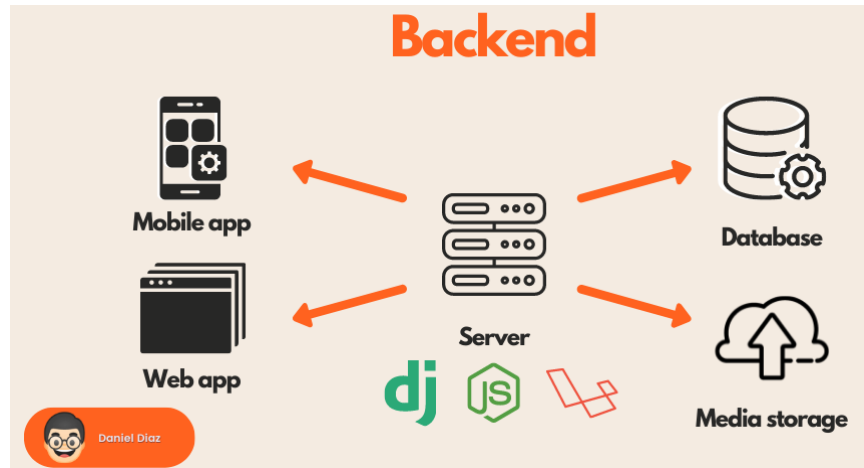


Ilustración 6 Backend tomado de <https://geekflare.com/es/backend-solutions-for-web-and-mobile-apps/>

2.2.7. FRAMEWORK

Un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software.

Utilizar frameworks puede simplificar una tarea o proceso, de ahí que se trate de una de las herramientas habituales que manejan los Digital Workers, porque les ayuda a ser más ágiles y productivos. Generalmente, los frameworks son usados por programadores porque permiten acelerar el trabajo y favorecer que este sea colaborativo, reducir errores y obtener un resultado de más calidad. (edix, 2021)



Ilustración 7 FRAMEWORKS tomado de <https://javifont.medium.com/comparativa-de-frameworks-front-end-con-pruebas-de-rendimiento-d400484fd9c6>

2.2.8. VUE.JS

Es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas Single-Page Applications cuando se utiliza en combinación con herramientas modernas y librerías de apoyo. (Vue.js, s.f.)



Ilustración 8 Vue.js tomado de <https://hotmart.com/es/marketplace/productos/curso-completo-de-vue-js/T66126685E>

2.2.9. NODE.JS

Node.js es un entorno de ejecución de un solo hilo, de código abierto y multiplataforma para crear aplicaciones de red y del lado del servidor rápidas y escalables. Se ejecuta en el motor de ejecución de JavaScript V8, y utiliza una arquitectura de E/S basada en eventos y sin bloqueos, lo que la hace eficiente y adecuada para aplicaciones en tiempo real. (Kinsta, 2021)



Ilustración 9 Node.js tomado de <https://damiandeluca.com.ar/como-usar-node-en-nuestros-proyectos>

2.2.10. PATRONES DE DISEÑO

Los patrones de diseño o design patterns, son una solución general, reutilizable y aplicable a diferentes problemas de diseño de software. Se trata de plantillas que identifican problemas en el sistema y proporcionan soluciones apropiadas a problemas generales a los que se han enfrentado los desarrolladores durante un largo periodo de tiempo, a través de prueba y error.

Los patrones de diseño ayudan a estar seguro de la validez del código, ya que son soluciones que funcionan y han sido probados por muchísimos desarrolladores siendo menos propensos a errores. (profile, 2020)



Ilustración 10 Patrones de Diseño tomado de <https://es.ccm.net/contents/224-patrones-de-diseno>

2.2.11. MODELO-VISTA-CONTROLADOR

MVC (Modelo-Vista-Controlador) es un patrón en el diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Enfatiza una separación entre la lógica de negocios y su visualización. Esta "separación de preocupaciones" proporciona una mejor división del trabajo y una mejora de mantenimiento.

Las tres partes del patrón de diseño de software MVC se pueden describir de la siguiente manera

- a) Modelo: Maneja datos y lógica de negocios.
- B) Vista: Se encarga del diseño y presentación.
- C) Controlador: Enruta comandos a los modelos y vistas. (mdn web docs , 2022)

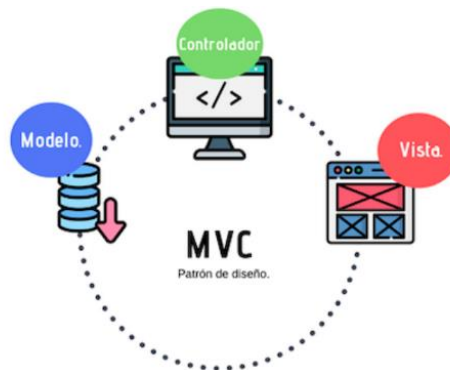


Ilustración 11 MVC tomado de <https://achetronic.medium.com/modelo-vista-controlador-el-incomprendido-controlador-1017cd1ed6d4>

2.2.12. API (Application Programming Interface)

El término API es una abreviatura de Application Programming Interfaces, que en español significa interfaz de programación de aplicaciones. Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.

Así pues, podemos hablar de una API como una especificación formal que establece cómo un módulo de un software se comunica o interactúa con otro para cumplir una o muchas funciones. Todo dependiendo de las aplicaciones que las vayan a utilizar, y de los permisos que les dé el propietario de la API a los desarrolladores de terceros. (Xataka, 2019)



Ilustración 12 API tomado de <https://www.zenvia.com/es/blog/que-es-api/>

2.2.13. POSTGRESQL

PostgreSQL es un gestor de bases de datos relacional y orientado a objetos. Su licencia y desarrollo es de código abierto, siendo mantenida por una comunidad de desarrolladores, colaboradores y organizaciones comerciales de forma libre y desinteresadamente. esta comunidad es denominada PDGD (PostgreSQL global Development Group, por sus siglas en inglés).

es reconocido actualmente como uno de los sistemas gestores de bases de datos relacionales más potentes del mercado. presenta fácil accesibilidad, es multiplataforma y está disponible para su utilización en casi todos los sistemas operativos utilizados en la actualidad sin disminuir su rendimiento. (HostingPedia, 2019)



ilustración 13 PostgreSQL tomado de <https://www.muylinux.com/2018/07/09/configurar-postgresql-ruby-on-rails/>

2.2.14. PRUEBAS UNITARIAS

Las pruebas unitarias consisten en aislar una parte del código y comprobar que funciona a la perfección. son pequeños tests que validan el comportamiento de un objeto y la lógica. el unit testing suele realizarse durante la fase de desarrollo de aplicaciones de software o móviles. normalmente las llevan a cabo los desarrolladores, aunque en la práctica, también pueden realizarlas los responsables de QA. (YeePLY, 2021)



ilustración 14 unit testing tomado de <https://jconsoftwares.com/blog/article/14>

2.2.15. METODOLOGÍAS DE DESARROLLO DE SOFTWARE

2.2.15.1 METODOLOGÍA

La metodología hace referencia al conjunto de procedimientos racionales utilizados para alcanzar un objetivo que requiera habilidades y conocimientos específicos. La metodología es una de las etapas específicas de un trabajo o proyecto que parte de una posición teórica y conlleva a una selección de técnicas concretas o métodos acerca del procedimiento para el cumplimiento de los objetivos. Es el conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla. Determina los pasos a seguir y cómo realizarlos para finalizar una tarea. (Esteban Gabriel, 2015)



Ilustración 15 Metodología tomado de: <https://ayvisa.es/wp-content/uploads/2021/04/generacion-de-ideas.jpg>

2.2.15.2 METODOLOGÍA TRADICIONAL

Desarrollar un buen software depende de un gran número de actividades y etapas, donde el impacto de elegir la metodología para un equipo en un determinado proyecto es trascendental para el éxito del producto. Las metodologías tradicionales son denominadas, a veces, de forma despectiva, como metodologías pesadas. Centran su atención en llevar una documentación exhaustiva de todo el proyecto, la planificación y control del mismo, en especificaciones precisas de requisitos y modelado y en cumplir con un plan de trabajo, definido todo esto, en la fase inicial del desarrollo del proyecto. Estas metodologías tradicionales imponen una disciplina rigurosa de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un

software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. (Esteban Gabriel, 2015)



Ilustración 16 Metodología tradicional tomado de: <https://sacpma.com/wp-content/uploads/2022/11/image-7.png>

2.2.15.3 METODOLOGÍA AGILES

Este enfoque nace como respuesta a los problemas que puedan ocasionar las metodologías tradicionales y se basa en dos aspectos fundamentales, retrasar las decisiones y la planificación adaptativa. Basan su fundamento en la adaptabilidad de los procesos de desarrollo. Un modelo de desarrollo ágil, generalmente es un proceso Incremental (entregas frecuentes con ciclos rápidos), también Cooperativo (clientes y desarrolladores trabajan constantemente con una comunicación muy fina y constante), Sencillo (el método es fácil de aprender y modificar para el equipo) y finalmente Adaptativo (capaz de permitir cambios de último momento). Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas pragmáticas que hacen que la entrega del proyecto sea menos complicada y más satisfactoria tanto para los clientes como para los equipos de trabajo, evitando de esta manera los caminos burocráticos de las metodologías tradicionales, generando poca documentación y no haciendo uso de métodos formales. Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. (Esteban Gabriel, 2015)

GANTTPRO

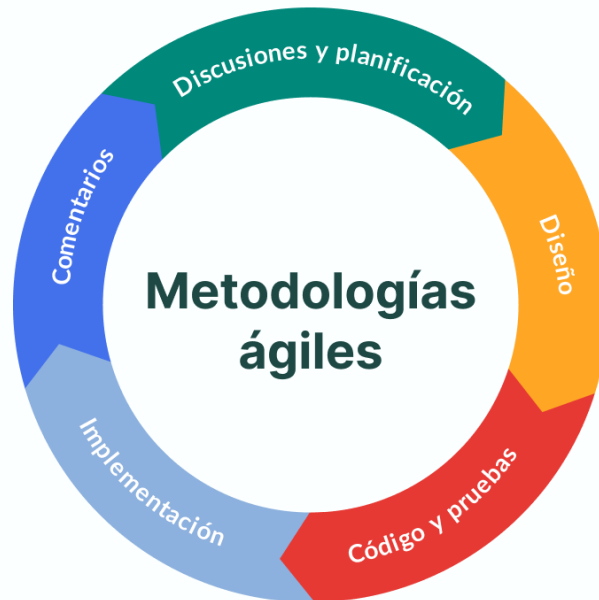


Ilustración 17 Metodología Ágiles tomado de: <https://cdnb.ganttpro.com/uploads/2021/11/que-es-metodologia-agil.png>

2.2.15.4 METODOLOGÍA ÁGIL SCRUM

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto. (proyectos agiles.org, 2018)

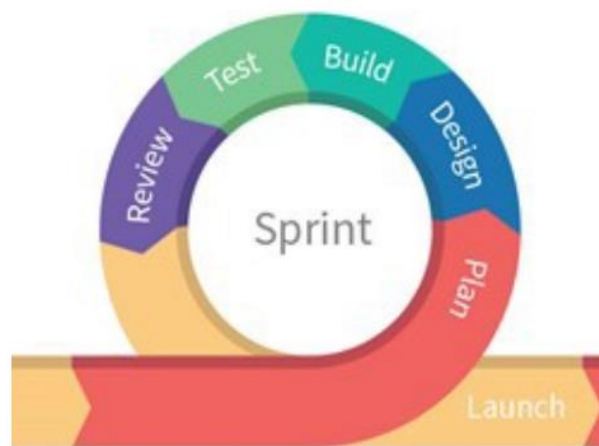


Ilustración 18 Scrum tomado de <https://chet.com.ar/blog/wp-content/uploads/2019/08/metodologia-agile.png>

2.3. MARCO LEGAL

A continuación, se presenta un listado de leyes que el gobierno de colombiano dispone para la elaboración de proyectos.

2.3.1. Ley Estatutaria 1581 de 2012

Por la cual se dictan disposiciones generales para la protección de datos personales. Según

ARTÍCULO 1°. OBJETO. La presente ley tiene por objeto desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos, y los demás derechos, libertades y garantías constitucionales a que se refiere el artículo 15 de la Constitución Política; así como el derecho a la información consagrado en el artículo 20 de la misma.

2.3.2. Ley 1341 TIC en Colombia

Por la cual se definen principios y conceptos sobre la sociedad de la información y la organización de las Tecnologías de la Información y las Comunicaciones –TIC–, se crea la Agencia Nacional de Espectro y se dictan otras disposiciones. Por lo tanto, **ARTÍCULO 1°. OBJETO.** La presente ley determina el marco general para la formulación de las políticas públicas que regirán el sector de las Tecnologías de la Información y las Comunicaciones, su ordenamiento general, el régimen de competencia, la protección al usuario, así como lo concerniente a la cobertura, la calidad del servicio, la promoción de la inversión en el sector y el desarrollo de estas tecnologías, el uso eficiente de las redes y del espectro radioeléctrico, así como las potestades del Estado en relación con la planeación, la gestión, la administración adecuada y eficiente de los recursos, regulación, control y vigilancia del mismo y facilitando el libre acceso y sin discriminación de los habitantes del territorio nacional a la Sociedad de la Información.

2.3.3. Ley 1314 de 2009

Por la cual se regulan los principios y normas de contabilidad e información financiera y de aseguramiento de información aceptados en Colombia, se señalan las autoridades competentes, el procedimiento para su expedición y se determinan las entidades responsables de vigilar su cumplimiento.

2.3.4. Artículo 63 del Decreto de 2649 de 1993

ARTICULO 63. INVENTARIOS. Los inventarios representan bienes corporales destinados a la venta en el curso normal de los negocios, así como aquellos que se hallen en proceso de producción o que se utilizarán o consumirán en la producción de otros que van a ser vendidos.

CAPITULO 3. METODOLOGÍA PARA DESARROLLO AGIL

A continuación, presentaremos la metodología de desarrollo, incluyendo las respectivas fases e iteraciones, y el backlog correspondiente, para luego continuar con el desarrollo de la aplicación y realizar la descripción. La metodología elegida para el desarrollo es SCRUM el cual es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos (proyectos agiles.org, 2018).

3.1.1. Product Backlog

El backlog de un producto es una lista de trabajo ordenado por prioridades para el equipo de desarrollo que se obtiene de la hoja de ruta y sus requisitos. Los elementos más importantes se muestran al principio del backlog del producto para que el equipo sepa qué hay que entregar primero. (Atlassian, s.f.). A continuación, se puede observar el Product backlog con las actividades a desarrollar una descripción y la prioridad.

Tabla 1 Product Backlog

Ítem		Descripción	Prioridad
1	Diseño de la arquitectura del software	Se realizar el diseño de la arquitectura del sistema de información haciendo uso de los diferentes diagramas que son necesarios para el desarrollo del software.	1
2	Realización del frontend de la aplicación	Desarrollo de los diseños de la capa de presentación que tendrá la aplicación web a través de Vue.js.	2
3	Codificación del backend de la aplicación	Desarrollo de la base de datos relacional en PostgreSQL y de la API en Node.js encargada de exponer los servicios para su posterior consumo mediante el frontend.	2
4	Consumo de los servicios de la API desde el frontend	Mediante la librería Axios consumir los servicios que han sido expuestos en la API y así consultar y enviar datos para así darle interactividad al sistema.	2

5	Realización de pruebas	Por medio de pruebas unitarias se verificará que la aplicación cumple con los requisitos establecidos.	2
6	Entrega de Manual de usuario	Desarrollo de un manual de usuario	1

3.1.2. Roles

En Scrum, el equipo se focaliza en construir software de calidad. La gestión de un proyecto Scrum se centra en definir cuáles son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en vencer cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. (Softeng, s.f.)

El equipo Scrum está formado por el Scrum Master, el Product owner y el equipo de desarrollo.

Tabla 2 Roles scrum

Rol	Persona
Product Owner	Meily Fabiana Pedroza Arévalo
Scrum Master	Dimar David Galvan Bayona
Developer	Dimar David Galvan Bayona

3.1.3. Sprint Backlog

Un sprint es un período breve de tiempo fijo en el que un equipo de scrum trabaja para completar una cantidad de trabajo establecida. Los sprints se encuentran en el corazón de las metodologías scrum y ágil, y hacer bien los sprints ayudará a tu equipo ágil a lanzar mejor software (Atlassian, s.f.)

3.1.3.1 Sprint 0

Tabla 3 Sprint 0

Sprint 0						
Ítem Backlog	Tareas		Fecha Fin	Fecha Fin	Total, Días	Completada al finalizar el Sprint
1. Diseño de la arquitectura del software	1	Análisis y diagnóstico inicial de las necesidades del proyecto.	01/10/2022	03/10/2022	3	Si
	2	Establecimiento de Requisitos	04/10/2022	07/10/2022	4	Si
	3	Realización de diagramas de caso de Uso y Actividades	08/10/2022	12/10/2022	5	Si

3.1.3.2 Sprint 1

Tabla 4 Sprint 1

Sprint 1						
Ítem Backlog	Tareas		Fecha Inicio	Fecha Fin	Total, Días	Completada al finalizar el Sprint
2.Realización del frontend de la aplicación	1	Diseño de las interfaces en Figma	09/10/2022	11/10/2022	3	Si
	2	Creación y configuración del proyecto en Vue.js	11/10/2022	13/10/2022	3	Si
	3	Codificación de los componentes de cada vista y configuración de las rutas	14/10/2022	19/10/2022	6	Si
	4	Creación y configuración de las rutas	20/10/2022	22/10/2022	3	Si

3.1.3.3 Sprint 2

Tabla 5 Sprint 2

Sprint 2						
Ítem Backlog	Tareas		Fecha Inicio	Fecha Fin	Total, Días	Completada al finalizar el Sprint
3.Codificación del backend de la aplicación	1	Creación de los scripts para la base de datos.	23/10/2022	25/10/2022	3	Si
	2	Creación, configuración e Instalación de módulos para el proyecto en Node.js	26/10/2022	27/10/2022	2	Si
	3	Creación de los controladores que contienen la lógica de negocio.	28/10/2022	3/11/2022	7	Si
	4	Creación y configuración de las rutas que sirven como endpoints.	4/11/2022	8/11/2022	5	Si

3.1.3.4 Sprint 3

Sprint 3						
Ítem Backlog	Tareas		Fecha Inicio	Fecha Fin	Total, Días	Completada al finalizar el Sprint
4. Consumo de los servicios de la API desde el frontend	1	Instalación de la librería Axios en el proyecto de Vue.js	9/11/2022	10/11/2022	2	Si
	2	Codificación de los métodos para el consumo de cada servicio	11/11/2022	15/11/2022	5	Si
	3	Configuración de eventos para cada uno de los formularios.	16/11/2022	18/11/2022	3	Si

3.1.3.5 Sprint 4

Tabla 6 Sprint 4

Sprint 4						
Ítem Backlog	Tareas		Fecha Inicio	Fecha Fin	Total, Días	Completada al finalizar el Sprint
5. Realización de pruebas	1	Instalación y configuración de las librerías necesarias para realizar las pruebas unitarias.	19/11/2022	21/11/2022	1	Si
	2	Creación de los códigos necesarios para las pruebas a los endpoints del backend.	22/11/2022	24/11/2022	1	Si
	3	Realización y verificación de las pruebas unitarias.	25/11/2022	28/11/2022	1	Si

3.1.3.6 Sprint 5

Tabla 7 Sprint 5

Sprint 5						
Ítem Backlog	Tareas		Fecha Inicio	Fecha Fin	Total, Semanas	Completada al finalizar el Sprint
5. Entrega de Manual de usuario	1	Desarrollo del manual de usuario que le permitirá al mismo conocer la interactividad del sistema.	29/11/2022	1/12/2022	1	Si

CAPITULO 4. DESARROLLO DEL SISTEMA

En este capítulo desarrollaremos una aplicación web para el manejo y gestión del inventario a través del framework Vue.js para la tienda “Yalmecris”

4.1. Diagnosticar el proceso actual que se tiene en la tienda, para el control de inventario.

Como se ha hecho saber a lo largo del presente documento el enfoque de este trabajo se basa en presentar una herramienta tecnológica como lo es una aplicación web para el modelo de negocio de la tienda Yalmecris, es por este motivo que se hace imprescindible conocer cómo se llevan los procesos antes de presentada esta alternativa de solución, resaltando como primer detalle que en el negocio todos los procesos se realizan de manera manual sin hacer uso de alguna herramienta tecnológica a excepción de unas tablas en Excel, partiendo de la base que en el negocio no existe propiamente un control del inventario ya que no se lleva un conteo de los productos que son recibidos dentro del establecimiento desconociendo así la totalidad de estos, como otro punto importante la realización de los pedidos se hace una vez o dos veces por semana dependiendo de la empresa y el flujo de venta de los productos, este proceso de pedido se realiza mediante el acercamiento que hacen los proveedores y estos se encargan de ofrecer los productos y el dueño del negocio estima cuantos productos necesita, pero esta estimación a veces puede resultar erróneo ya que en ocasiones se desconocía cuantos productos quedaban realmente y no se ha hecho el pedido adecuado dando como resultado la escases de los productos o la falta de estos durante 8 días, con lo que respecta a la ventas estas se anotan en un cuaderno pero en este cuaderno actualmente no se lleva control de los clientes ya que la tienda no toma los datos de los estos, tampoco se apunta que productos se vendió solo se coloca un valor de la venta y al final del día estos valores son llevados a una tabla de Excel para tener un estimado de las ventas del día, al final de cada mes se hace un estimado del margen de las ganancias con los datos que se tienen guardados, en algunas ocasiones se abren saldos para algunos de los clientes y este saldo es registrado en el mismo cuaderno donde se llevan las cuentas de las ventas realizadas, para el pago de estos saldos no se lleva registro ya que los clientes se acercan al negocio y hacen el pago del saldo y este se tacha del cuaderno

4.2. Establecer los requisitos funcionales y no funcionales en base a los requerimientos que necesita el cliente para el desarrollo de la aplicación web.

A continuación, se mostrarán las tablas de los requisitos de funcionales y no funcionales lo cual resulta imprescindible para dar conocimiento de las funcionales que va a necesitar la aplicación web todo esto se ha definido en base a diagnostico que se ha realizado.

4.2.1. Requisitos Funcionales

4.2.1.1 Requisito funcional 1

Tabla 8 Requisito funcional 1

Identificación del requisito:	RF01
Nombre del requisito:	Inicio de Sesión
Características:	Para acceder se necesita Iniciar Sesión
Descripción del requisito:	El sistema deberá permitir el acceso mediante el ingreso de los campos usuario y contraseña.
Prioridad del requisito: Alta	

4.2.1.2 Requisito funcional 2

Tabla 9 Requisito funcional 2

Identificación del requisito:	RF02
Nombre del requisito:	Registro de Cliente
Características:	Registrar Datos de un cliente
Descripción del requisito:	El sistema deberá permitir registrar los datos de un cliente como su Nombre, Tipo de Documento, Numero de Documento, Teléfono y la dirección.
Prioridad del requisito: Alta	

4.2.1.3 Requisito funcional 3

Tabla 10 Requisito funcional 3

Identificación del requisito:	RF03
Nombre del requisito:	Registro de Proveedor
Características:	Registrar Datos de un cliente
Descripción del requisito:	El sistema deberá permitir registrar los siguientes campos de un proveedor su Nombre, Tipo de Documento, Numero de Documento, Teléfono y correo.
Prioridad del requisito: Alta	

4.2.1.4 Requisito funcional 4

Tabla 11 Requisito funcional 4

Identificación del requisito:	RF04
Nombre del requisito:	Registro de Operador
Características:	Registrar Datos de un operador
Descripción del requisito:	El sistema deberá permitir registrar los siguientes campos de un operador como su Nombre, Numero de Documento, Nick, correo y una contraseña.
Prioridad del requisito: Alta	

4.2.1.5 Requisito funcional 5

Tabla 12 Requisito funcional 5

Identificación del requisito:	RF05
Nombre del requisito:	Registro Producto
Características:	Registrar Datos de un producto
Descripción del requisito:	El sistema deberá permitir registrar un producto y los detalles de este como el nombre, su código, la marca, la categoría a la que pertenece, la unidad de medida de este y la cantidad en base a la medida que posee precio de entrada y precio de venta.
Prioridad del requisito: Alta	

4.2.1.6 Requisito funcional 7

Tabla 13 Requisito funcional 7

Identificación del requisito:	RF07
Nombre del requisito:	Registro Factura de compra
Características:	Registrar Datos de un Ingreso de factura de productos
Descripción del requisito:	El sistema deberá permitir registrar el ingreso de productos especificando el código del producto del cual va hacer el ingreso, el precio de entrada del producto, el precio de venta para el público y la cantidad de productos, el código de la factura, la fecha de la factura, el total de la factura y la cedula del proveedor.
Prioridad del requisito: Alta	

4.2.1.7 Requisito funcional 8

Tabla 14 Requisito funcional 8

Identificación del requisito:	RF08
Nombre del requisito:	Registrar Venta
Características:	Registrar Datos de una Venta
Descripción del requisito:	El sistema deberá permitir al usuario seleccionar los productos y añadirlos para realizar una venta debe calcular y guardar el total de la venta además de la fecha en que se realiza la venta y la cedula del cliente al que se realiza la venta.
Prioridad del requisito: Alta	

4.2.1.8 Requisito funcional 9

Tabla 15 Requisito funcional 9

Identificación del requisito:	RF09
Nombre del requisito:	Abrir Saldo
Características:	Registrar datos de un saldo para un cliente
Descripción del requisito:	El sistema deberá permitir que en el momento de realizar una venta el cliente tenga la opción para poder abrir un saldo con el valor de la venta y si lo desea poder realizar un abono del mismo.
Prioridad del requisito: Alta	

4.2.1.9 Requisito funcional 10

Tabla 16 Requisito funcional 10

Identificación del requisito:	RF10
Nombre del requisito:	Ver Inventario
Características:	Ver los datos de los productos en el inventario
Descripción del requisito:	El sistema deberá permitir al usuario poder visualizar los productos con lo que actualmente cuenta el negocio y deberá mostrar algunos datos como el nombre, marca, la cantidad disponible, el precio de venta y de entrada y el código del producto.
Prioridad del requisito:	Alta

4.2.1.9 Requisito funcional 11

Tabla 17 Requisito funcional 11

Identificación del requisito:	RF11
Nombre del requisito:	Filtrar Inventario
Características:	Mostrar los productos mediante un filtro de búsqueda.
Descripción del requisito:	El sistema deberá tener un filtro de búsqueda de los productos en la interfaz de inventario para hacer una consulta de los productos según el código, nombre, o según la cantidad de productos disponibles.
Prioridad del requisito:	Media

4.2.1.10 Requisito funcional 12

Tabla 18 Requisito funcional 12

Identificación del requisito:	RF12
Nombre del requisito:	Ver Historial de Ventas
Características:	Ver los datos de las ventas registradas.
Descripción del requisito:	El sistema deberá permitir al usuario poder visualizar las ventas que se encuentran registradas y deberá mostrar algunos datos como la fecha en que se realizó la venta, el total de la venta y mostrar algunos de los productos que se vendieron.
Prioridad del requisito:	Alta

4.2.1.11 Requisito funcional 13

Tabla 19 Requisito funcional 13

Identificación del requisito:	RF13
Nombre del requisito:	Filtrar Ventas
Características:	Mostrar las ventas mediante un filtro de búsqueda.
Descripción del requisito:	El sistema deberá permitir al usuario poder realizar un filtro a lista de las ventas realizadas por medio de una consulta de la fecha en que fue registrada la venta.
Prioridad del requisito:	Media

4.2.1.12 Requisito funcional 14

Tabla 20 Requisito funcional 14

Identificación del requisito:	RF14
Nombre del requisito:	Ver Historial de Saldos
Características:	Ver los datos de los saldos registrados.
Descripción del requisito:	El sistema deberá permitir al usuario poder visualizar los saldos que se encuentran registrados y deberá mostrar algunos datos como el código del cliente que debe el saldo, el total del saldo y el estado en que se encuentra.
Prioridad del requisito:	Alta

4.2.1.13 Requisito funcional 15

Tabla 21 Requisito funcional 15

Identificación del requisito:	RF15
Nombre del requisito:	Filtrar Saldos
Características:	Mostrar los saldos mediante un filtro de búsqueda.
Descripción del requisito:	El sistema deberá permitir al usuario realizar un filtro de búsqueda de los saldos en el sistema mediante el código del cliente o por el estado en el que se encuentra el saldo ya sea pagado o pendiente.
Prioridad del requisito:	Media

4.2.2 Requisitos no funcionales

4.2.2.1 Requisito no funcionales 1

Tabla 22 Requisito no funcional 1

Identificación del requisito:	RNF01
Nombre del requisito:	Encriptación de contraseña
Características:	Guardar la contraseña encriptada
Descripción del requisito:	El sistema deberá almacenar la contraseña encriptada del usuario en la base de datos.
Prioridad del requisito:	Media

4.2.2.2 Requisito no funcionales 2

Tabla 23 Requisito no funcional 2

Identificación del requisito:	RNF02
Nombre del requisito:	Interfaces intuitivas
Características:	- Manejo de colores - Menús sencillos - Opciones fáciles de identificar
Descripción del requisito:	El sistema deberá contar con una misma paleta de colores para todas las interfaces, el menú debe ser sencillo y las opciones fáciles de identificar.
Prioridad del requisito:	Media

4.2.2.3 Requisito no funcionales 3

Tabla 24 Requisito no funcional 3

Identificación del requisito:	RNF03
Nombre del requisito:	Alerta Formularios
Características:	Mostrar una alerta cuando se envían los datos desde un formulario
Descripción del requisito:	El sistema deberá mostrar mediante pantalla una alerta que indica si el ingreso y el envío de datos por medio de un formulario fue exitoso o hubo algún error.
Prioridad del requisito: Media	

4.2.2.4 Requisito no funcionales 4

Tabla 25 Requisito no funcional 4

Identificación del requisito:	RNF04
Nombre del requisito:	Base de Datos
Características:	Tipo Relacional
Descripción del requisito:	El sistema deberá almacenar los datos en una base de datos relacional PostgreSQL
Prioridad del requisito: Media	

4.3. Diseñar la aplicación web para la gestión de inventario de la tienda.

En este apartado se diseñaron los diferentes diagramas con lo son los de caso de uso, los diagramas de actividades, el diagrama entidad relación y se colocaron las capturas de los Wireframes para los diseños de las vistas de la aplicación web.

4.3.1. Diagrama de Casos de Uso

A continuación, se muestra los diagramas de los casos de uso según el tipo de usuario en relación con los requisitos funcionales.

4.3.1.1 Diagrama de caso uso administrador

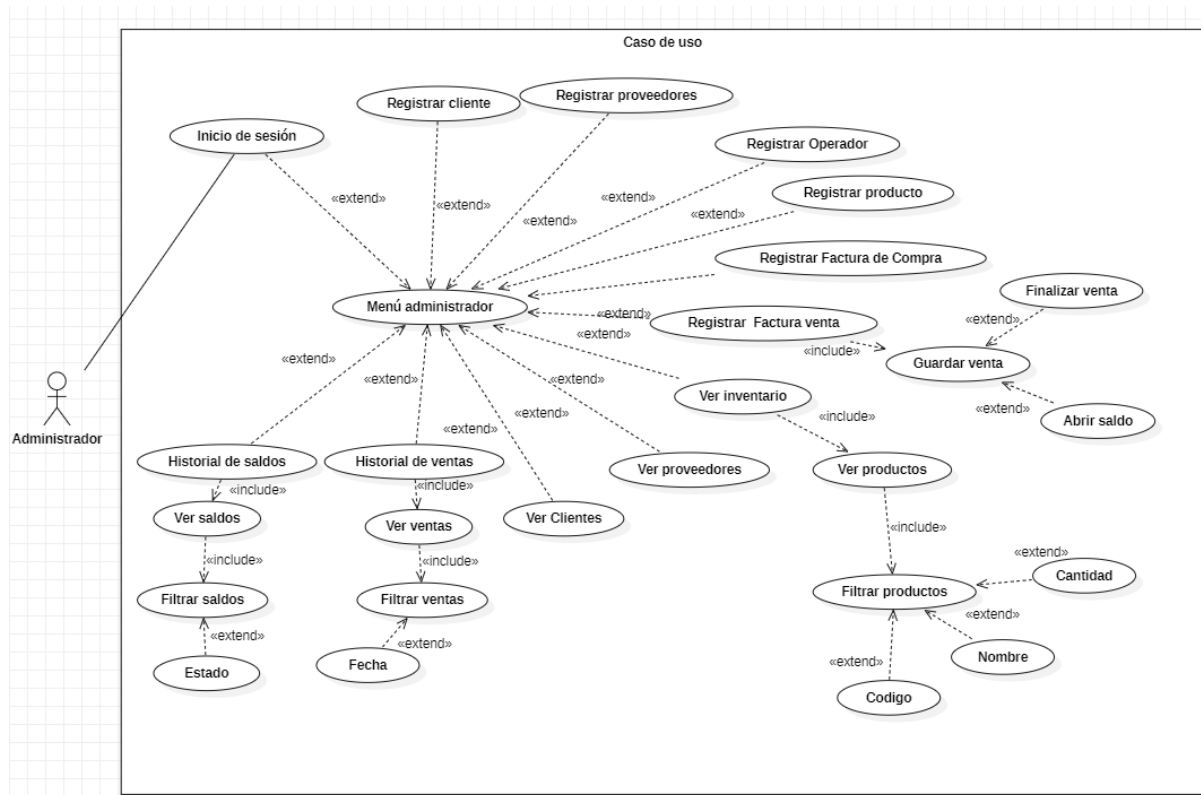


Ilustración 19 Diagrama de caso de uso administrador

En este diagrama se muestra la interacción que tiene el usuario de tipo administrador con el sistema en la que inicialmente empieza desde la vista de iniciar sesión, luego de validar el inicio de sesión el sistema deberá mostrar el menú que tiene como administrador en la que tiene las opciones de registrar un cliente, registrar proveedores, registrar operador, registrar

producto, registrar factura de compra, registrar factura de venta, ver inventario, ver clientes, ver proveedores, ver historial de ventas y ver el historial de saldos.

4.3.1.2 Diagrama de caso uso operador

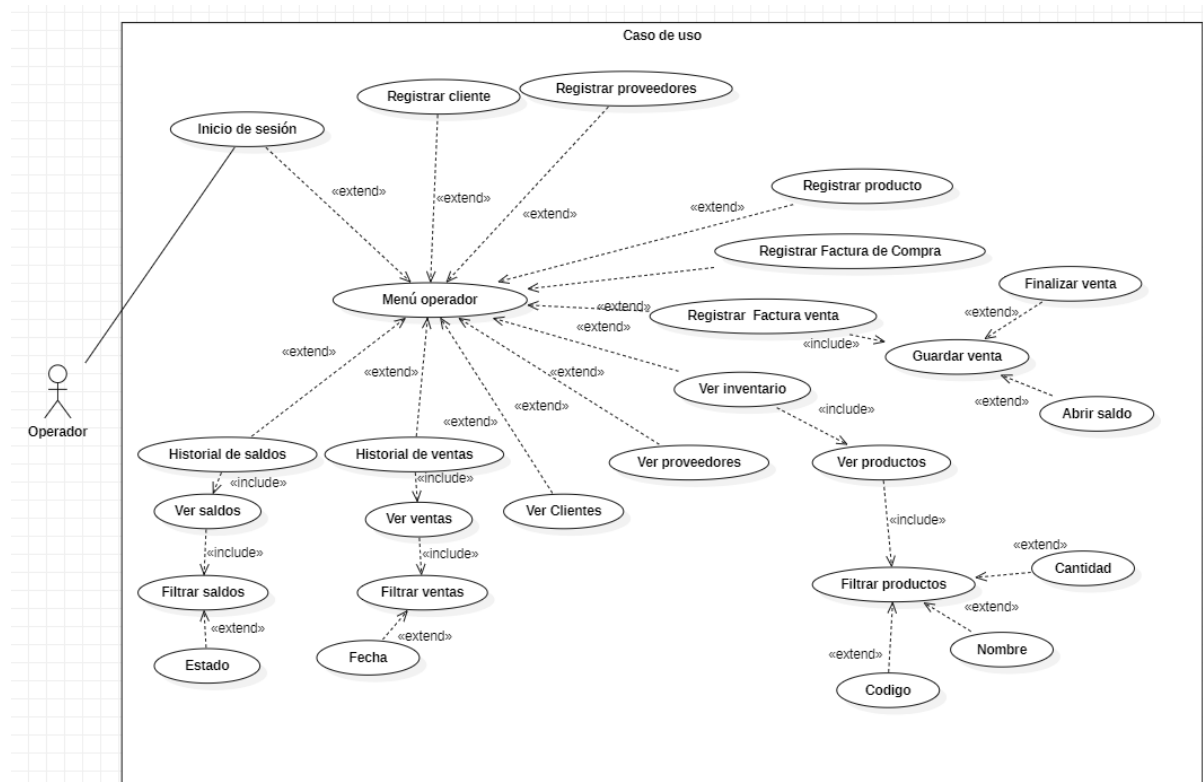


Ilustración 20 Diagrama de caso de uso operador

En este diagrama se muestra la interacción que tiene el usuario de tipo operador con el sistema en la que inicialmente empieza desde la vista de iniciar sesión, luego de validar el inicio de sesión el sistema deberá mostrar el menú de opciones que tiene como operador al igual que sucede con el diagrama anterior contiene las mismas opciones a excepción de la opción de registrar un operador.

4.2.2. Diagrama de Actividades

A continuación, se presenta los diferentes diagramas de actividades los cuales representan las funciones de las actividades y el flujo del proceso de negocio, para la elaboración de los distintos diagramas se ha hecho uso de la herramienta STARUML.

4.2.2.1 Diagrama de actividades inicio de sesión

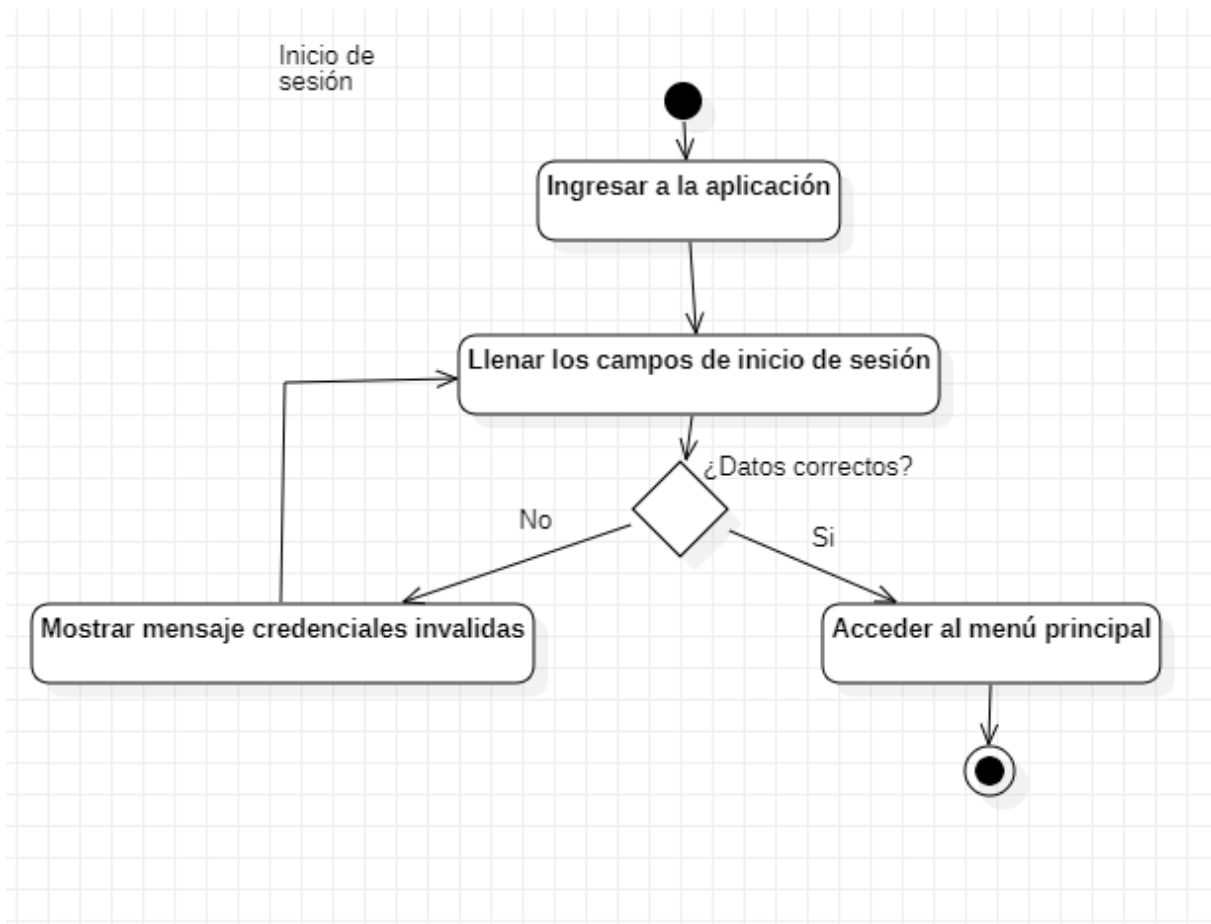


Ilustración 21 Diagrama de actividades Inicio de Sesión

En este diagrama se muestra las actividades necesarias para el inicio de sesión, para eso se debe ingresar a la aplicación, inicialmente saldrá un formulario el cual recibe los campos de usuario y contraseña se debe llenar este formulario e iniciar sesión, si los campos son correctos el sistema le permitirá acceder al menú principal en caso de que alguno de los campos sea erróneos el sistema deberá mostrar un mensaje el cual informe que las credenciales son

invalidas y por lo tanto no debe permitir el acceso al menú y deberá permanecer en la vista de iniciar sesión hasta que se digiten los campos correctos.

4.2.2.2 Diagrama de actividades registrar operador

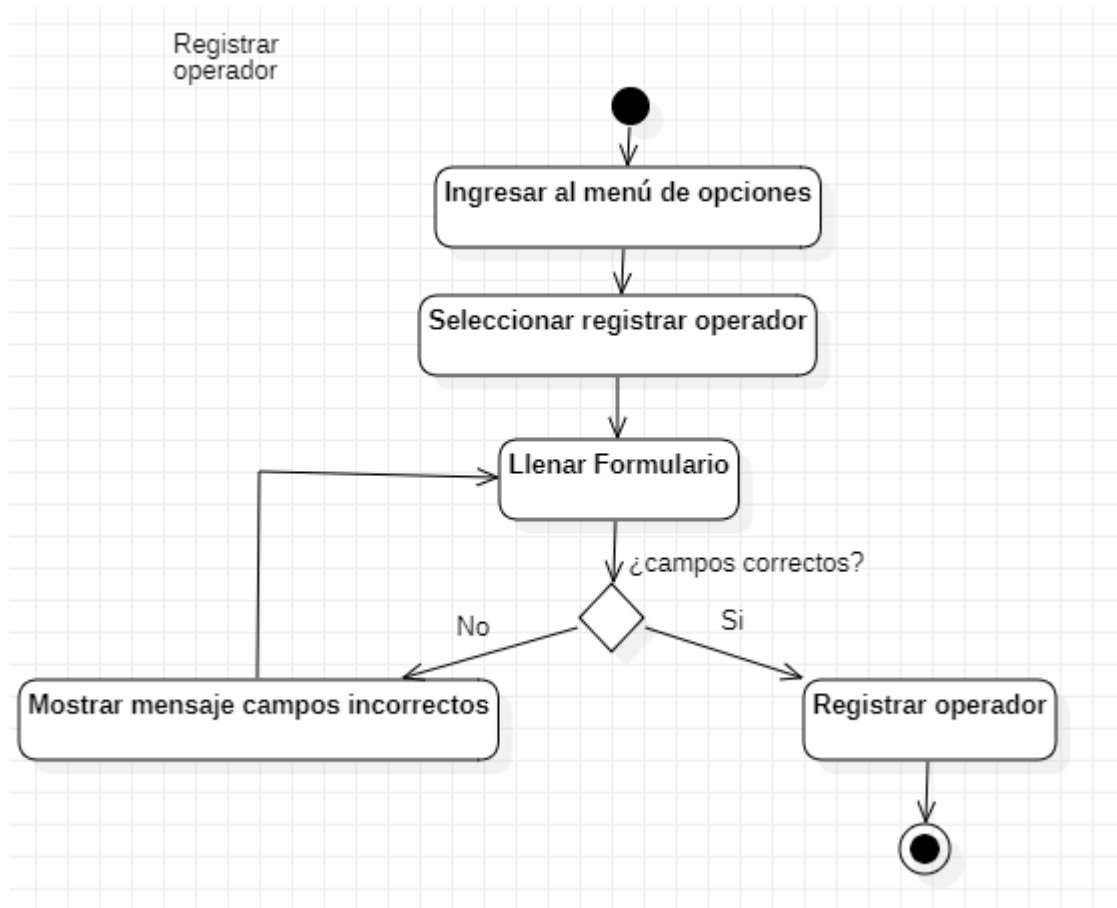


Ilustración 22 Diagrama de actividades Registrar operador

En este diagrama se muestra las actividades que son necesarias para poder registrar un operador inicialmente el usuario debe estar en el menú de opciones de entre las opciones se deberá seleccionar la opción de registrar un operador, luego de esto se vera la vista que permite registrar un operador a continuación se llenaran los campos que son solicitados desde el formulario en caso de que los campos que se han diligenciado en el formulario sean correctos se registrará un operador, en el caso contrario se mostrará un mensaje de alerta que informe que los campos son erróneos y no se debe permitir el registro hasta que los campos sean correctos.

4.2.2.3 Diagrama de actividades registrar cliente

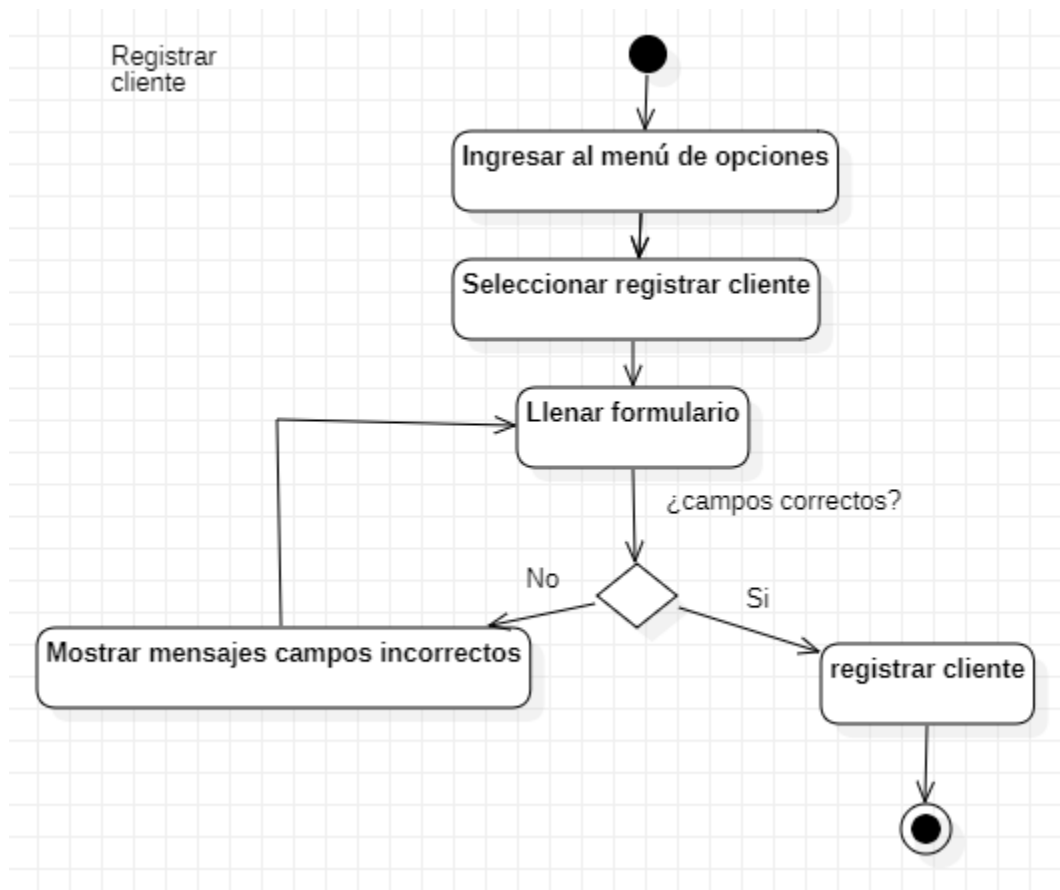


Ilustración 23 Diagrama de actividades Registrar Cliente

En este diagrama se muestra las actividades que son necesarias para poder registrar un cliente dentro del sistema, para empezar el usuario debe estar en el menú de opciones de entre las opciones se deberá seleccionar la opción de registrar cliente, a continuación se vera la vista que permite registrar un cliente se deberá llenar los campos necesarios del formulario, en caso de que los campos que se han diligenciado sean correctos se registrará un cliente, en el caso contrario se mostrará un mensaje de alerta que informe que los campos son erróneos y no se debe permitir el registro hasta que los campos sean correctos.

4.2.2.4 Diagrama de actividades registrar proveedor

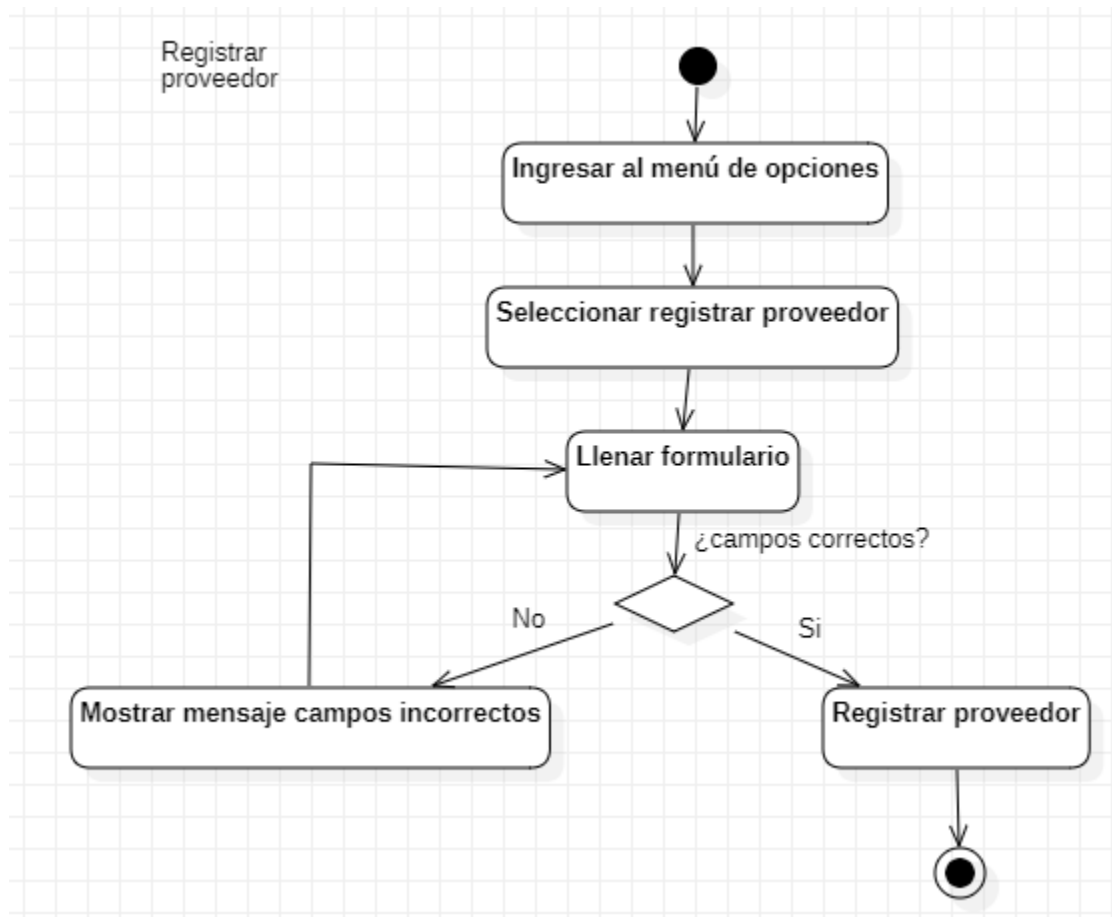


Ilustración 24 Diagrama de actividades Registrar proveedor

En este diagrama se muestra las actividades que permiten registrar un proveedor en el sistema, inicialmente el usuario debe estar en el menú de opciones dentro de las opciones posibles se deberá seleccionar la opción de registrar proveedor, a continuación el sistema le mostrará la vista de registrar proveedor donde se encontrara un formulario con los campos correspondientes para un proveedor, se deberá llenar el formulario y finalmente en caso de que los campos que sean correctos se registrará un proveedor, en caso contrario se mostrará un mensaje de alerta que informe que los campos son erróneos y no se debe permitir el registro hasta que los campos sean correctos.

4.2.2.5 Diagrama de actividades registrar producto

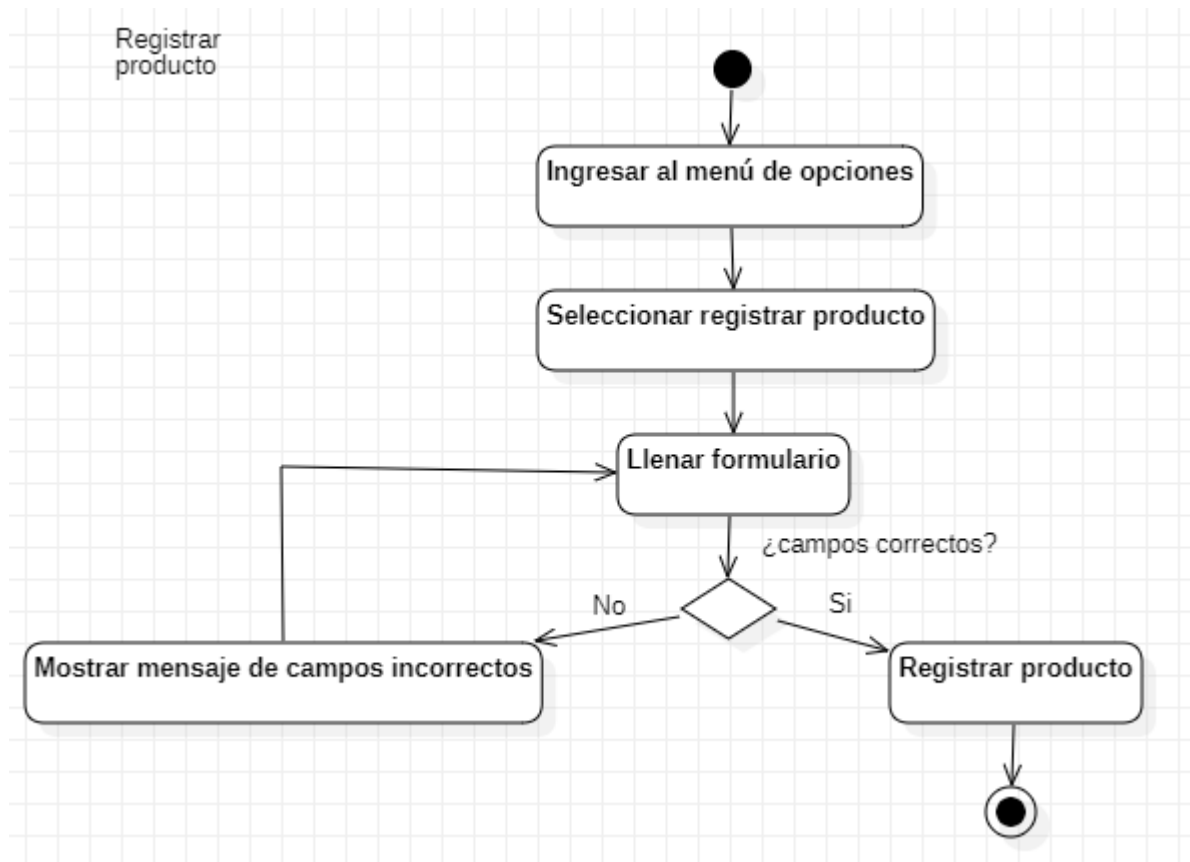


Ilustración 25 Diagrama de actividades Registrar producto

En este diagrama se muestra las actividades que se necesitan para poder registrar un producto para empezar el usuario debe estar en el menú de opciones luego seleccionar la opción de registrar producto, se abrirá la vista con el formulario que permite registrar un producto se deberá llenar los campos, en caso de que los campos que se han diligenciado sean correctos se registrará un producto, y si los campos están mal diligenciados se mostrará un mensaje de alerta que informe que los campos son erróneos y no se debe va hacer el registro del producto hasta que los campos sean correctos.

4.2.2.5 Diagrama de actividades registrar ingreso

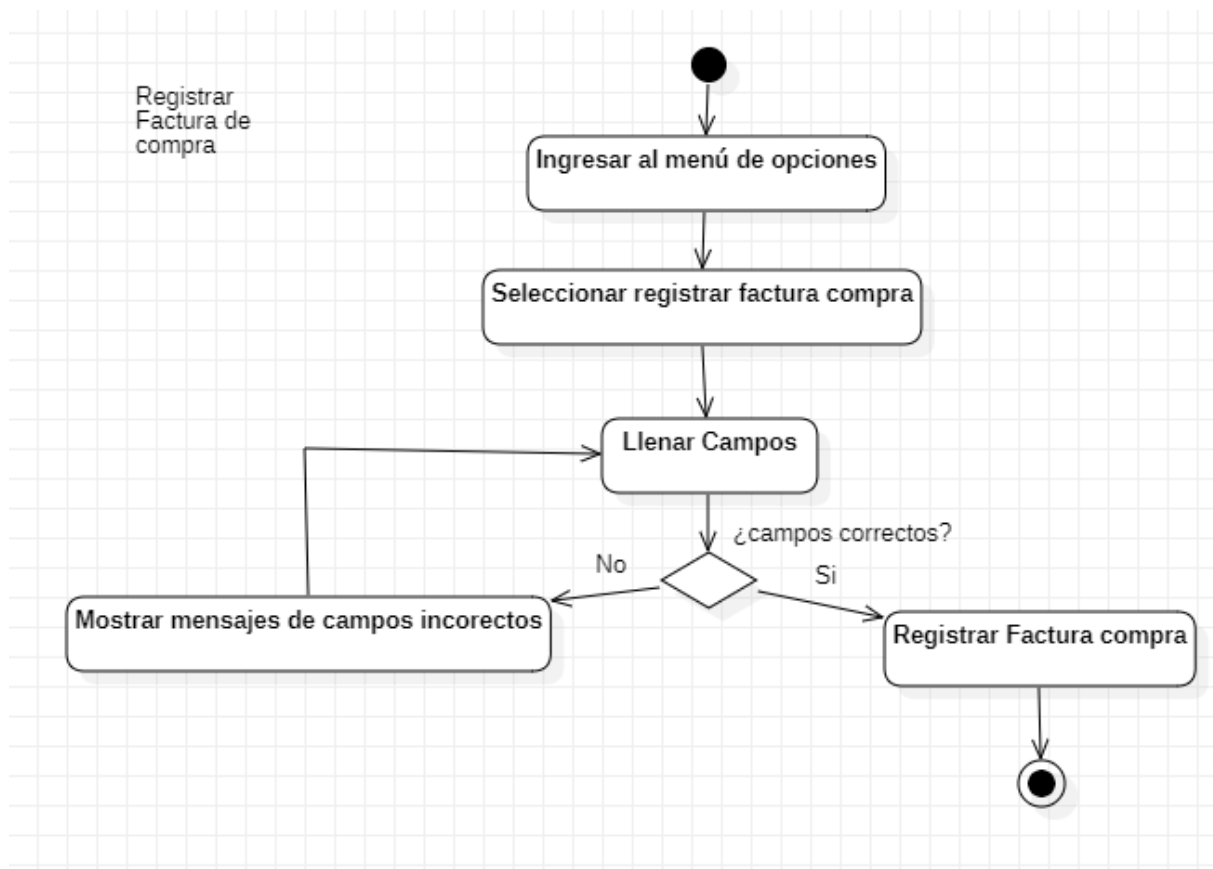


Ilustración 26 Diagrama de actividades Registrar Factura de compra

En este diagrama se muestra las actividades que son necesarias para poder registrar una factura de compra, para comenzar el usuario debe estar en el menú de opciones luego, de las opciones se deberá seleccionar la opción de registrar factura de ingreso, a continuación se vera la vista que permite registrar una factura, esta vista contiene los campos que son necesarios para registrar una factura de compra, se deberá llenar los campos, y en caso de que los campos que sean correctos se registrará los datos en el sistema, en el caso contrario se mostrará un mensaje de alerta que informe que los campos son erróneos y no se debe permitir el registro hasta que los campos sean correctos.

4.2.2.6 Diagrama de actividades registrar venta

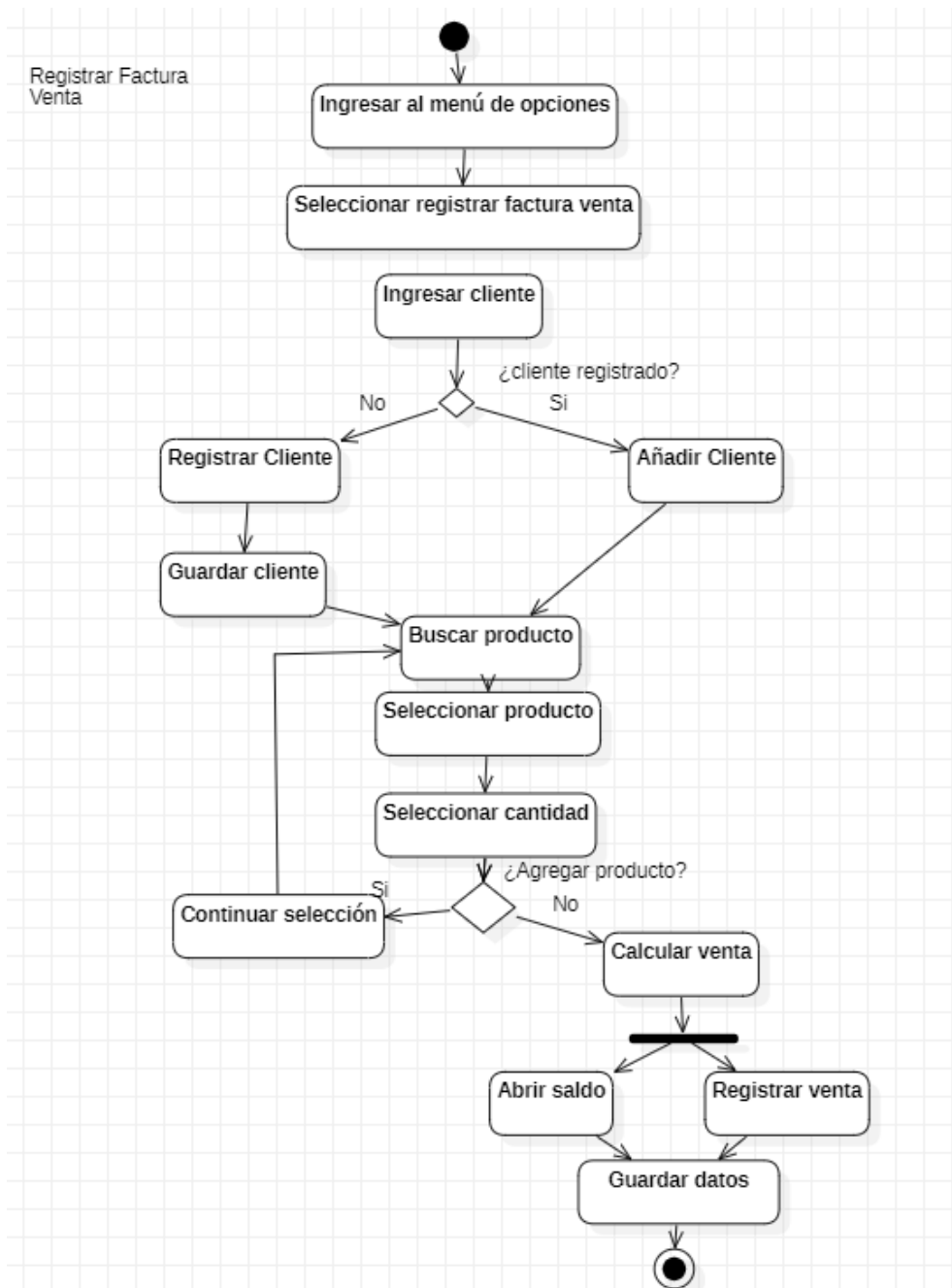


Ilustración 27 Diagrama de actividades Registrar venta

En este diagrama se muestra las actividades que son necesarias para poder hacer el registro de una factura de venta, inicialmente el usuario debe estar en el menú de opciones y seleccionar la opción de registrar una venta, a continuación se vera la vista que permite realizar la factura de venta se deberá llenar el campo que ingresar cliente si cliente no está registrado se

podrá hacer el registro de este y luego continuar con la factura de venta llenado el campo que permite la búsqueda del producto, luego seleccionar el producto y seleccionar la cantidad del mismo que se desea vender, luego de esto si aún se desea continuar con más productos para la venta se repite el mismo procedimiento hasta finalizar con la selección de productos, el sistema calculara la venta y de ahí salen dos opciones que el usuario puede realizar una de las opciones permite abrir un saldo a un cliente y la otra opción es la de registrar la venta finalmente cualquiera de las opciones acaba guardando la información de la factura de venta.

4.2.3 Diagrama entidad relación

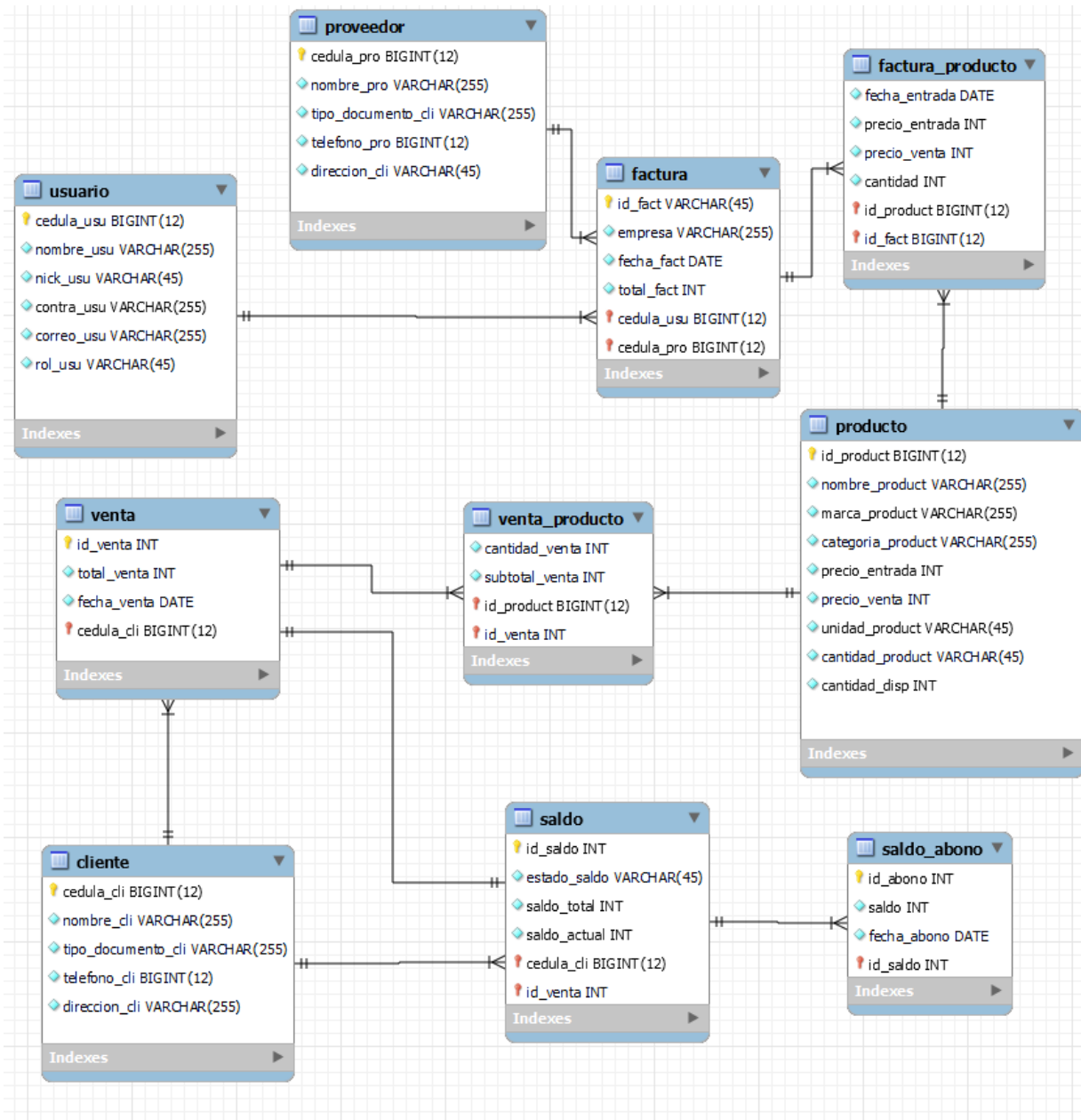


Ilustración 28 Diagrama entidad relación

4.2.4 Diccionario de la base de datos

En este apartado se muestra el diccionario de la base de datos, el cual es un repositorio de información sobre los datos que se están manejando en el cual se enlista de manera organizada los nombres, definiciones y características de cada uno de los campos o atributos de

una base de datos; A continuación, se presentan las tablas que contienen la información necesaria de cada entidad de la BD.

Tabla 26 Tabla entidad usuario

Tabla Usuario			
Nombre de atributo	Tipo	Restricción	Descripción
cedula_usu	Bigint	primary key	Clave única de la tabla que contiene la cedula del usuario.
nombre_usu	varchar (255)		Campo que contiene el nombre del usuario.
nick_usu	varchar (255)		Campo que contiene el Nick del usuario que se utiliza para iniciar sesión.
contra_usu	varchar (255)		Campo que guarda la contraseña el usuario de manera encriptada.
correo_usu	varchar (255)	unique	Campo que guarda el correo del usuario.
rol_usu	varchar (255)		Campo que contiene el rol que posee el usuario.

Tabla 27 Tabla entidad cliente

Tabla Cliente			
Nombre de atributo	Tipo	Restricción	Descripción
cedula_cli	Bigint	primary key	Clave única de la tabla cliente la cual almacena la cedula.
nombre_cli	varchar (255)		Campo que guarda el nombre del cliente.
tipo_documento_cli	varchar (255)		Campo que guarda el tipo de documento del cliente.
telefono_cli	bigint		Campo que guarda el teléfono del cliente.
direccion_cli	varchar (255)		Campo que contiene la dirección de residencia del cliente.

Tabla 28 Tabla entidad proveedor

Tabla Proveedor			
Nombre de atributo	Tipo	Restricción	Descripción
cedula_pro	Bigint	primary key	Clave única de la tabla que contiene la cedula del proveedor.
nombre_pro	varchar (255)		Campo que contiene el nombre del proveedor.
tipo_documento_pro	varchar (255)		Campo que contiene el tipo de documento del proveedor.
telefono_pro	bigint		Campo que contiene el teléfono del proveedor.
correo_pro	varchar (255)		Campo que contiene el correo del proveedor.

Tabla 29 Tabla entidad producto

Tabla Producto			
Nombre de atributo	Tipo	Restricción	Descripción
id_prouct	bigint	primary key	Clave única de la tabla producto que guarda el identificador de un producto.
nombre_product	varchar (255)		Campo que contiene el nombre del producto.
marca_product	varchar (255)		Campo que contiene la marca del producto.
categoria_product	varchar (255)		Campo que contiene la categoría a la que pertenece el producto.
precio_entrada	int		Campo que contiene el valor que cuesta el producto.
precio_venta	int		Campo que contiene el valor de un producto para el público.
unidad_product	varchar (45)		Campo que contiene la unidad de medida del producto.
cantidad_product	varchar (45)		Campo que contiene la cantidad de medida del producto.
cantidad_disp	int		Campo que contiene la cantidad de productos existentes en el inventario.

Tabla 30 tabla entidad lote

Tabla factura			
Nombre de atributo	Tipo	Restricción	Descripción
id_fac	varchar (45)	primary key	Clave única que contiene el identificador de una factura de compra.
empresa	varchar (255)		Campo que contiene el nombre de la empresa de la factura de compra.
fecha_fact	date		Campo que contiene la fecha de la factura.
total	int		Campo que contiene el total de la factura.
cedula_pro	int	foreing key	Campo que contiene la cedula del proveedor.
cedula_usu	varchar (45)	foreing key	Campo que contiene la cedula del usuario que registra la factura.

Tabla 31 Tabla entidad lote_producto

Tabla factura_producto			
Nombre de atributo	Tipo	Restricción	Descripción
fecha_entrada	date		Campo que la fecha de una factura de productos.
precio_entrada	int		Campo que contiene el precio de entrada de una factura de productos.
precio_venta	int		Campo que contiene el precio de venta de los productos de la factura.
cantidad	int		Campo que contiene la cantidad de productos de la factura.
id_fact	varchar (45)	foreing key	Campo que contiene el identificador de una factura de productos.
id_product	bigint	foreing key	Campo que contiene el identificador de un producto.

Tabla 32 Tabla entidad venta

Tabla venta			
Nombre de atributo	Tipo	Restricción	Descripción
id_venta	int	Primary key	Clave única de la tabla venta que guarda el identificador de una venta.
total_venta	int		Campo que contiene el total de una venta.
fecha_venta	date		Campo que guarda la fecha en que se realizó una venta.
cedula_cli	bigint	Foreing key	Campo que contiene el identificador de un cliente.

Tabla 33 Tabla entidad venta_producto

Tabla venta_producto			
Nombre de atributo	Tipo	Restricción	Descripción
id_venta	int	foreing key	Campo que contiene el id de una venta.
id_product	bigint	foreing key	Campo que contiene el id de un producto.
subtotal_venta	int		Campo que guarda el subtotal de una venta.
cantidad_venta	int		Campo que guarda la cantidad de producto vendidos en una venta.

Tabla 34 Tabla entidad saldo

Tabla saldo			
Nombre de atributo	Tipo	Restricción	Descripción
id_saldo	int	Primary key	Clave única que contiene el identificador de un saldo.
estado_saldo	varchar (45)		Campo que contiene en qué estado se encuentra un saldo.
saldo_total	int		Campo que contiene el total de un saldo.
saldo_actual	int		Campo que contiene el valor del saldo actual.
id_venta	int	Foreing key	Campo que contiene el id de una venta.
cedula_cli	int	Foreing key	Campo que contiene la cedula un cliente

Tabla 35 Tabla entidad saldo_abono

Tabla saldo_abono			
Nombre de atributo	Tipo	Restricción	Descripción
id_abono	int	Primary key	Clave única que contiene el identificador de un abono.
saldo	int		Campo que contiene el valor de un saldo.
fecha_abono	date		Campo que contiene la fecha de un abono.
id_saldo	int	Foreing key	Campo que contiene el id de un saldo.

4.2.5. Diseños en Figma

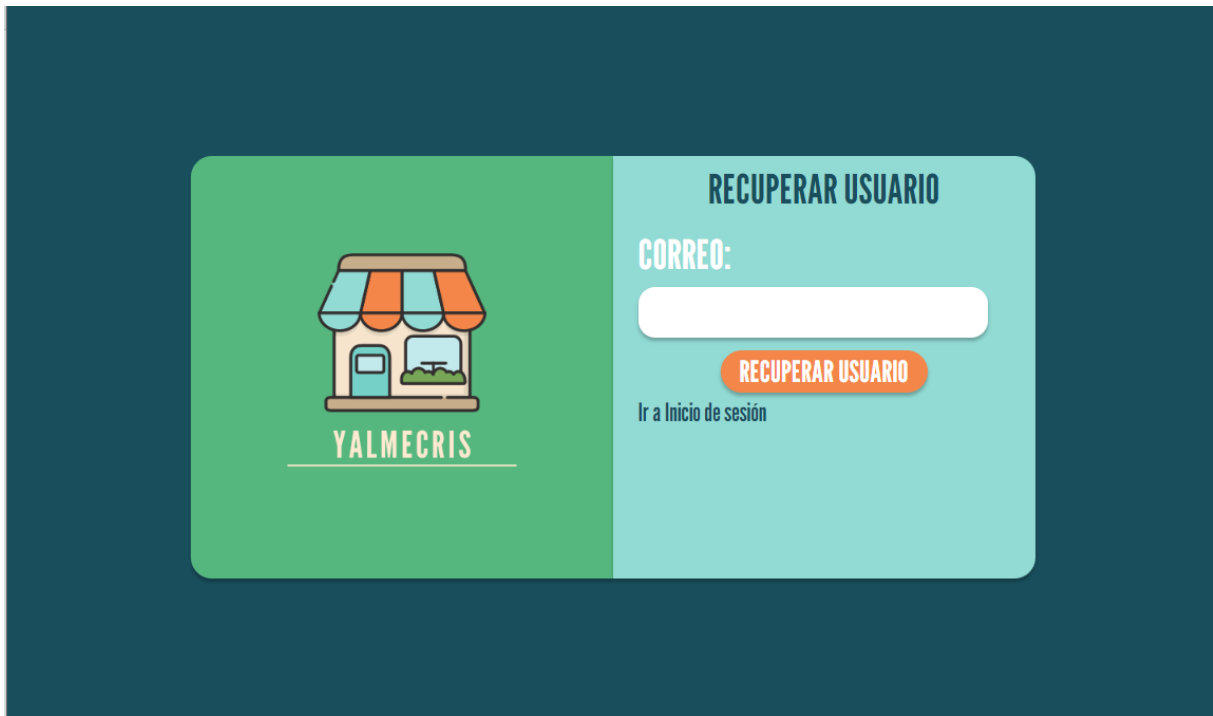
En este apartado se mostrará el resultado de la realización de los diseños de Wireframes mediante la herramienta Figma, esta herramienta es un editor de gráficos vectorial y una herramienta de generación de prototipos, principalmente basada en la web, herramienta que resulta útil para tener una idea de la cual partir para su posterior codificación

4.2.5.1 Wireframe Inicio de sesión



Ilustración 29 Wireframe Inicio de Sesión

4.2.5.2 Wireframe Recuperación de usuario



The wireframe for the 'Recuperación de usuario' page is set against a dark teal background. It features a central light blue rounded rectangle divided into two vertical panels. The left panel has a green background and contains a logo of a shop with an orange and blue striped awning, with the text 'YALMECRIS' below it. The right panel has a light blue background and is titled 'RECUPERAR USUARIO' in bold. Below the title is the label 'CORREO:' followed by a white input field. Under the input field is an orange button with the text 'RECUPERAR USUARIO'. At the bottom of the right panel is a link that says 'Ir a Inicio de sesión'.

Ilustración 30 Wireframe Recuperar Usuario

4.2.5.3 Wireframe Recuperación de contraseña



The wireframe for the 'Recuperación de contraseña' page is set against a dark teal background. It features a central light blue rounded rectangle divided into two vertical panels. The left panel has a green background and contains the same 'YALMECRIS' shop logo as the previous wireframe. The right panel has a light blue background and is titled 'RECUPERAR CONTRASEÑA' in bold. Below the title is the label 'CORREO:' followed by a white input field. Under the input field is an orange button with the text 'OBTENER CODIGO'. At the bottom of the right panel is a link that says 'Ir a Inicio de sesión'.

Ilustración 31 Wireframe Recuperar contraseña

4.2.5.3 Wireframe Validación de código

VALIDAR CODIGO

CODIGO:

VALIDAR

[Ir a Inicio de sesión](#)

Ilustración 32 Wireframe Validación de código

4.2.5.4 Wireframe Cambio de contraseña

CAMBIAR CONTRASEÑA

NUEVA CONTRASEÑA:

CONFIRMAR CONTRASEÑA:

GUARDAR

Ilustración 33 Wireframe Cambio de contraseña

4.2.5.2 Wireframe Menú de opciones administrador



Ilustración 34 Wireframe Menú de opciones administrador

4.2.5.3 Wireframe Menú de opciones operador



Ilustración 35 Wireframe Menú de opciones operador

4.2.5.4 Wireframe registrar cliente

Registrar Cliente



REGISTRAR CLIENTE



Nombre:

Tipo de Documento:

Numero de Documento:

Telefono:

Dirección:

REGISTRAR

Ilustración 36 Wireframe Registrar cliente

4.2.5.5 Wireframe registrar operador

The wireframe shows a registration form titled "REGISTRAR OPERADOR" within a green header bar labeled "YALMECRIS". The form is set against a dark teal background. It includes a user icon, input fields for "Nombre", "Cedula:", "Correo:", "Nick:", and "Contraseña:", and an orange "REGISTRAR" button.

YALMECRIS

REGISTRAR OPERADOR

 **Nombre**

Cedula:

Correo:

Nick:

Contraseña:

REGISTRAR

Ilustración 37 Wireframe Registrar operador

4.2.5.6 Wireframe registrar proveedor

Registrar Proveedor

YALMECRIS

REGISTRAR PROVEEDOR

 **Nombre:**

Tipo de Documento:

Numero de Documento:

Telefono:

Correo:

REGISTRAR

Ilustración 38 Wireframe Registrar proveedor

4.2.5.7 Wireframe registrar producto

**YALMECRIS**

REGISTRAR PRODUCTO

**Codigo ***

Categoria *

Nombre *

Marca *

Unidad de medida *

Cantidad de medida

Precio entrada *

Precio venta *

REGISTRAR

Ilustración 39 Wireframe Registrar producto

4.2.5.8 Wireframe registrar factura de compra

YALMECRIS

REGISTRO FACTURA INGRESO

Id de la factura:

Empresa:

Cedula del Proveedor:

Codigo del producto:

FACTURA COMPRA

Numero de factura:

Empresa:

Fecha de la factura: 12/15/2022

DATOS DEL NEGOCIO

Nombre: Yalmecris

Telefono: 5782288

Dirección: Av 12 entre 14 n an-08 villa nueva la isla

DATOS DEL PROVEEDOR

CC:

Nombre:

Telefono:

DATOS DEL USUARIO

CC: 1183235212

Nombre: Dimar

Cargo: Administrador

PRODUCTO	CODIGO	CAN	PRECIO UN	SUBTOTAL	ACCIÓN
----------	--------	-----	-----------	----------	--------

Total: COP 0.00

REGISTRAR FACTURA

Ilustración 40 Wireframe Registrar factura de compra

4.2.5.9 Wireframe realizar venta

YALMECRIS

VENTA

Buscar Producto

Cedula cliente **BUSCAR**

FACTURA VENTA

Numero de Venta: 15 Fecha de Venta: 12/13/2022

DATOS DEL NEGOCIO

Nombre: Yalmecris Telefono: 5782288 Dirección: Av 12 entre 14 n an-08 villa nueva la isla

DATOS DEL VENDEDOR

CC: 1193235212 Nombre: Dimar Cargo: Administrador

DATOS DEL CLIENTE

CC: 1004845349 Nombre: David

PRODUCTO	CODIGO	CAN	PRECIO UN	SUBTOTAL	ACCIÓN
Arroz Zulia 1	7709990854251	- 2 +	4000	8000	X

total: COP 8,000.00

REGISTRAR VENTA **ABRIR SALDO**

Ilustración 41 Realizar Venta

4.2.5.10 Wireframe añadir saldo

YALMECRIS

Añadir Saldo

NOMBRE	CODIGO	TELEFONO
Maria Bayona Castro	88245375	3208003712

Abono:

Total: \$9600
Abono: \$
Total: \$9600

Registrar Saldo

Ilustración 42 Añadir Saldo

4.2.5.10 Wireframe ver clientes



Ilustración 43 Ver clientes

4.2.5.11 Wireframe ver proveedores



Ilustración 44 Ver proveedores

4.2.5.12 Wireframe ver inventario



Ilustración 45 Wireframe Ver Inventario

4.2.5.13 Wireframe historial de ventas



Ilustración 46 Wireframe Historial de Ventas

4.2.5.13 Wireframe historial de saldos



Ilustración 47 Wireframe Historial saldos

4.2.5.14 Wireframe historial de saldos



Ilustración 48 Abonar Saldo

CAPITULO 5. IMPLEMENTACIÓN DE LA APLICACIÓN Y PRUEBAS

A continuación, se mostrará todo lo referente a como se realizó la implementación de la aplicación web, el desarrollo del backend de la aplicación en el cual se encuentran los servicios que contienen la lógica del negocio, el desarrollo del frontend en el cual se encuentra la parte grafica de la aplicación, o también conocido como la capa de presentación de la aplicación, así como las pruebas unitarias realizadas a los servicios expuestos en el backend para la comprobación del correcto funcionamiento del mismo.

5.1. Implementar la aplicación web que optimice la gestión y manejo de inventario de la tienda.

5.1.1. Desarrollo del backend

El backend de la aplicación web se realizó con NodeJS y ExpressJS, este último es un framework el cual proporciona características y herramientas para desarrollar aplicaciones de backend escalables. Para la creación del proyecto se hizo uso del comando de NodeJS “npm init” para la generación del archivo “package.json” el cual contendrá los metadatos de la API entre los cuales se puede encontrar las dependencias usadas para el funcionamiento del backend.

Posteriormente se hizo la creación de la estructura del proyecto la cual sigue el patrón Modelo-Vista-Controlador (MVC) en el cual se separa los datos (lógica de negocio) de la vista o capa de presentación. Debido a esto, en el backend se encuentran los controladores y el modelo que hace referencia al repositorio de la base de datos, además de las rutas las cuales sirven para organizar donde funcionarían los controladores.

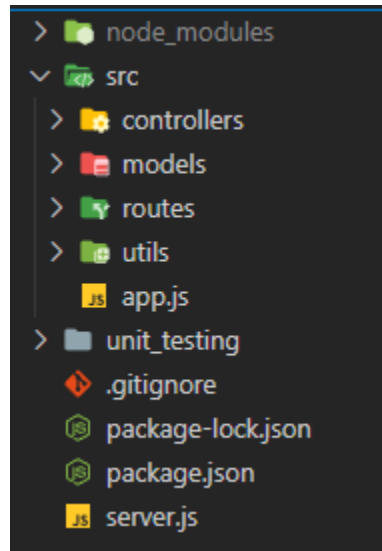


Ilustración 49. Estructura del proyecto en NodeJS - Elaboración propia

5.1.1.1 Codificación del proyecto

Una vez creada la estructura y archivos principales del proyecto se comenzará por la codificación del archivo que contendrá el servidor de la API, este es el archivo llamado “server.js”, en el cual se importa la aplicación ubicada dentro de la carpeta “src” del proyecto y con esta se ejecuta el método `listen()` el cual recibe como parámetros el puerto y una función la cual imprime un mensaje en consola confirmando el inicio del servidor como se ve en la siguiente ilustración.

```
server.js  X
server.js > ...
1  const app = require('./src/app');
2  const keys = require('./src/utils/keys');
3
4  app.listen(keys.port, ()=>{
5    console.log('Server iniciado en el puerto: ' + keys.port);
6  })
```

Ilustración 50 Server.js

Como se puede observar en la ilustración anterior el servidor se inicia haciendo uso de la importación de la app la cual como se mencionó anteriormente se encuentra en la carpeta

“src” del proyecto, este archivo se puede explicar de la siguiente manera. Primeramente, se encuentran las importaciones de las dependencias usadas en la API, así como el uso de “middlewares” los cuales son funciones que se ejecutan cada vez que se realiza una petición al servidor.



```
src > JS app.js > ...
1  const express = require('express');
2  const bodyParser = require('body-parser');
3  const cors = require('cors');
4  const app = express();
5
6  //middlewares
7  app.use(bodyParser.json());
8  app.use(bodyParser.urlencoded({extended: false}));
9  app.use(cors());
10
```

Ilustración 51 App.js

Luego se encuentra la importación de las rutas ubicadas en la carpeta “routes” del proyecto, las cuales contienen sus respectivos endpoints que hacen referencia a las funcionalidades de la lógica del negocio a las cuales se acceden mediante peticiones http. Posterior a la importación de los archivos que contienen las rutas, se especifica el path o ruta principal en la cual estarán disponibles los endpoints previamente mencionados como se observa en la siguiente ilustración.

```

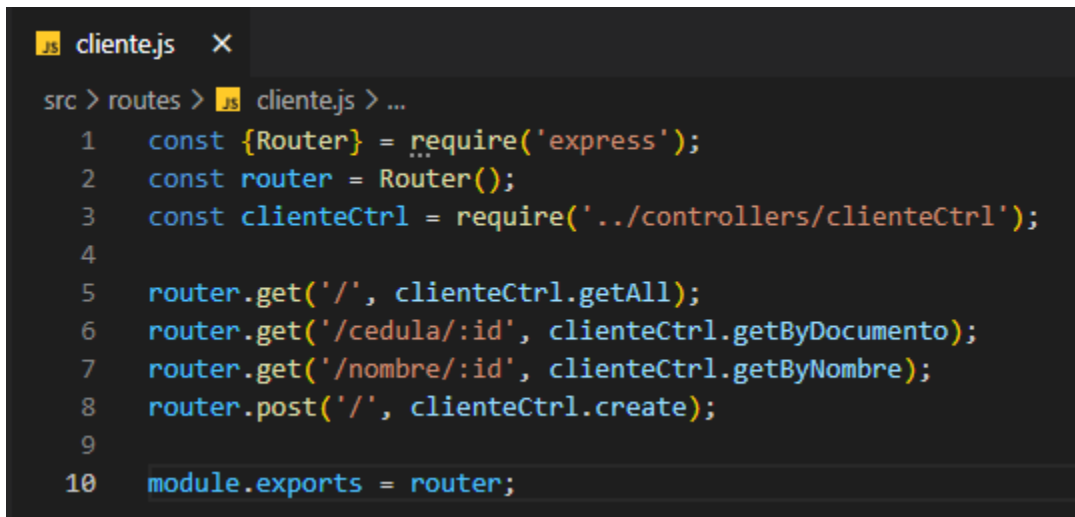
11 //routes
12 const usuarioRoutes = require('./routes/usuario');
13 const clienteRoutes = require('./routes/cliente');
14 const proveedorRoutes = require('./routes/proveedor');
15 const productoRoutes = require('./routes/producto');
16 const loteRoutes = require('./routes/lote');
17 const loteproductoRoutes = require('./routes/loteproducto');
18 const ventaRoutes = require('./routes/venta');
19 const saldoRoutes = require('./routes/saldo');
20
21 //endpoints
22 app.use('/usuario', usuarioRoutes);
23 app.use('/cliente', clienteRoutes);
24 app.use('/proveedor', proveedorRoutes);
25 app.use('/producto', productoRoutes);
26 app.use('/lote', loteRoutes);
27 app.use('/loteP', loteproductoRoutes);
28 app.use('/venta', ventaRoutes);
29 app.use('/saldo', saldoRoutes);
30
31 module.exports = app;

```

Ilustración 52 Rutas de los endpoints

5.1.1.2 Rutas

Como se mencionó anteriormente, las rutas contienen los endpoints de las funcionalidades del sistema, dichos endpoints solo pueden ser accedidos mediante peticiones http a través de los métodos de solicitud como lo son GET, POST, PUT y DELETE, esto se hace con el fin de diferenciar la funcionalidad que realiza en cada endpoint. A continuación, se mostrará la codificación de la ruta de “cliente”.

A screenshot of a code editor window titled 'cliente.js'. The editor shows the following JavaScript code:

```
src > routes > cliente.js > ...
1  const {Router} = require('express');
2  const router = Router();
3  const clienteCtrl = require('../controllers/clienteCtrl');
4
5  router.get('/', clienteCtrl.getAll);
6  router.get('/cedula/:id', clienteCtrl.getByDocumento);
7  router.get('/nombre/:id', clienteCtrl.getByNombre);
8  router.post('/', clienteCtrl.create);
9
10 module.exports = router;
```

Ilustración 53 cliente.js

En la ilustración anterior podemos observar que se importa el router a través de express esto se hace con el fin de poder definir los endpoints que se expondrán en cada ruta. Además, se importa el controlador del cliente llamado “clienteCtrl” ya que en este se encuentra la lógica del negocio. Para definir un endpoint se hace uso del router, seguido de un punto (.) luego se especifica el método http por el cual se podrá acceder a dicho endpoint, y este último recibe dos parámetros los cuales son el path o ruta del endpoint y el método del controlador que se quiere ejecutar. Por último, se exporta el router para que pueda ser utilizado en el archivo app del proyecto.

5.1.1.3 Controladores

Como se ha mencionado anteriormente los controladores contienen la lógica de negocio la cual hace referencias a las acciones y operaciones que se realizan sobre la base datos. A continuación, se mostrará la codificación realizada en el controlador de cliente.

```

JS clienteCtrl.js X
src > controllers > JS clienteCtrl.js > ...
1  const pool = require('../models/repository_postgre');
2
3  > exports.create = async (req, res) => { ...
20 }
21
22 exports.getAll = async (req, res) => {
23   try {
24     const response = await pool.query(`SELECT * FROM cliente`);
25     res.status(200).send({ success: true, body: response.rows});
26   } catch (error) {
27     res.status(500).send({success: false, body: error});
28   }
29 }
30
31 > exports.getByDocumento = async (req, res) => { ...
39 }
40
41 > exports.getByNombre = async (req, res) => { ...
49 }

```

Ilustración 54 Controlador del cliente

Como se puede observar en el controlador del cliente se realiza la importación del repositorio de PostgreSQL el cual contiene la conexión con la base de datos, y a través de este se pueden realizar consultas SQL que permiten la comunicación entre el backend y la base de datos, posterior a esto se encuentran los métodos los cuales contienen pequeños fragmentos de código que se encargan de realizar una funcionalidad en específico, en la ilustración se puede observar el método “getAll” el cual se encarga de enviar una consulta SQL la cual debe obtener todos los registros de la tabla cliente, dicha información se almacena en la constante declarada con el nombre “response” para luego ser retornada en la respuesta de la petición con un código de estado exitoso. En caso de ocurrir algún fallo, se responderá a la petición con un código de error y el mensaje del error en el cuerpo de la respuesta.

5.1.2. Desarrollo del frontend

El frontend de la aplicación web se realizó con Vue.js y Axios, como ya se ha mencionado Vue es un framework de desarrollo enfocado a la construcción de interfaces de usuario mediante la creación de componentes y Axios es una librería JavaScript que nos permite hacer peticiones HTTP a un servidor. Para la creación de proyecto se usó el comando “vue create app-inventario” donde app-inventario hace referencia al nombre del proyecto, lo siguiente es seleccionar la configuración del archivo

```
? Please pick a preset: (Use arrow keys)
> vue-preset ([Vue 3] babel, router)
  Default ([Vue 2] babel, eslint)
  Default (Vue 3) ([Vue 3] babel, eslint)
  Manually select features
```

Ilustración 55 Configuración Vue.js

La configuración del proyecto seleccionada fue vue-preset esta configuración es una guardada con anterioridad la cual tiene seleccionada las opciones de Vue 3, babel y router, luego de esto Vue se encargará de instalar las dependencias necesarias para iniciar el proyecto y la estructura que resulta es la siguiente

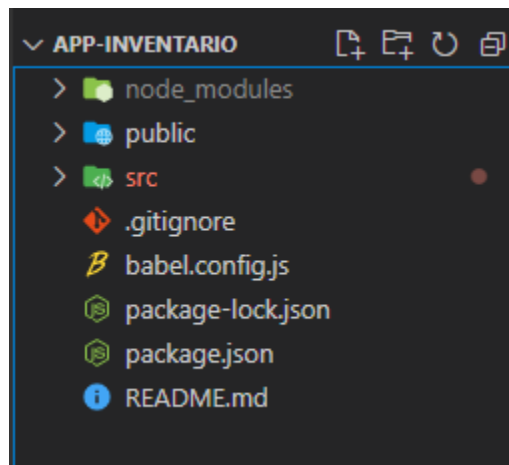
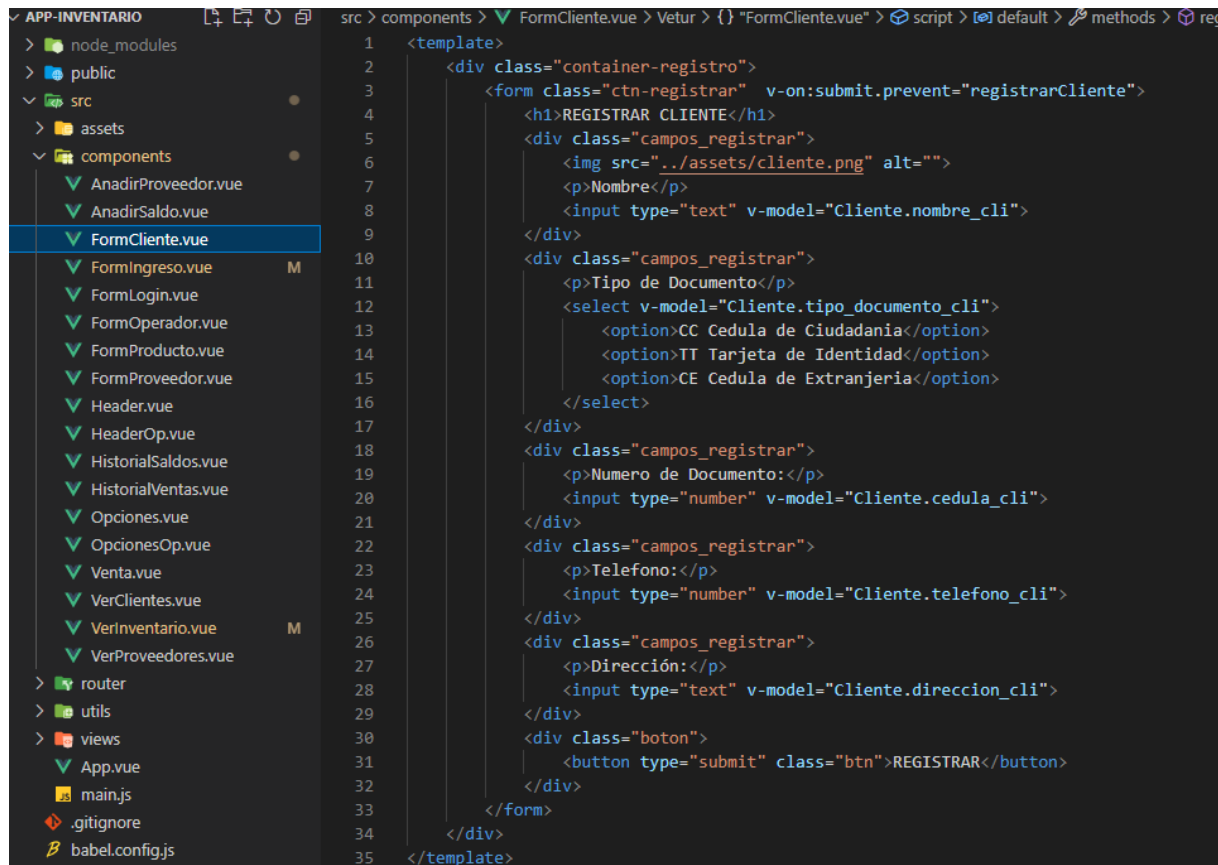


Ilustración 56 Estructura del proyecto en Vue.js

5.1.2.1 Codificación del frontend

Una vez creado el proyecto lo siguiente será la creación de los componentes a continuación se evidencia el componente de contiene el formulario para el registro de un cliente



```
1 <template>
2   <div class="container-registro">
3     <form class="ctn-registrar" v-on:submit.prevent="registrarCliente">
4       <h1>REGISTRAR CLIENTE</h1>
5       <div class="campos_registrar">
6         
7         <p>Nombre</p>
8         <input type="text" v-model="Cliente.nombre_cli">
9       </div>
10      <div class="campos_registrar">
11        <p>Tipo de Documento</p>
12        <select v-model="Cliente.tipo_documento_cli">
13          <option>CC Cedula de Ciudadania</option>
14          <option>TI Tarjeta de Identidad</option>
15          <option>CE Cedula de Extranjeria</option>
16        </select>
17      </div>
18      <div class="campos_registrar">
19        <p>Numero de Documento:</p>
20        <input type="number" v-model="Cliente.cedula_cli">
21      </div>
22      <div class="campos_registrar">
23        <p>Telefono:</p>
24        <input type="number" v-model="Cliente.telefono_cli">
25      </div>
26      <div class="campos_registrar">
27        <p>Dirección:</p>
28        <input type="text" v-model="Cliente.direccion_cli">
29      </div>
30      <div class="boton">
31        <button type="submit" class="btn">REGISTRAR</button>
32      </div>
33    </form>
34  </div>
35 </template>
```

Ilustración 57 Componente Cliente

Un componente en Vue se separa en tres partes, una primera es el template con contiene la estructura HTML, una siguiente es el SCRIP donde se define el nombre del componente para ser exportado y contiene los métodos y desde donde se puede hacer el consumo del servicio de la API, y finalmente tiene un STYLE, aunque este puede ser opcional ya que Vue permite la reutilización de código CSS.

Para la utilización de un componente se necesita que este sea llamado desde una vista como en la imagen a continuación

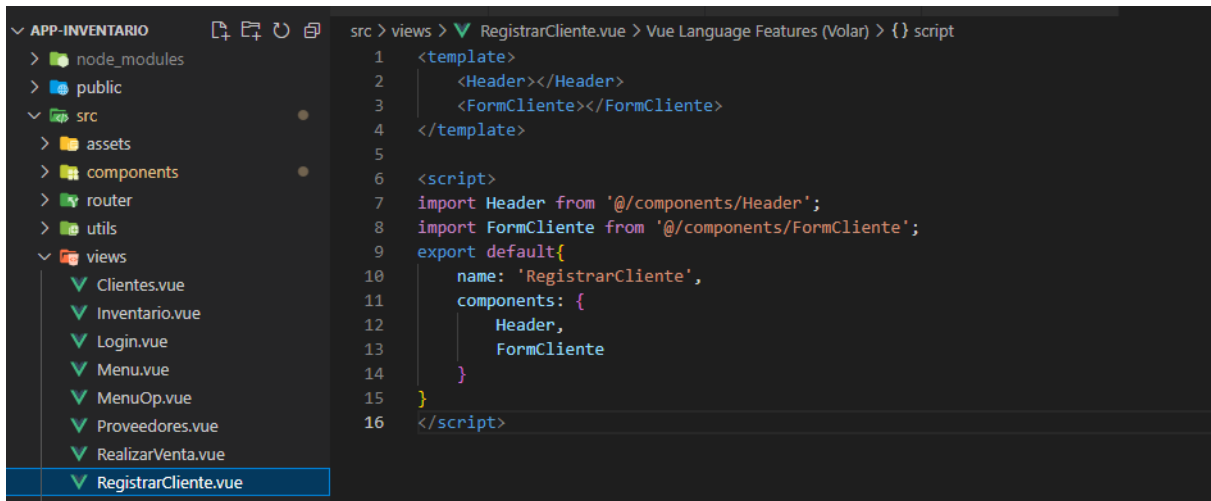


Ilustración 58 View Cliente

En las líneas 7 y 8 se importan los componentes Header y FormCliente los cuales son llamados en la vista en las líneas 2 y 3.

Para poder visualizar una vista con los componentes se necesita exportar la vista y ser llamada desde el router

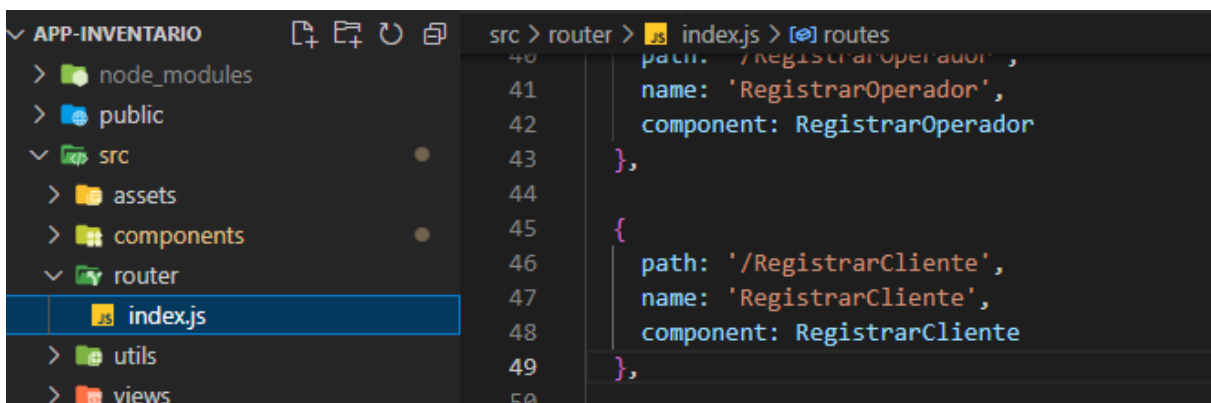


Ilustración 59 index.js

En la carpeta de Router en el archivo index.js se define la vista que se va a llamar y como será el path de la ruta.

5.2. Validar el correcto funcionamiento de la aplicación web con respecto a los requerimientos solicitados por el cliente mediante pruebas unitarias.

A continuación, se mostrará como se realizaron las pruebas unitarias a los endpoints más importantes de los servicios que fueron creados en el backend de la aplicación web.

5.2.1 Instalación y configuración de las librerías para los test

Se realizó la instalación de la librería “jest” la cual es un framework para la ejecución de pruebas que permite realizar tests de forma rápida y sencilla al backend. Además, se instaló “supertest” lo cual nos permite realizar peticiones HTTP a la API, para la instalación se necesitó usar el comando “npm install -D jest supertest”.

Una vez instalada la librería se creó una carpeta llamada “unit_testing” donde tendrá los archivos con las pruebas a los endpoints de cada servicio.

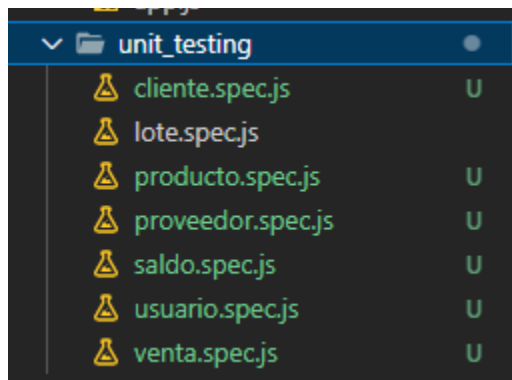


Ilustración 60 carpeta unit_testing

A continuación, se debe configurar el archivo package.json para poder realizar las pruebas unitarias que han sido creadas para los servicios de endpoints, para esto se necesitó crear un nuevo script que se llamó “test”

```
Debug
"scripts": {
  "start": "node .",
  "start:dev": "nodemon .",
  "test": "jest"
},
```

Ilustración 61 package.json

5.2.2 Realización de pruebas

5.2.2.1 Pruebas al Backend

A continuación, se va a ver un ejemplo de la codificación y resultado de la ejecución de una prueba unitaria al endpoint

```
unit_testing > producto.spec.js > ...
1  const request = require('supertest');
2  const app = require('../src/app');
3
4  describe('Servicio producto', ()=>{
5      test('deberia consultar todos los productos', async ()=>{
6          const res = await request(app)
7              .get('/producto/')
8              .send();
9          expect(res.statusCode).toEqual(200);
10         expect(res.body.success).toEqual(true);
11     })
12
13     test('deberia consultar producto por id', async ()=>{
14         const res = await request(app)
15             .get('/producto/id/'+7709990)
16             .send();
17         expect(res.statusCode).toEqual(200);
18         expect(res.body.success).toEqual(true);
19         expect(res.body.body[0].id_producto).toEqual('7709990');
20     })
21
22     test('deberia consultar todos los productos por nombre', async ()=>{
23         const res = await request(app)
24             .get('/producto/nombre/'+ 'Desodorante')
25             .send();
26         expect(res.statusCode).toEqual(200);
27         expect(res.body.success).toEqual(true);
28     })
29 })
30
```

Ilustración 62 Prueba Unitaria a los endpoint de producto

En la imagen anterior se puede observar que se está realizando unas pruebas de consulta para los diferentes endpoints de la tabla producto se está realizando una consulta de todos los productos, la siguiente es por id y la última es una consulta por nombre del producto.

5.2.3 Ejecución de pruebas

5.2.3.1 prueba del servicio cliente

```
PASS unit_testing/cliente.spec.js
Servicio cliente
  ✓ debería consultar todos los clientes (983 ms)
  ✓ debería consultar cliente por cedula (116 ms)
  ✓ debería consultar todos los clientes por nombre (117 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        2.369 s
Ran all test suites.
```

Ilustración 63 Resultado testing servicio cliente

5.2.3.2 prueba del servicio lote

```
PASS unit_testing/lote.spec.js
Servicio lote
  ✓ debería consultar todos los lotes (933 ms)
  ✓ debería consultar lote por id (115 ms)
  ✓ debería consultar todos los lotes por nombre (178 ms)
  ✓ debería consultar todos los lotes por cantidad disponible (108 ms)

Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        2.525 s
Ran all test suites.
```

Ilustración 64 Resultado testing servicio lote

5.2.3.3 prueba del servicio producto

```
PASS unit_testing/producto.spec.js
Servicio producto
  ✓ debería consultar todos los productos (1003 ms)
  ✓ debería consultar producto por id (140 ms)
  ✓ debería consultar todos los productos por nombre (107 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        2.355 s, estimated 3 s
Ran all test suites.
```

Ilustración 65 Resultado testing servicio producto

5.2.3.4 prueba del servicio proveedor

```
PASS unit_testing/proveedor.spec.js
  Servicio proveedor
    ✓ debería consultar todos los proveedores (910 ms)
    ✓ debería consultar proveedor por cedula (114 ms)
    ✓ debería consultar todos los proveedores por nombre (126 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        2.3 s
Ran all test suites.
```

Ilustración 66 Resultado testing servicio proveedor

5.2.3.5 prueba del servicio usuario

```
PASS unit_testing/usuario.spec.js
  Servicio usuario
    ✓ debería consultar todos los usuarios (964 ms)
    ✓ debería iniciar sesion un usuario (135 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        2.255 s
Ran all test suites.
```

Ilustración 67 Resultado testing servicio usuario

5.2.3.6 prueba del servicio venta

```
PASS unit_testing/venta.spec.js
  Servicio venta
    ✓ debería consultar todas los ventas (831 ms)
    ✓ debería consultar las ventas por filtro (103 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        2.137 s
Ran all test suites.
```

Ilustración 68 Resultado testing servicio venta

5.2.3.7 prueba del servicio saldo

```
PASS unit_testing/saldo.spec.js
  Servicio saldo
    ✓ debería consultar todos los saldos (937 ms)
    ✓ debería consultar los saldos por filtro (113 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        2.229 s
Ran all test suites.
```

Ilustración 69 Resultado testing servicio saldo

5.2.2.2 Pruebas al frontend

A continuación, se va ver las pruebas y validaciones que se realizaron a la aplicación web desde la parte del frontend principalmente las validaciones corresponden a los formularios que se manejan en las diferentes vistas.

5.2.2.2.1 Pruebas en Login

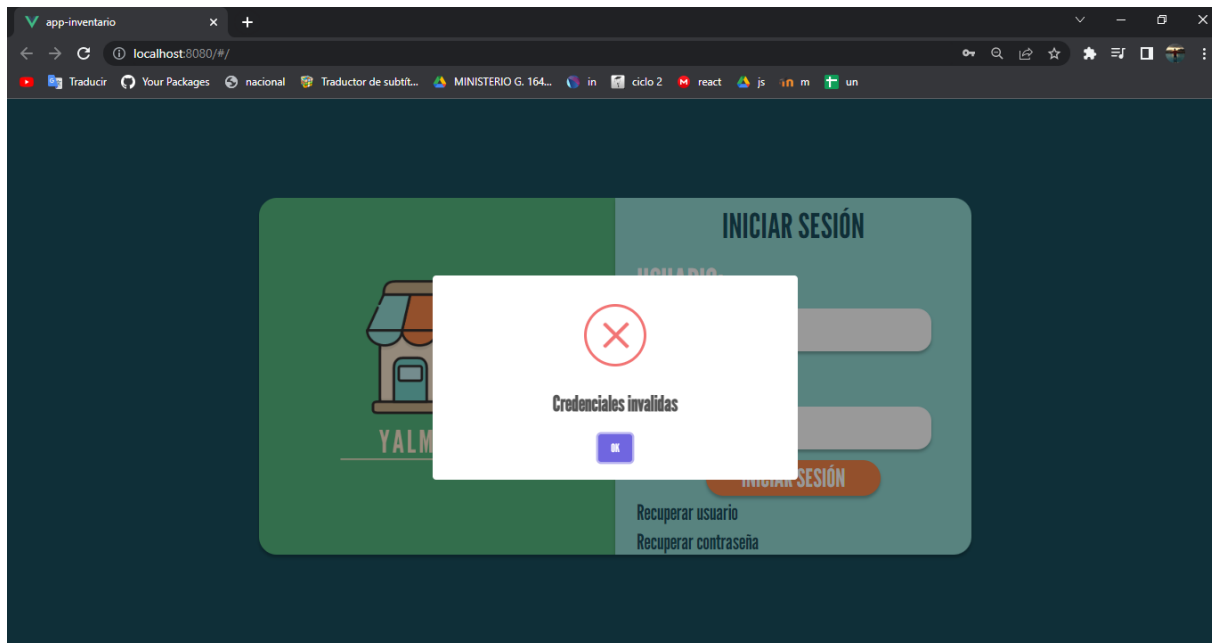


Ilustración 70 Credenciales Invalidas en Login

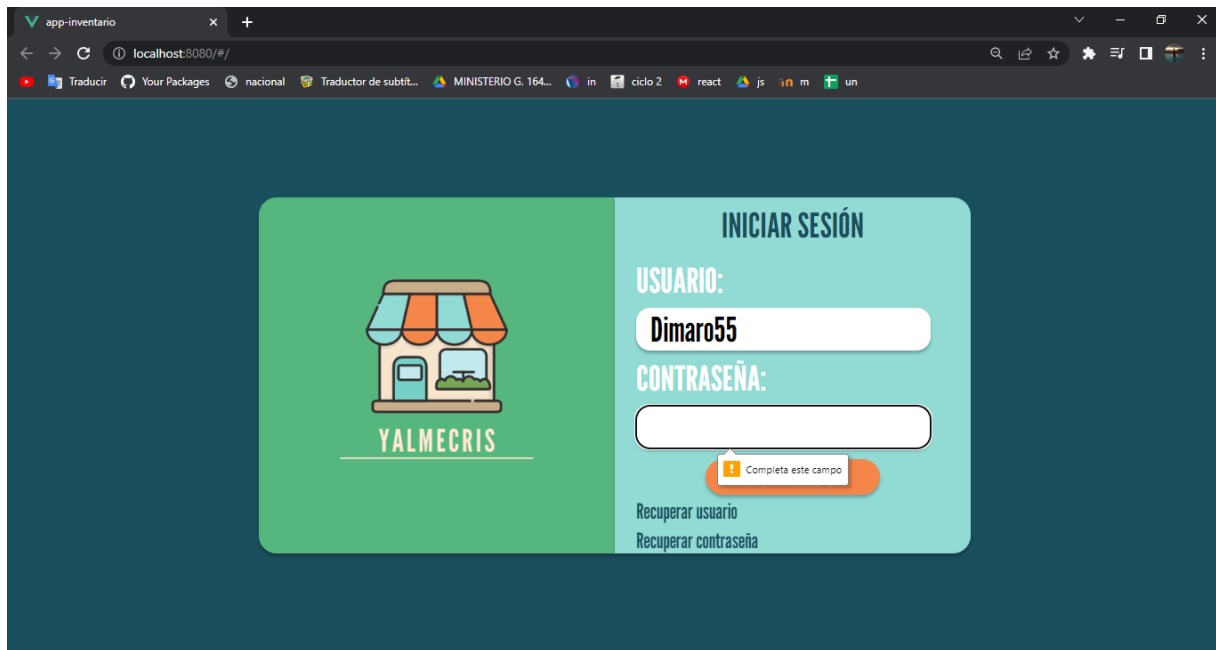


Ilustración 71 Campos incompletos en Login

5.2.2.2 Recuperar Usuario

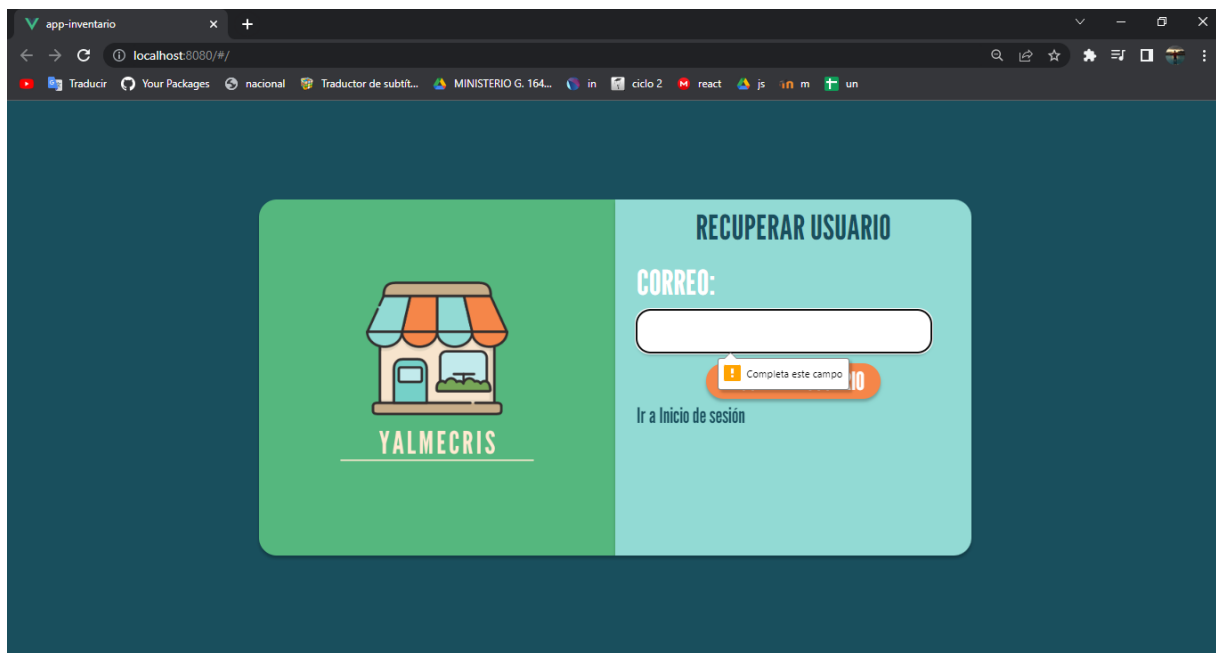


Ilustración 72 Campo incompleto en Recuperar Usuario

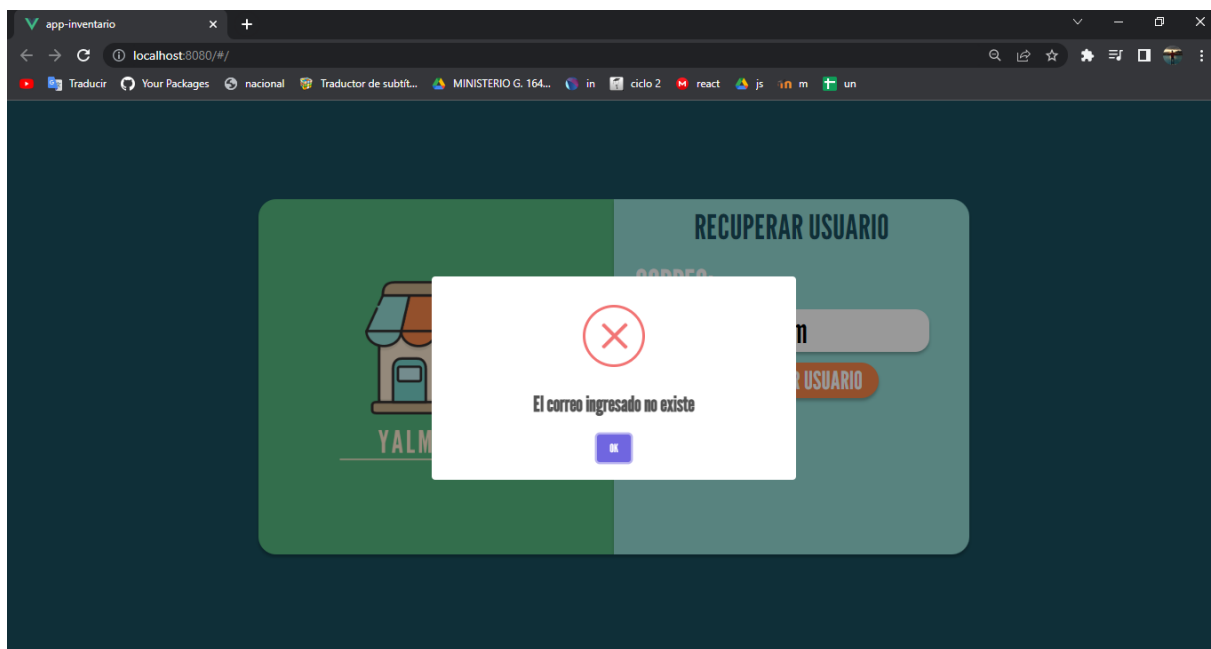


Ilustración 73 Correo no existe

5.2.2.2.3 Recuperar Contraseña

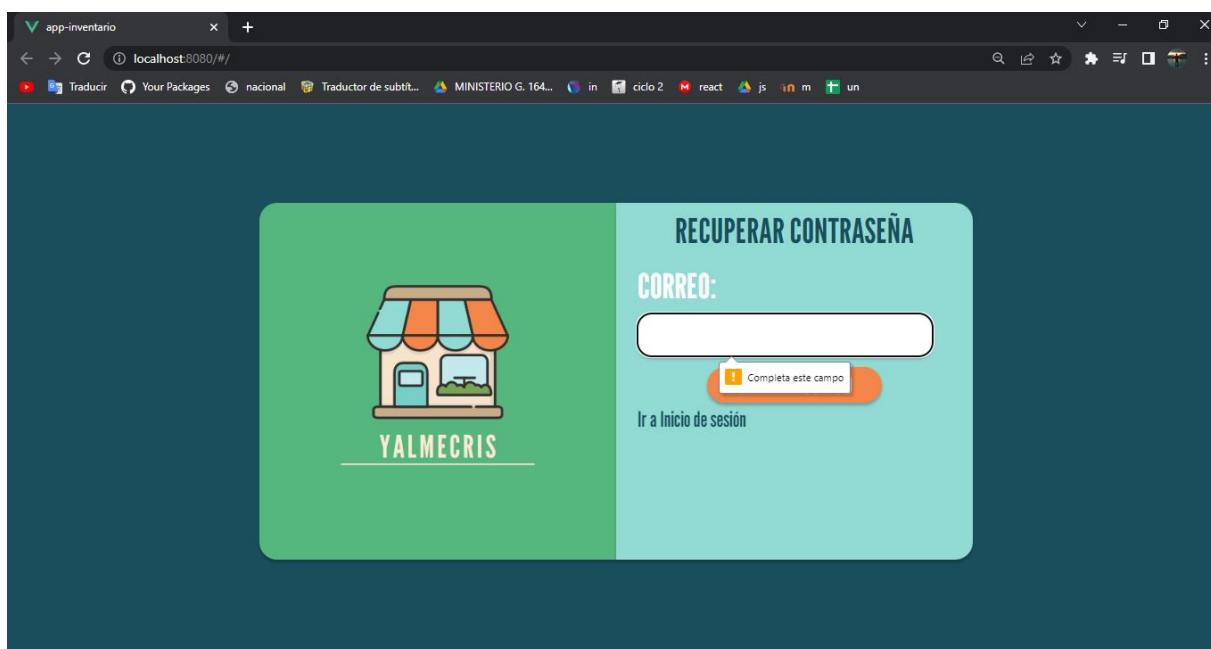


Ilustración 74 Campo incompleto en Recuperar contraseña

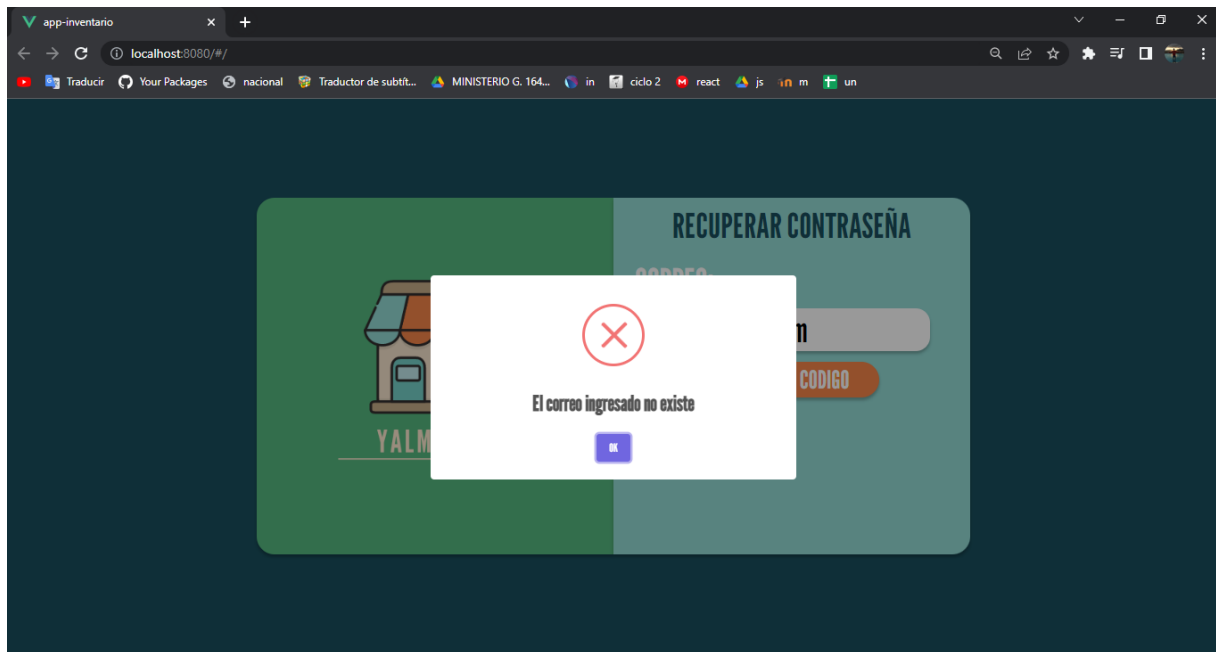


Ilustración 75 Correo no existe

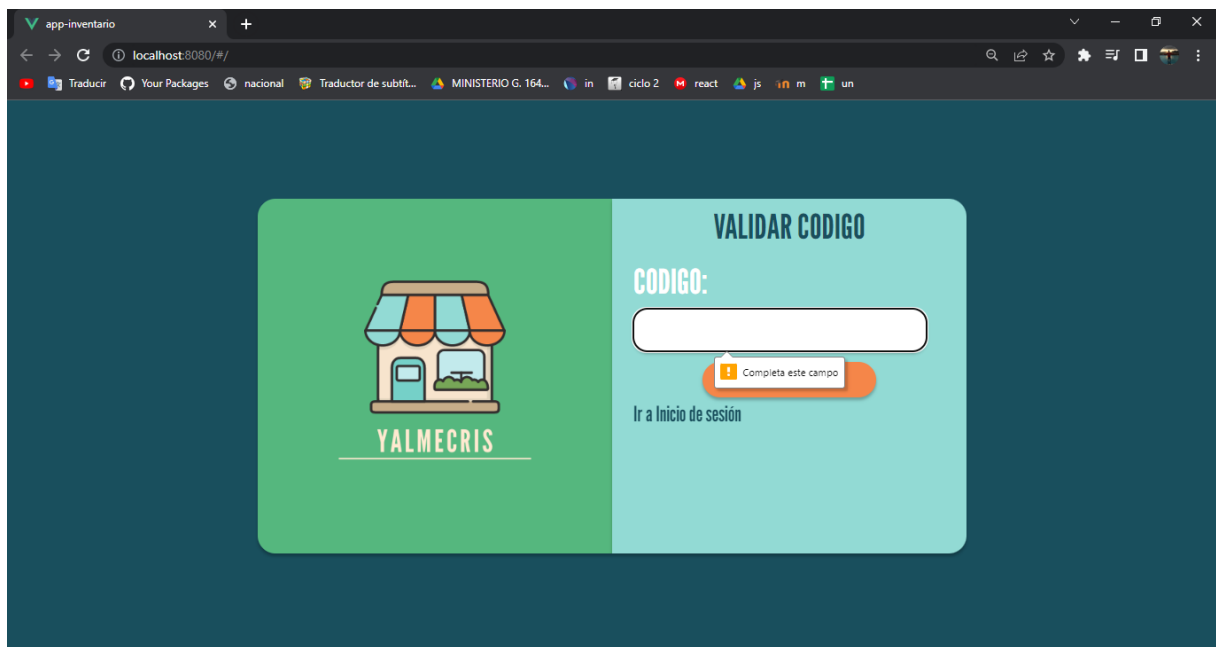


Ilustración 76 Campo incompleto en validar código

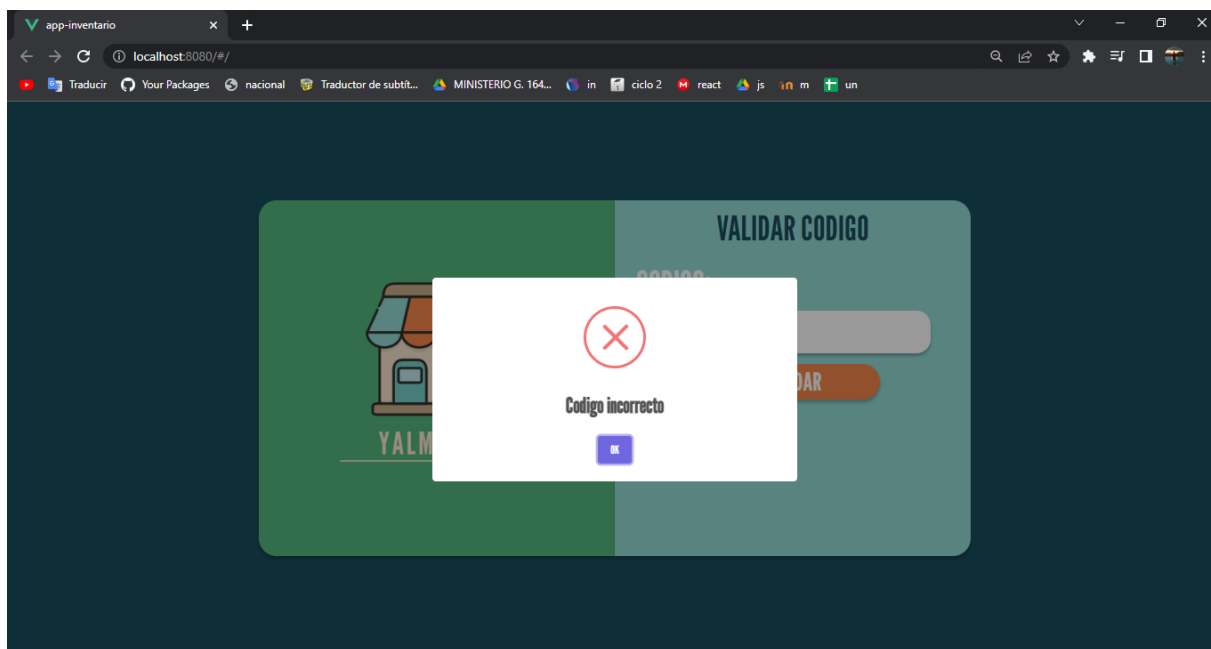


Ilustración 77 Código incorrecto

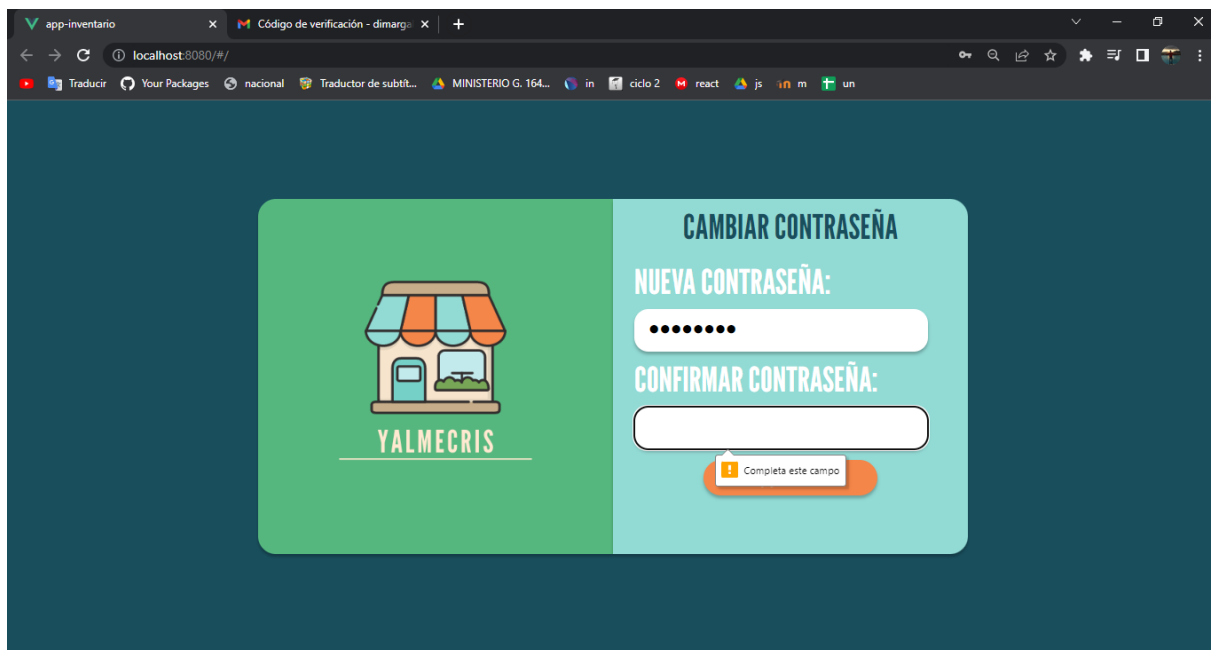


Ilustración 78 campos incompletos en cambiar contraseña

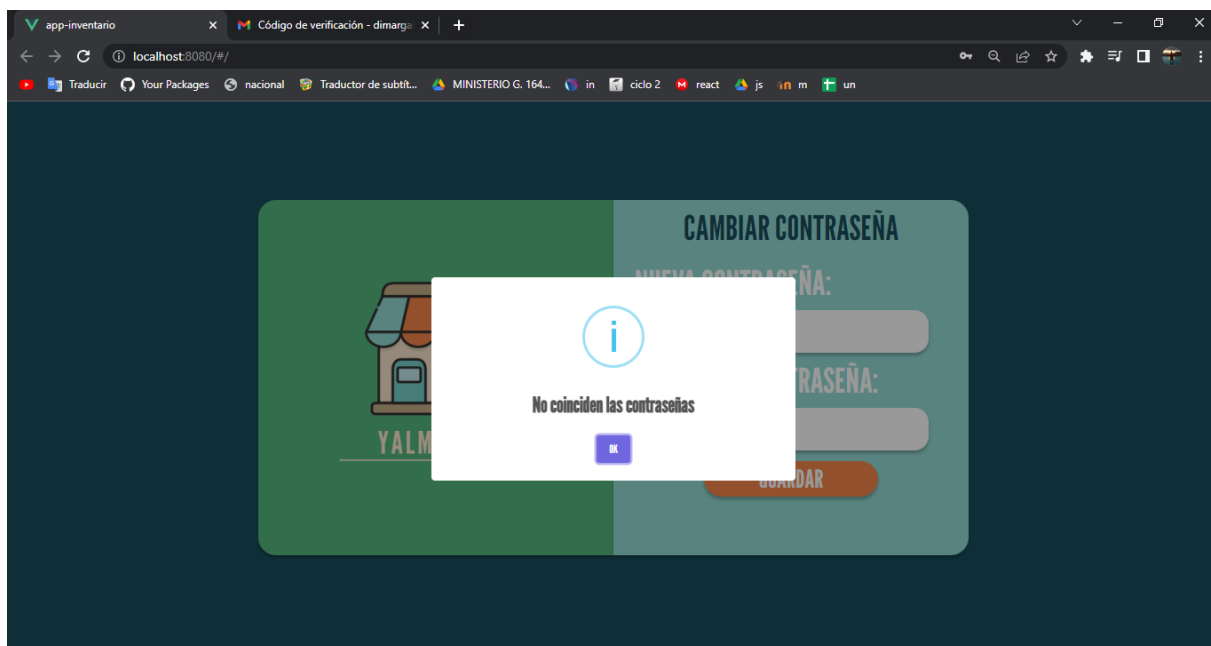


Ilustración 79 No coinciden las contraseñas

5.2.2.2.4 Registrar proveedores

A screenshot of a web browser displaying a provider registration form titled "REGISTRAR PROVEEDOR". The form is light blue and contains several input fields: "Nombre *" (with a red cap icon), "Tipo de Documento *" (a dropdown menu), "Numero de Documento *" (a text field), "Telefono *" (a text field), and "Correo *" (a text field). A red tooltip with a warning icon and the text "Completa este campo" (Complete this field) is visible next to the "Nombre *" field. At the bottom of the form is an orange button labeled "REGISTRAR".

Ilustración 80 campos incompletos formulario

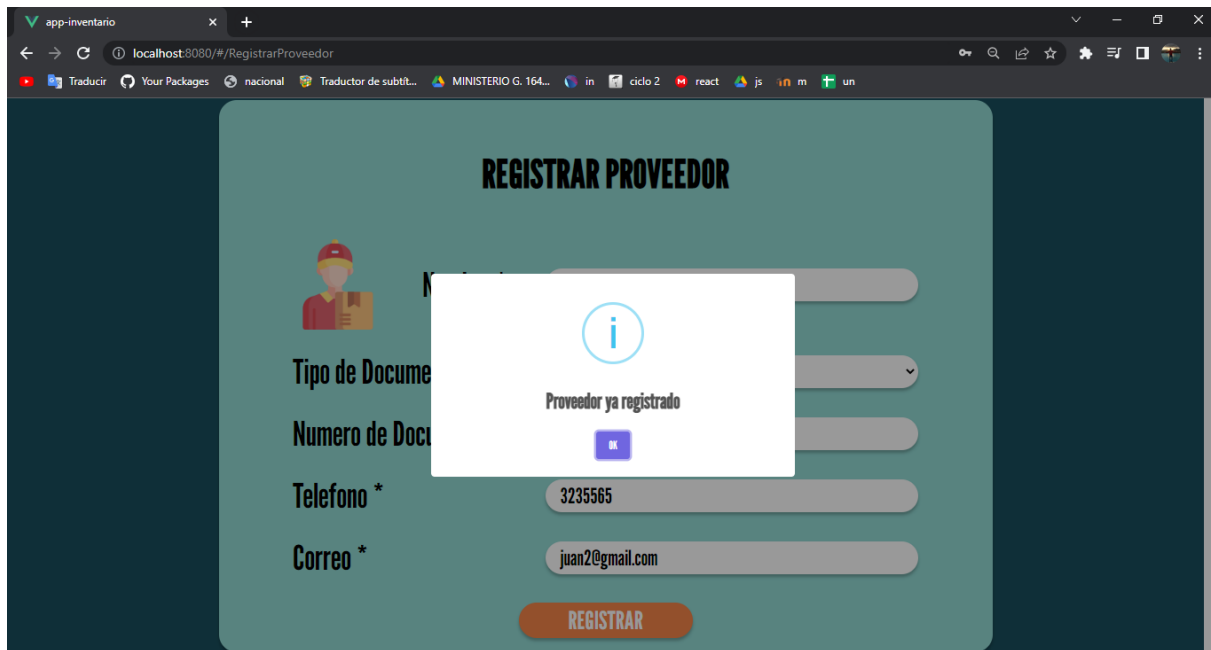


Ilustración 81 Proveedor registrado

5.2.2.2.5 Registrar Cliente

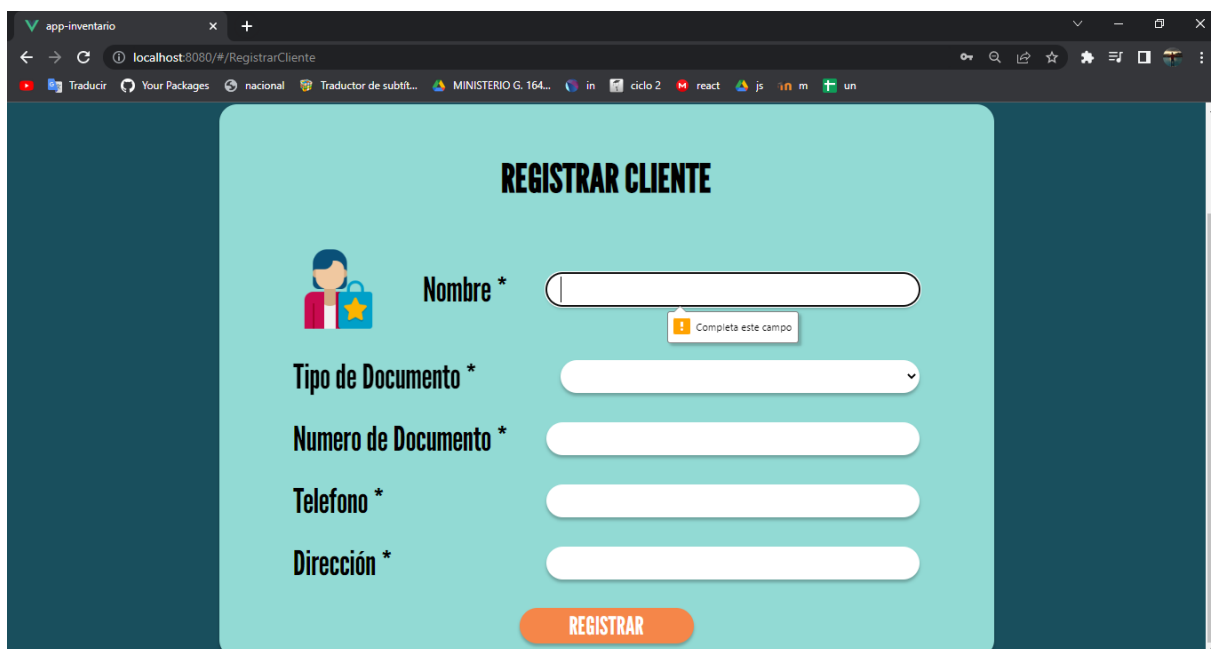


Ilustración 82 Campos incompletos

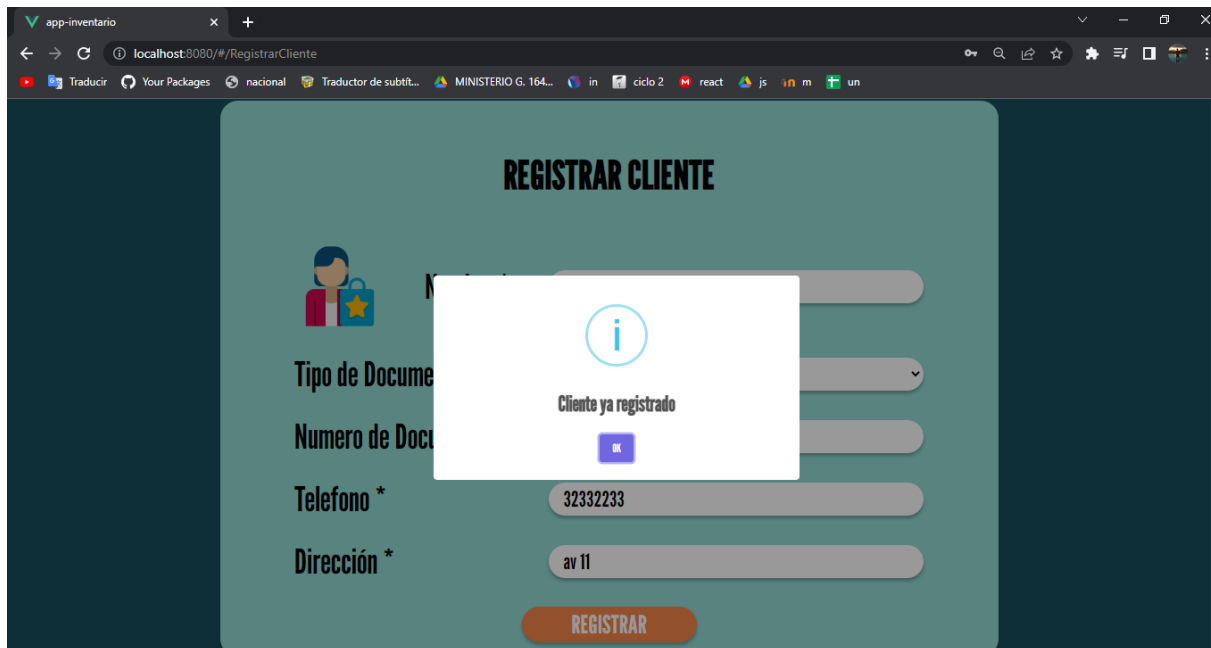


Ilustración 83 Cliente Registrado

5.2.2.2.6 Registrar Operador

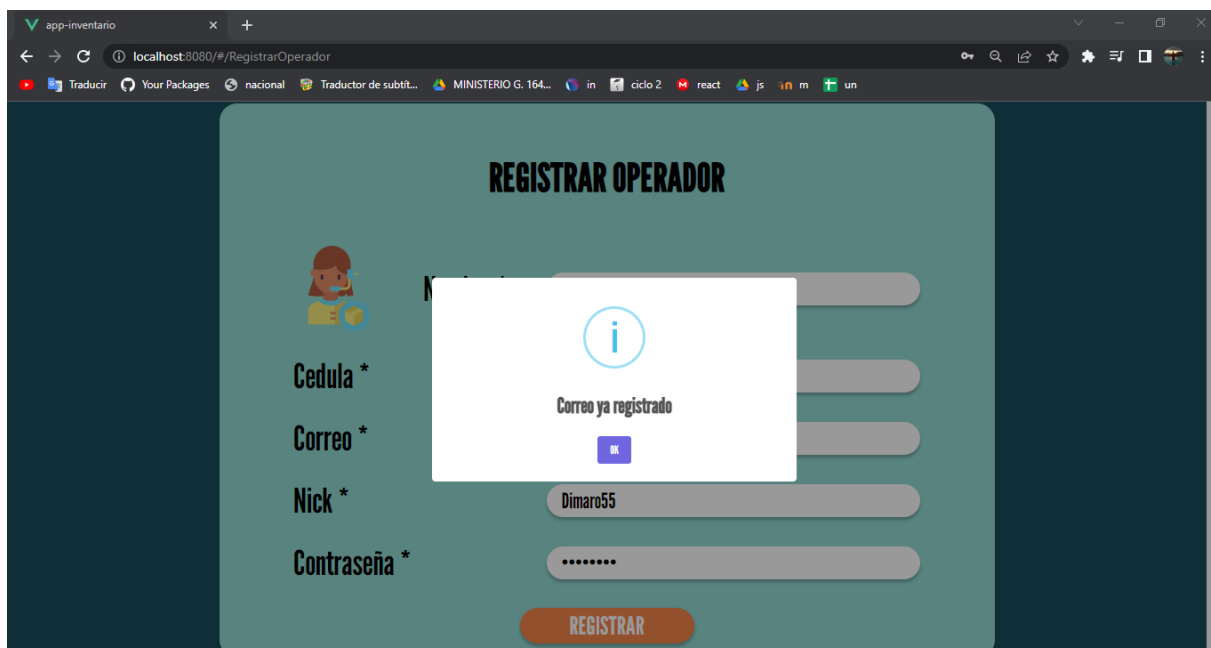


Ilustración 84 Correo usuario ya registrado

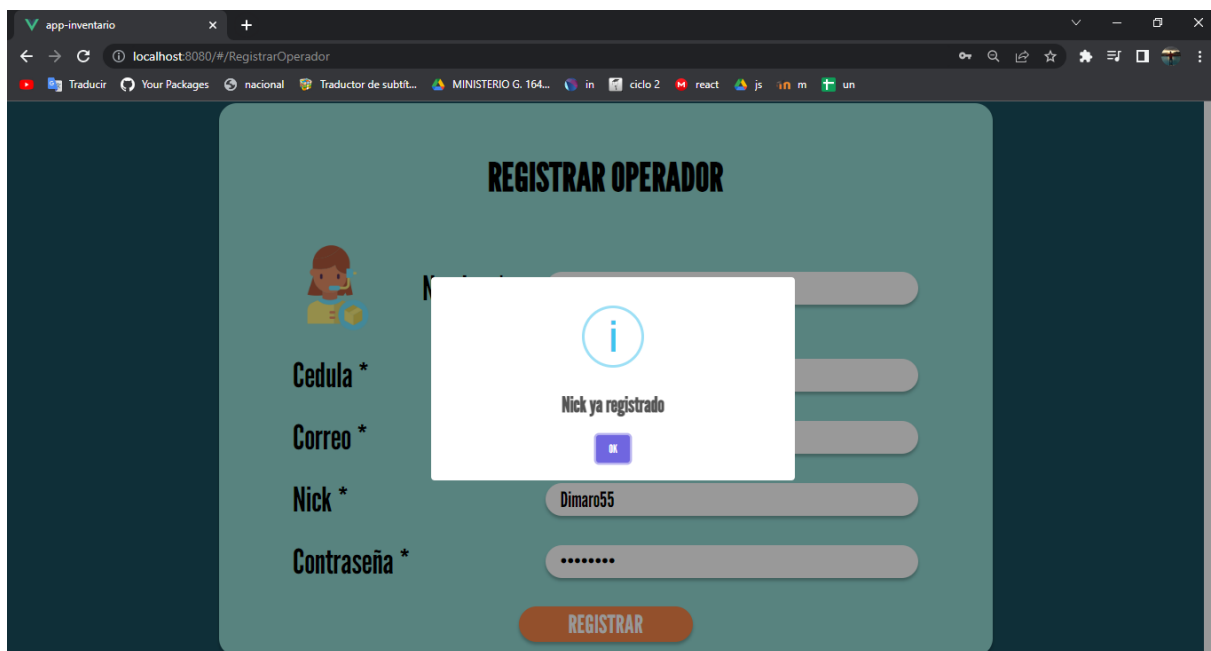


Ilustración 85 Nick ya registrado

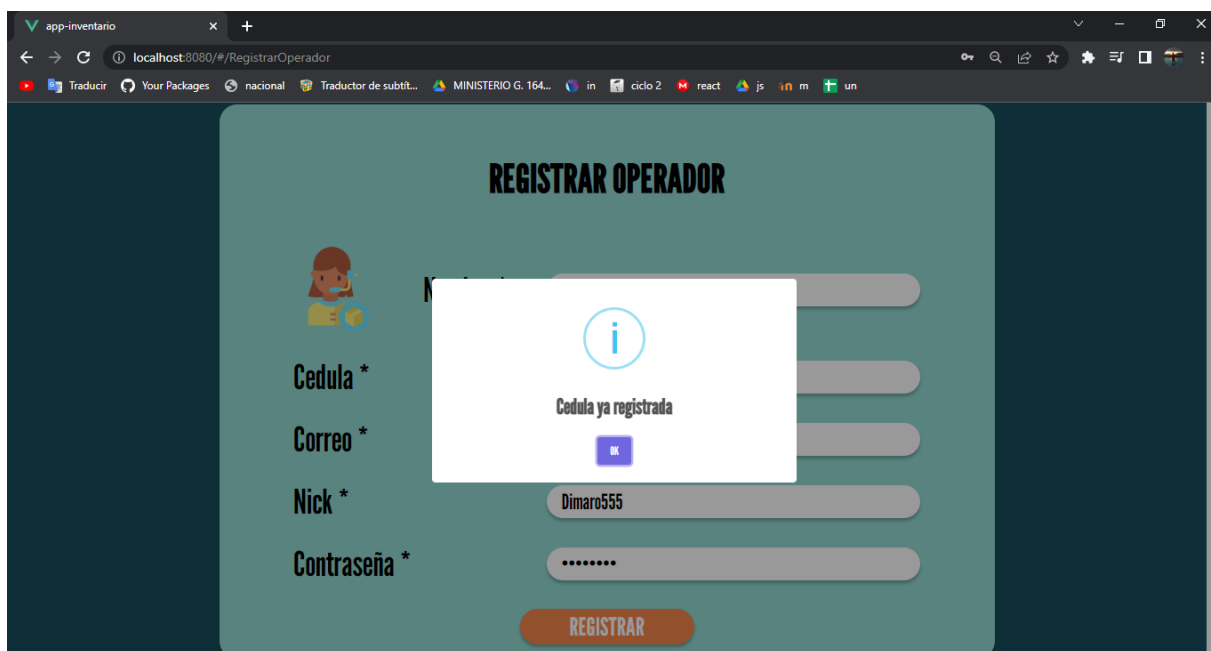


Ilustración 86 Cedula usuario ya registrada

5.2.2.2.6 Registrar Producto

REGISTRAR PRODUCTO

 **Codigo *** 23213

Categoria * ASEO

Nombre *

Marca *

Unidad de medida *

Cantidad de medida

Completa este campo

Ilustración 87 Formulario producto incompleto

REGISTRAR PRODUCTO

 **Codigo *** 7702032252114

Categoria * ASEO

Nombre * iabon

Marca *

Unidad de medida *

Cantidad de medida

Precio entrada * 2500

Precio venta * 3000

producto ya registrado

REGISTRAR

Ilustración 88 producto ya registrado

5.2.2.2.7 Registrar Factura Ingreso



Ilustración 89 productos sin agregar a Factura

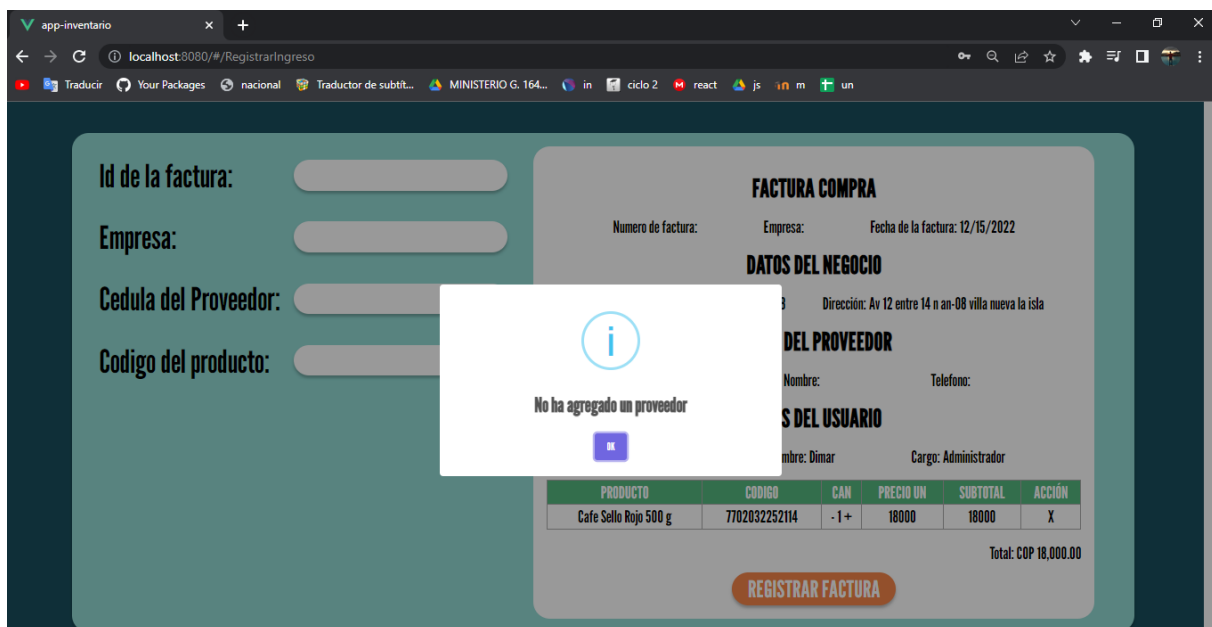


Ilustración 90 Proveedor sin asociar

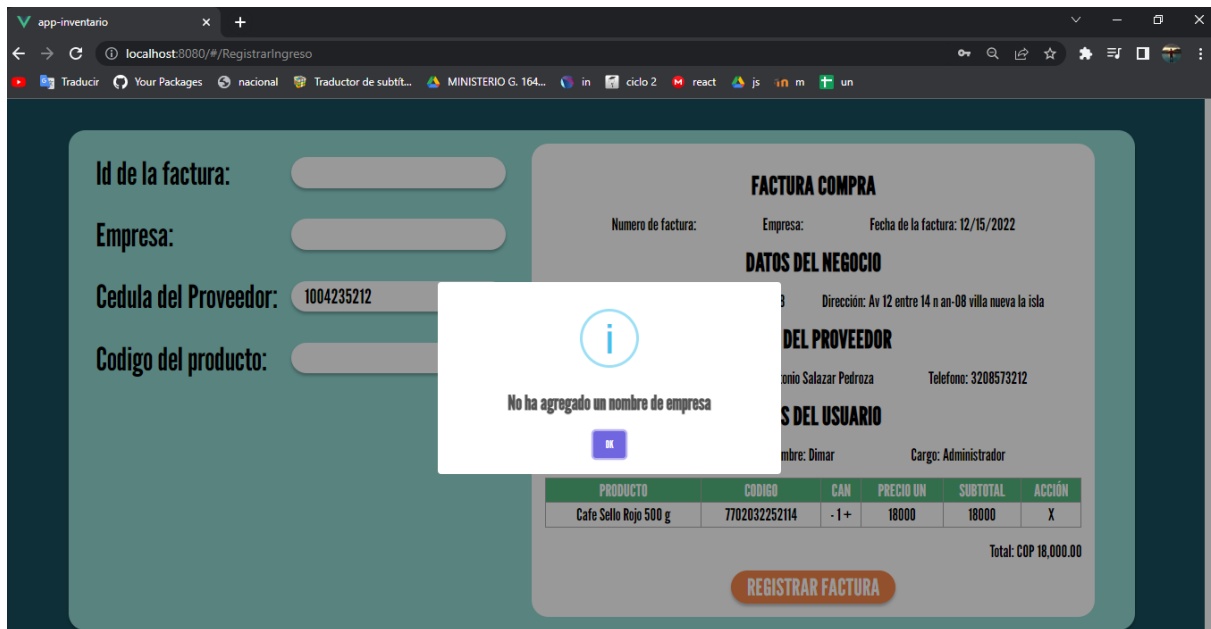


Ilustración 91 empresa sin agregar

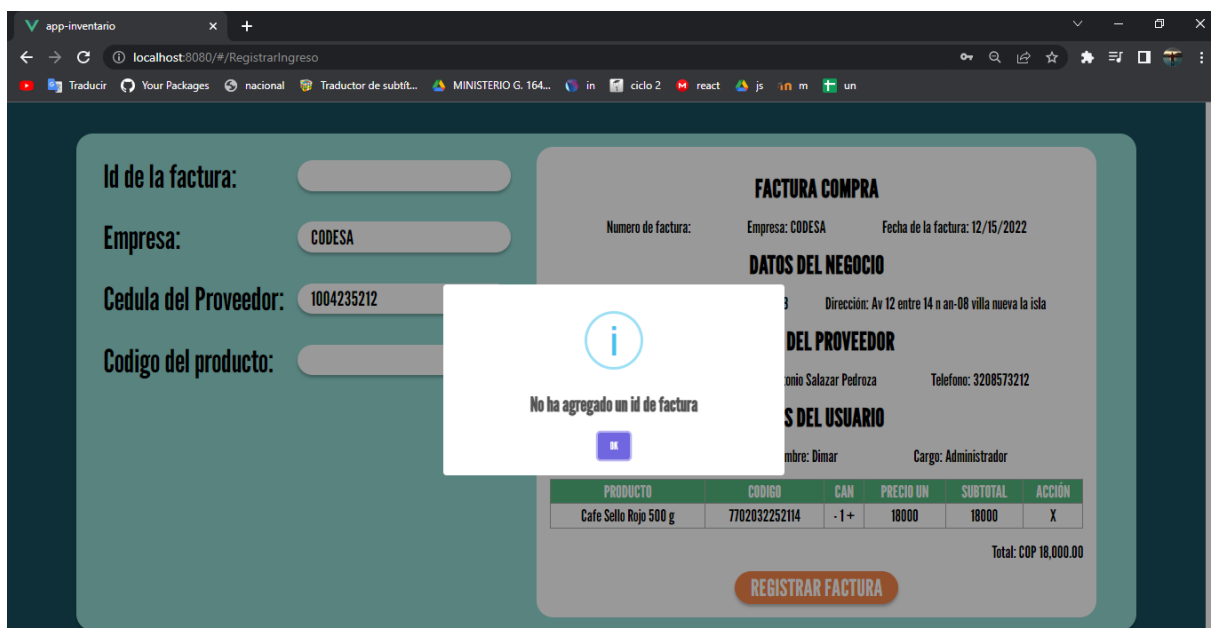


Ilustración 92 Id factura sin agregar

5.2.2.2.8 Registrar Venta

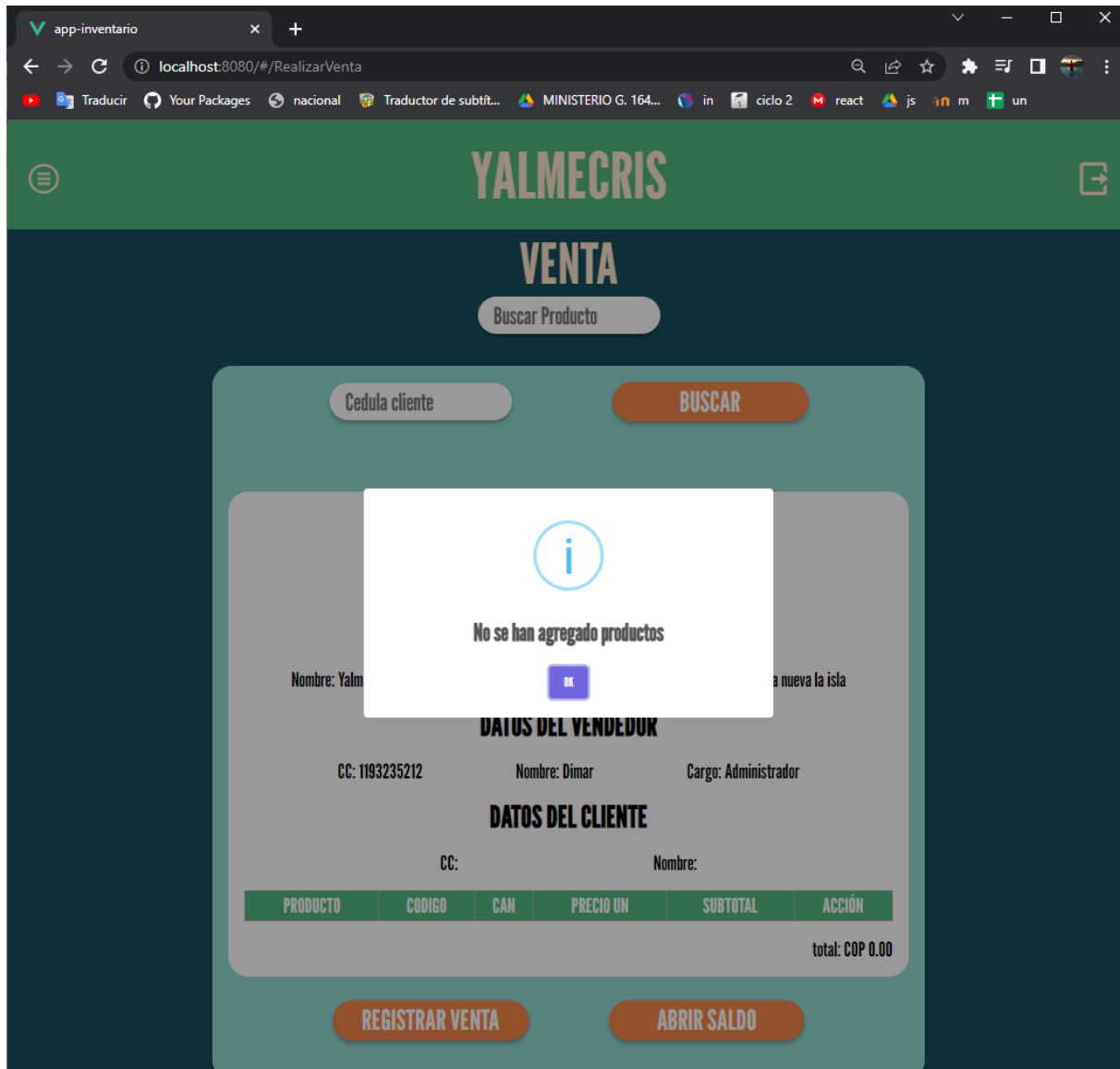


Ilustración 93 producto sin agregar a venta

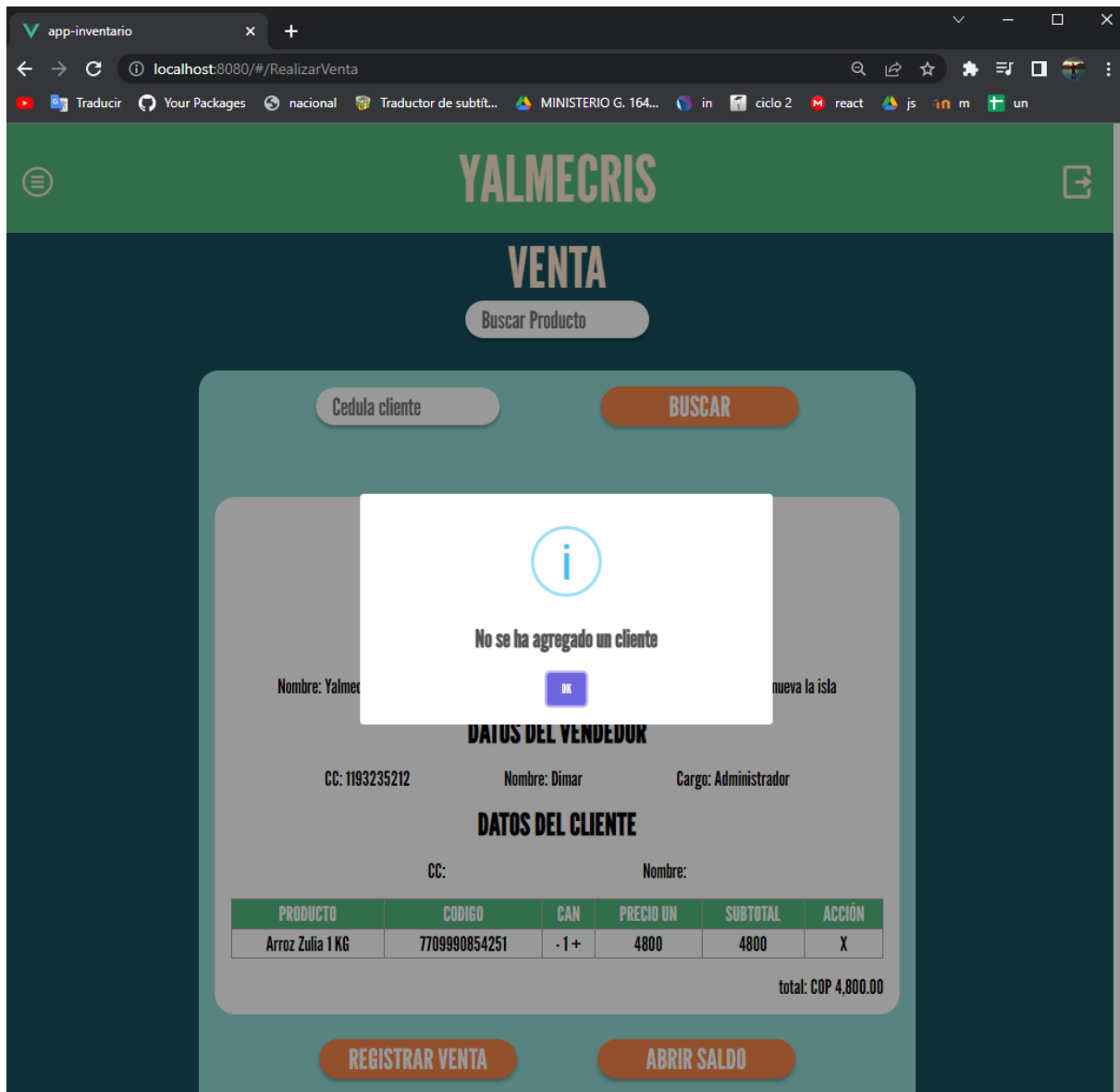


Ilustración 94 Cliente sin agregar a Venta

6. CONCLUSIONES

Con el presente trabajo se logró construir una aplicación web para el manejo y gestión del inventario en el negocio Yalmecris, este desarrollo partió de un diagnóstico inicial el cual permitió establecer los requisitos con lo que debería contar el sistema, posteriormente se realizaron los diferentes diagramas que son necesarios para llevar a cabo el diseño de la arquitectura del sistema, en el desarrollo de la aplicación web se optó por la separación de la capa lógica del negocio de la capa de presentación a lo cual llamamos backend y frontend respectivamente, el backend realizo con Node.js y la capa presentación con vue.js procurando mantener un diseño simple y una interfaz amigable con el usuario, usando como marco de trabajo la metodología ágil de desarrollo Scrum para gestión de las tareas y actividades necesarias para la finalización del proyecto. Este trabajo ha permitido tener una visión de las implicaciones que conlleva la realización de un desarrollo real permitiendo conseguir nuevas habilidades que serán necesarias para mantener el un ambiente laboral.

El trabajo resulto en una aplicación la cual permite al usuario el registro de clientes, proveedores y usuarios de tipo operador, el registro de los productos además del registro del ingreso de un lote de productos en el sistema, también cuenta con el registro de las ventas y posibilidad de abrir un saldo para los clientes, además de poder visualizar el inventario del negocio, el historial de ventas y el historial de los saldos. Finalmente, a la aplicación se le hicieron pruebas unitarias a los endpoints hechos en el backend de la aplicación.

7. REFERENCIAS BIBLIOGRÁFICAS

- Betancur Rosas, V. A. (2018). *APLICACIÓN WEB PARA LA GESTION DE INVENTARIO DE REACTIVOS DEL LABORATORIO DE QUIMICA DE LA UNIVERSIDAD DE PAMPLONA SEDE VILLA DEL ROSARIO*. Villa del Rosario: Universidad de Pamplona.
- Córdoba Lemus, J. (2021). *Diseño e implementación de un sistema de gestión integral aplicado al mantenimiento, seguimiento y control de los equipos de rayos x*. Medellín: Universidad de Antioquia. Obtenido de <https://hdl.handle.net/10495/20158>
- descubre comunicación. (26 de julio de 2019). *¿Que es el backend y frontend?* Obtenido de <https://descubrecomunicacion.com/que-es-backend-y-frontend/>
- edix. (19 de Agosto de 2021). *Framework*. Obtenido de <https://www.edix.com/es/instituto/framework/>
- Esteban Gabriel, M. (2015). *Metodologías de desarrollo de software*. Buenos Aires: Universidad Catolica de Argentina. Obtenido de <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- HostingPedia. (07 de Febrero de 2019). *PostgreSQL*. Obtenido de <https://hostingpedia.net/postgresql.html>
- Kinsta. (13 de Mayo de 2021). *Que es Node.js y porque usarlo* . Obtenido de <https://kinsta.com/es/base-de-conocimiento/que-es-node-js/>
- Lema Sagbaycela, F. J. (2018). *Desarrollo del sistema web para el control de inventarios, ventas, facturación y publicidad del Taller de Aluminio y Vidrio “López” aplicando la metodología Lean Software Development*. Riobamba: Escuela Superior Politécnica de Chimborazo. Obtenido de <http://dspace.esPOCH.edu.ec/handle/123456789/9116>
- mdn web docs . (5 de Septiembre de 2022). *MVC*. Obtenido de <https://developer.mozilla.org/es/docs/Glossary/MVC>
- mdn web docs. (7 de julio de 2021). *CSS*. Obtenido de <https://developer.mozilla.org/es/docs/Web/CSS>
- mdn web docs. (4 de Agosto de 2022). *HTML lenguaje de etiquetas de hipertexto*. Obtenido de <https://developer.mozilla.org/es/docs/Web/HTML>
- mdn web docs. (4 de Agosto de 2022). *JavaScript*. Obtenido de <https://developer.mozilla.org/es/docs/Web/JavaScript>
- profile. (24 de Junio de 2020). *¿Qué son los patrones de diseño de software?* . Obtenido de <https://profile.es/blog/patrones-de-diseno-de-software/>
- proyectos agiles.org. (2018). *¿Que es Scrum?*
- Reyes Getial, F. J. (2017). *ANALISIS Y DISEÑO DE UN SISTEMA DE INFORMACIÓN, BASADO EN WEB MOVIL PARA LA ASIGNACIÓN DE CITAS MÉDICAS DE LA IPS CENTRO CLINICO INTEGRAL TIBÚ LTDA*. Villa del Rosario: Universidad de Pamplona.
- Sánchez Villalva, D. E. (2018). *Diseño e implementación de una aplicación web para el control de inventario de la empresa “DGC Jeans*. Riobamba, Ecuador: Escuela Superior Politécnica de Chimborazo. Obtenido de <http://dspace.esPOCH.edu.ec/handle/123456789/15965>

Tamayo Flórez, P. A. (2019). *Desarrollo de sistema de gestión de activos del área de ingeniería clínica de la fundación colombiana de cancerología Clínica Vida*. Medellín: Universidad de Antioquia. Obtenido de <https://hdl.handle.net/10495/12189>

Vue.js. (s.f.). *Introducción Vue.js*. Obtenido de <https://es.vuejs.org/v2/guide/>

W- ictea. (2022). *¿Que es una aplicación Web?* Obtenido de <https://www.ictea.com/cs/index.php?rp=/knowledgebase/4205/iQue-es-una-aplicacion-web.html>

Xataka. (23 de Agosto de 2019). *API: qué es y para qué sirve*. Obtenido de <https://www.xataka.com/basics/api-que-sirve>

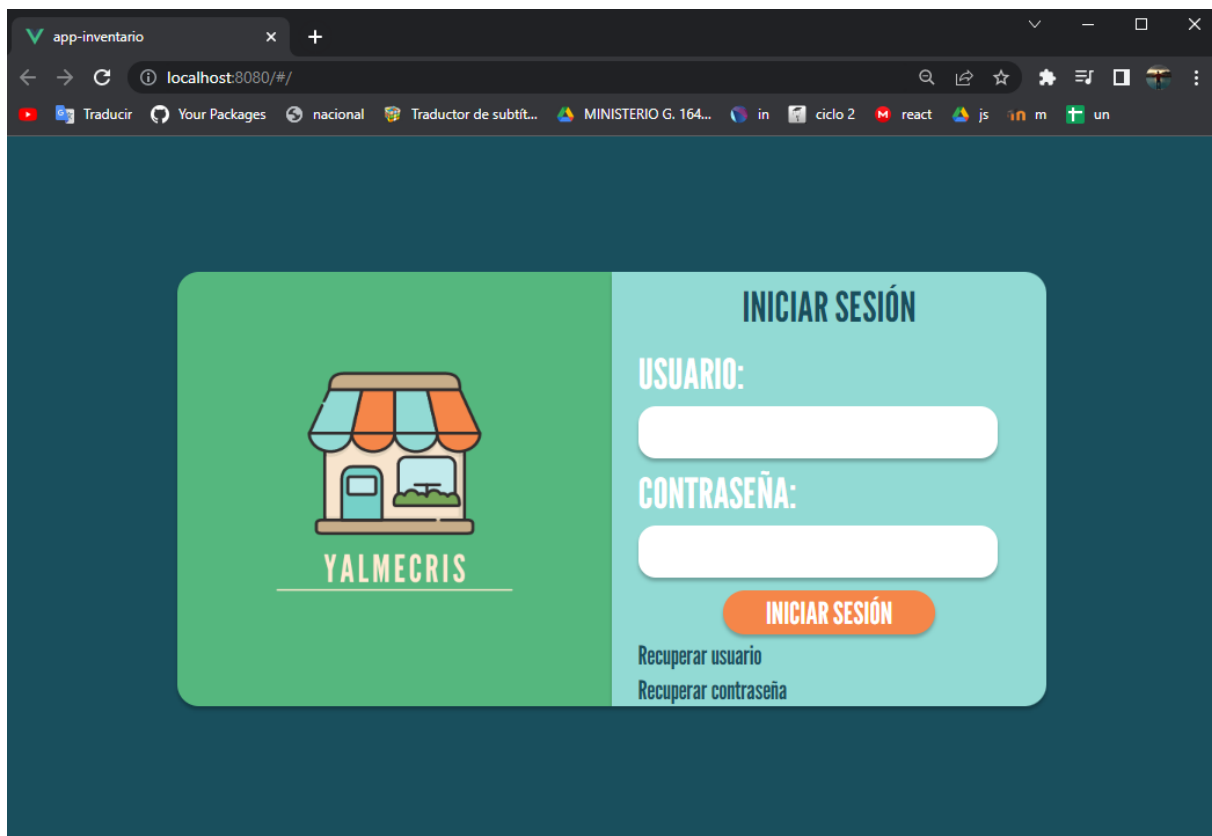
YeePLY. (2021). *¿Que son las pruebas unitarias y como llevarlas a cabo?* Obtenido de <https://www.yeeply.com/blog/que-son-pruebas-unitarias/>

ANEXOS

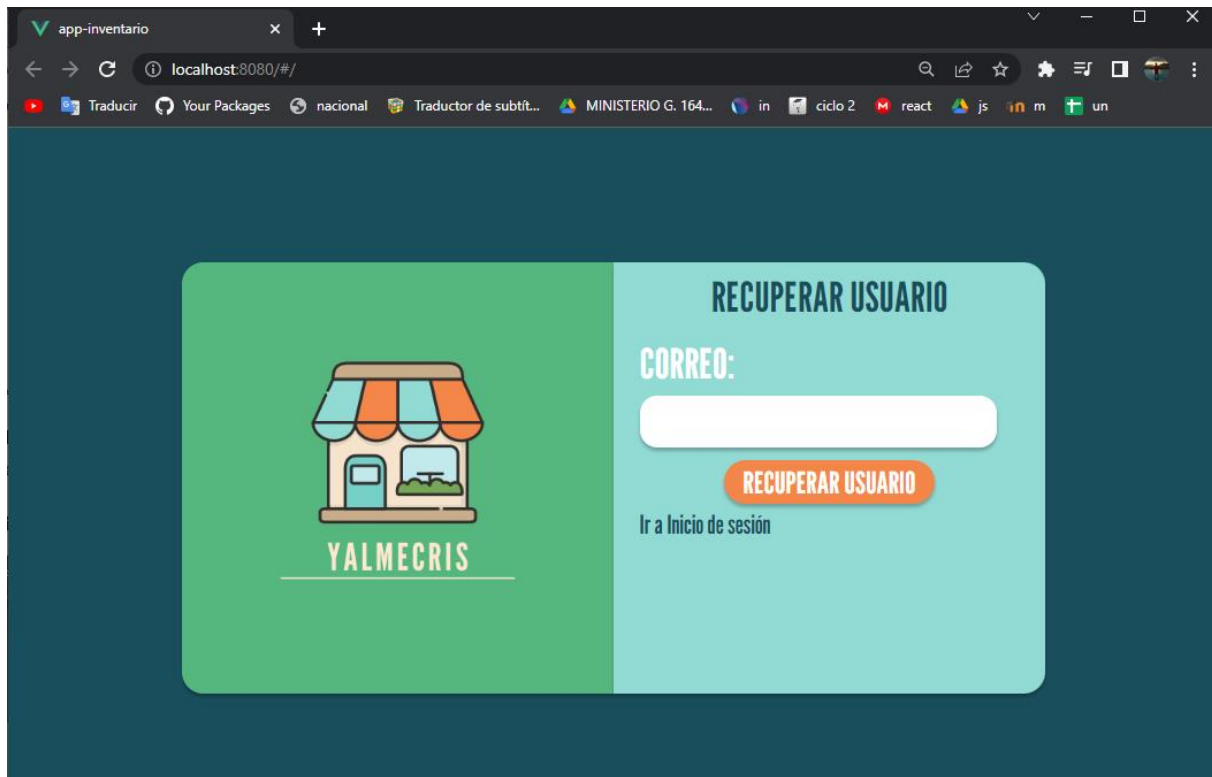
ANEXO A. MANUAL DEL USUARIO

Una vez iniciada la aplicación, la primera pantalla que se va a visualizar es la de inicio de sesión tal como se muestra a continuación

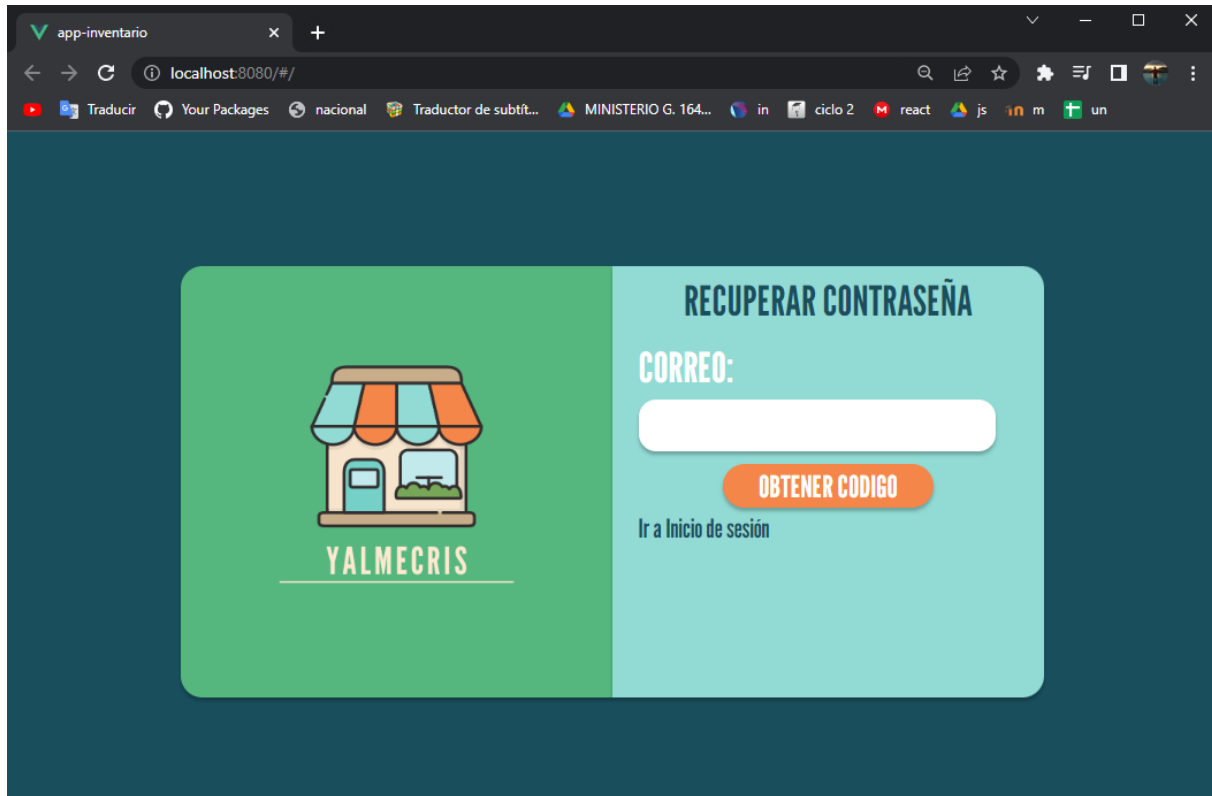
1. Inicio de sesión



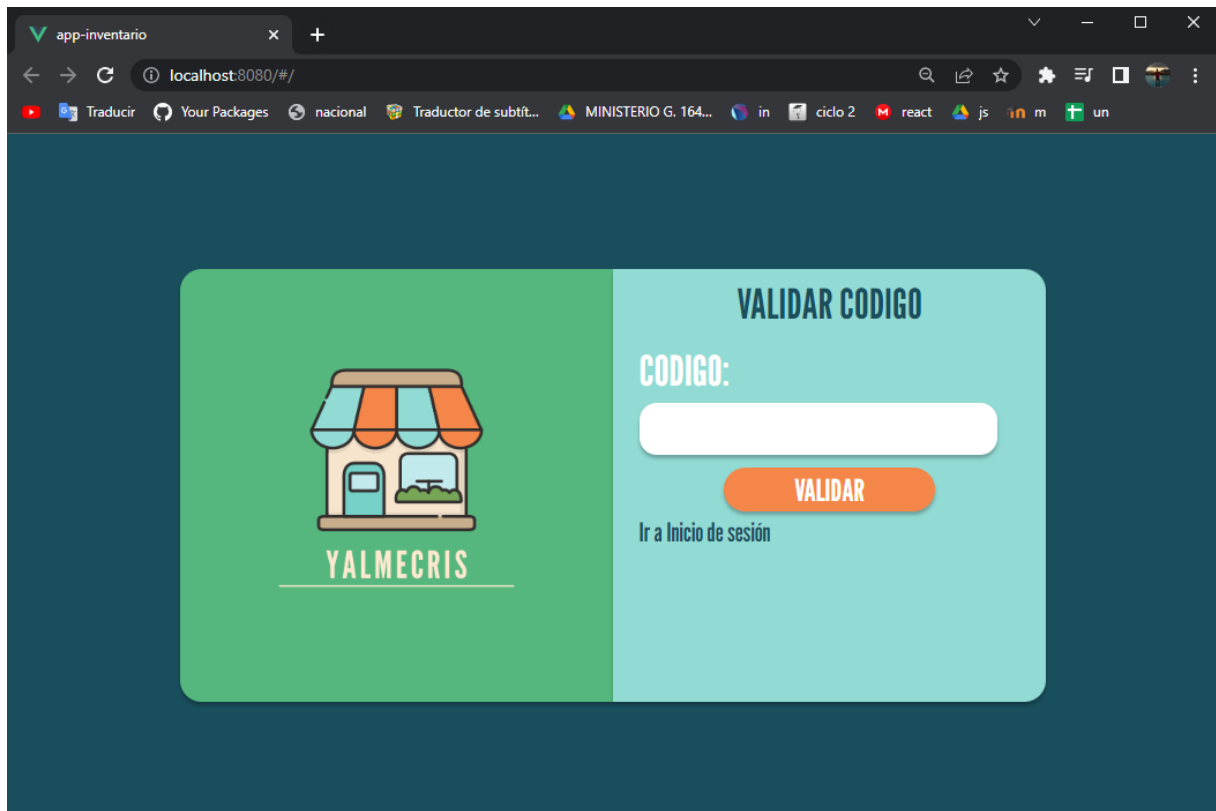
En esta interfaz el usuario deberá diligenciar sus credenciales las cuales son el Nick de usuario y la contraseña para poder tener acceso al menú de opciones según el rol del usuario que inicio sesión para ingresar con el rol operador la contraseña tendrá que ser suministrada por el operador ya que este es el único autorizado de registrar a un operador, Además en la parte inferior se puede visualizar las opciones de recuperar usuario o recuperar contraseña.



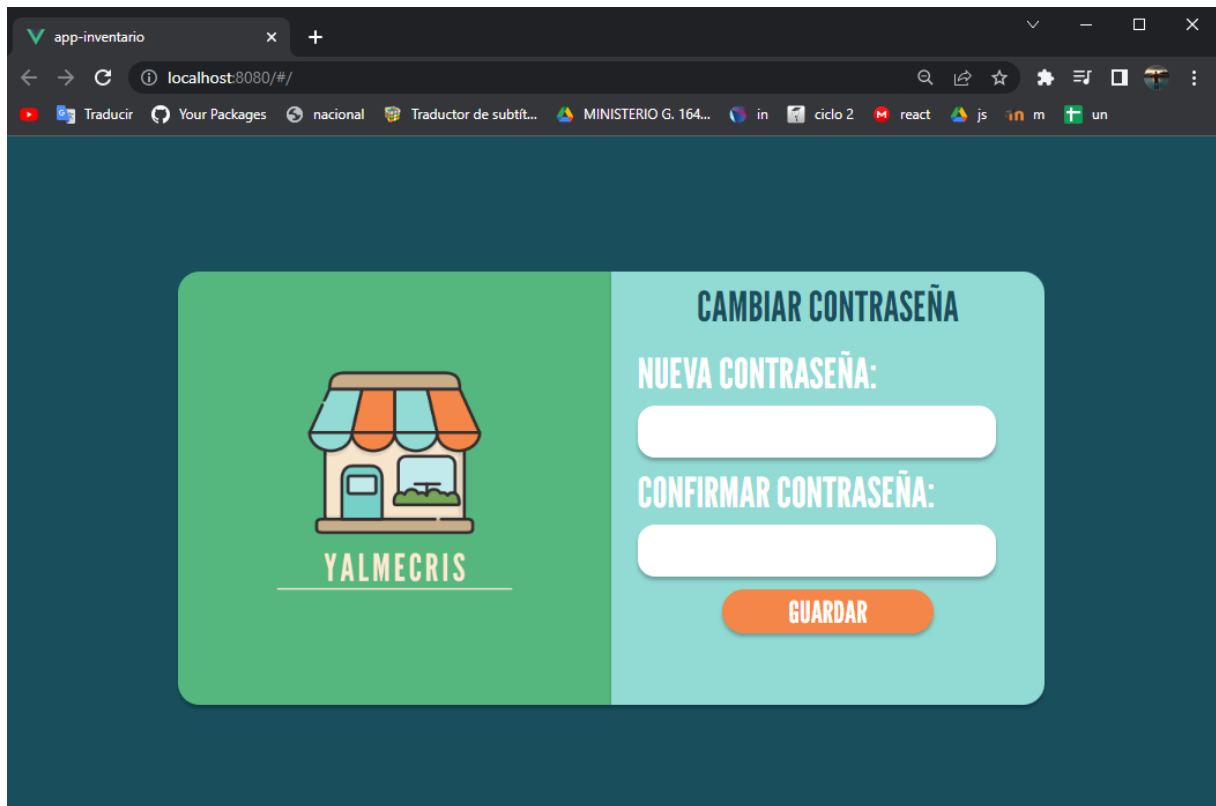
Para recuperar el usuario o Nick para iniciar sesión el usuario deberá escribir su correo el campo y se le enviará al correo el Nick que utiliza para iniciar sesión.



Si se ha olvidado la contraseña y usuario desea iniciar sesión para recuperar contraseña el usuario tendrá que escribir su correo con lo cual obtendrá un código que le permitirá cambiar la contraseña.



En esta vista el usuario deberá agregar el código que se le envió al correo para validar que el código es correcto.



Luego de validarse el código el usuario podrá cambiar la contraseña para poder iniciar sesión

2. Menú de opciones (Administrador)



Esta interfaz solo la podrá visualizar el usuario de tipo administrador después de haber iniciado sesión, en este menú se tiene acceso a las opciones del sistema las cuales son registrar proveedores, registrar cliente, registrar operador, registrar producto, registrar entrada, ver el inventario, ver los clientes, ver los proveedores, registrar una venta, ver el historial de ventas y ver el historial de saldos.

3. Menú de opciones (Operador)



Esta interfaz solo se podrá visualizar cuando un usuario de tipo Operador inicie sesión en el sistema, en este menú no se puede acceder a la opción de registrar operador.

4. Registrar proveedor

The screenshot shows a web interface for 'YALMECRIS'. At the top is a green header bar with a menu icon on the left and a share icon on the right. The main content area has a dark teal background. In the center is a light blue rounded rectangle titled 'REGISTRAR PROVEEDOR'. Inside this rectangle, on the left, is an icon of a person in a red cap and uniform holding a box. To the right of the icon are five input fields: 'Nombre' (text), 'Tipo de Documento' (dropdown), 'Numero de Documento:' (text), 'Telefono:' (text), and 'Correo:' (text). At the bottom right of the light blue rectangle is an orange button labeled 'REGISTRAR'.

Esta interfaz le permite al usuario registrar un proveedor en el sistema para realizar el registro se debe llenar los campos de nombre, tipo de documento, numero de documento, teléfono y correo.

5. Registrar cliente

The screenshot shows a web interface for 'YALMECRIS'. At the top is a green header bar with a menu icon on the left and a share icon on the right. The main content area has a dark teal background. In the center is a light blue rounded rectangle titled 'REGISTRAR CLIENTE'. Inside this rectangle, on the left, is an icon of a person in a blue cap and uniform holding a box with a star. To the right of the icon are five input fields: 'Nombre' (text), 'Tipo de Documento' (dropdown), 'Numero de Documento:' (text), 'Telefono:' (text), and 'Dirección:' (text). At the bottom right of the light blue rectangle is an orange button labeled 'REGISTRAR'.

Esta interfaz le permite al usuario registrar un cliente en el sistema para realizar el registro se debe llenar los campos de nombre, tipo de documento, numero de documento, teléfono y dirección.

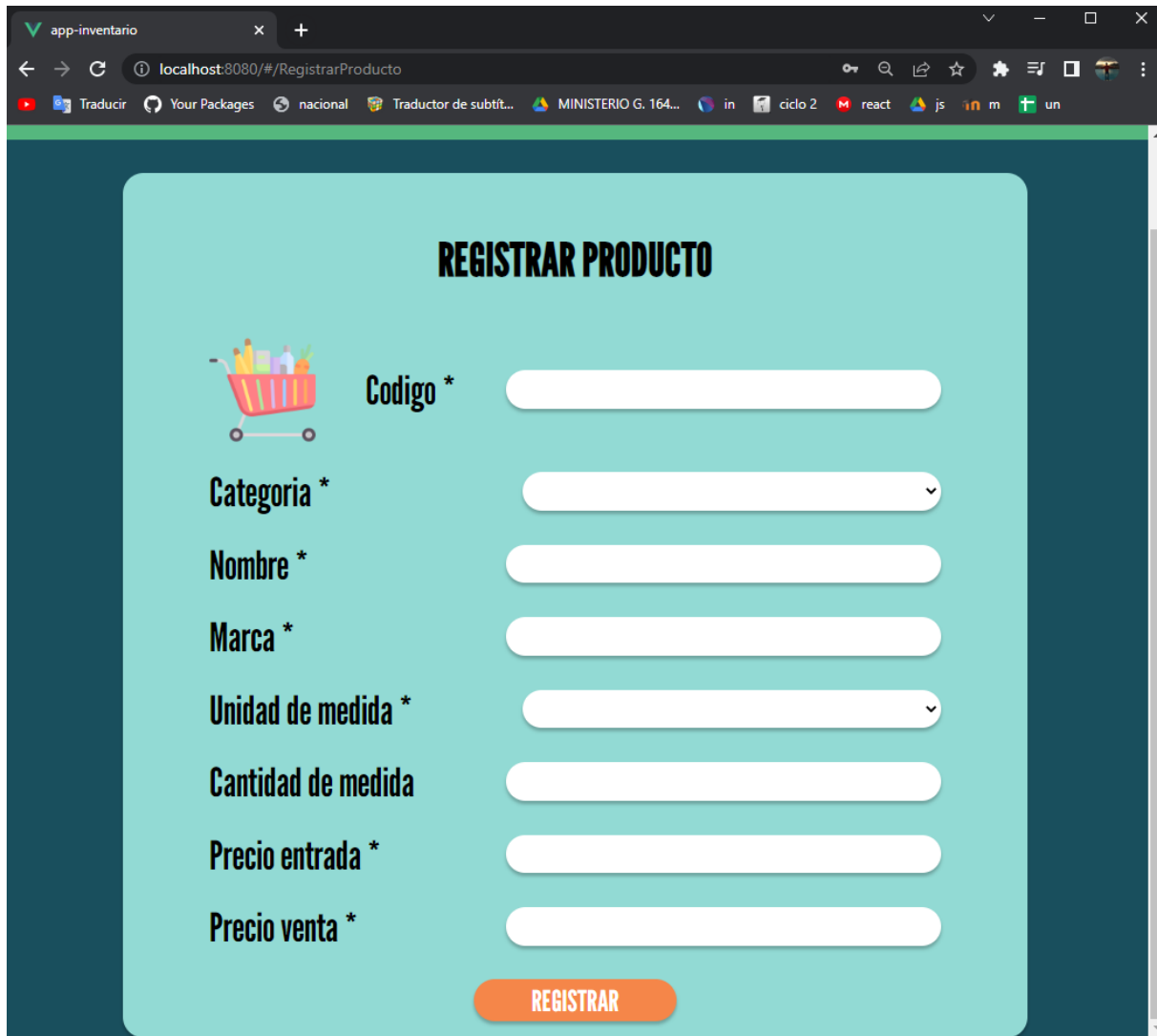
6. Registrar operador



The image shows a web interface for 'YALMECRIS' with a green header bar. The main content area is dark teal. Centered on this is a light blue rounded rectangle titled 'REGISTRAR OPERADOR'. Inside this rectangle, on the left, is a small cartoon icon of a person with brown hair and a yellow shirt. To the right of the icon are four labels: 'Nombre', 'Cedula:', 'Nick:', and 'Contraseña:'. Each label is followed by a white rounded rectangular input field. At the bottom of the light blue rectangle is an orange button with the word 'REGISTRAR' in white capital letters. The header bar also contains a white menu icon on the left and a white square icon on the right.

Esta interfaz le permite al usuario registrar un operador en el sistema para realizar el registro se debe llenar los campos de nombre, cedula, Nick, y contraseña.

7. Registrar producto



The screenshot displays a web application interface for registering a product. The browser's address bar shows the URL 'localhost:8080/#/RegistrarProducto'. The page has a dark teal background with a light teal rounded rectangle in the center containing the form. The form is titled 'REGISTRAR PRODUCTO' in bold black text. To the left of the 'Codigo *' field is a shopping cart icon. The form contains the following fields and labels:

- Codigo ***: A text input field.
- Categoría ***: A dropdown menu.
- Nombre ***: A text input field.
- Marca ***: A text input field.
- Unidad de medida ***: A dropdown menu.
- Cantidad de medida**: A text input field.
- Precio entrada ***: A text input field.
- Precio venta ***: A text input field.

At the bottom of the form is an orange button labeled 'REGISTRAR'.

Esta interfaz le permite al usuario registrar un producto en el sistema para realizar el registro se debe llenar los campos como el código del producto, la categoría a la que pertenece, el nombre, la marca, la unidad de medida del producto y la cantidad en base a esta medida en precio de entrada y el precio de venta al público.

8. Registrar Factura Compra

The screenshot shows a web browser window with the URL `localhost:8080/#/RegistrarIngreso`. The application header is green with the logo 'YALMECRIS'. The main title is 'REGISTRO FACTURA INGRESO'.

Form Fields:

- Id de la factura:
- Empresa:
- Cedula del Proveedor:
- Codigo del producto:

FACTURA COMPRA Summary:

- Numero de factura:
- Empresa:
- Fecha de la factura: 12/15/2022

DATOS DEL NEGOCIO:

- Nombre: Yalmecris
- Telefono: 5782288
- Dirección: Av 12 entre 14 n an-08 villa nueva la isla

DATOS DEL PROVEEDOR:

- CC:
- Nombre:
- Telefono:

DATOS DEL USUARIO:

- CC: 1193235212
- Nombre: Dimar
- Cargo: Administrador

PRODUCTO	CODIGO	CAN	PRECIO UN	SUBTOTAL	ACCIÓN
					Total: COP 0.00

REGISTRAR FACTURA

Esta interfaz le permite al usuario registrar el ingreso de uno de los productos en el sistema para realizar mediante el registro de una factura de ingreso o de compra se debe llenar los campos como el id de la factura, el nombre de la empresa, la cedula de un proveedor y el código de un producto.

9. Registrar venta

The screenshot shows a web application titled 'YALMECRIS' with a 'VENTA' section. The interface includes a search bar for products, a form for entering client information (Cedula cliente), and a 'BUSCAR' button. Below this is a 'FACTURA VENTA' section with fields for 'Numero de Venta' (22) and 'Fecha de Venta' (12/15/2022). The 'DATOS DEL NEGOCIO' section contains 'Nombre: Yalmecris', 'Telefono: 5782288', and 'Dirección: Av 12 entre 14 n an-08 villa nueva la isla'. The 'DATOS DEL VENDEDOR' section includes 'CC: 1193235212', 'Nombre: Dimar', and 'Cargo: Administrador'. The 'DATOS DEL CLIENTE' section has 'CC:' and 'Nombre:' fields. A table with columns 'PRODUCTO', 'CODIGO', 'CAN', 'PRECIO UN', 'SUBTOTAL', and 'ACCIÓN' is present, with a 'total: COP 0.00' displayed below it. At the bottom are 'REGISTRAR VENTA' and 'ABRIR SALDO' buttons.

YALMECRIS

VENTA

Buscar Producto

Cedula cliente **BUSCAR**

FACTURA VENTA

Numero de Venta: 22 Fecha de Venta: 12/15/2022

DATOS DEL NEGOCIO

Nombre: Yalmecris Telefono: 5782288 Dirección: Av 12 entre 14 n an-08 villa nueva la isla

DATOS DEL VENDEDOR

CC: 1193235212 Nombre: Dimar Cargo: Administrador

DATOS DEL CLIENTE

CC: Nombre:

PRODUCTO	CODIGO	CAN	PRECIO UN	SUBTOTAL	ACCIÓN
----------	--------	-----	-----------	----------	--------

total: COP 0.00

REGISTRAR VENTA **ABRIR SALDO**

Esta interfaz le permite al usuario realizar una venta, para realizar el registro de la venta se debe buscar el producto mediante su código y este va añadir a la tabla en donde se podrá modificar la cantidad que se desea vender, también se deberá llenar el campo de la cedula de cliente y para finalizar la venta se deberá presionar en registrar venta, si la venta se le va abrir como saldo a un cliente entonces se deberá seleccionar en abrir saldo.

10. Abrir saldo

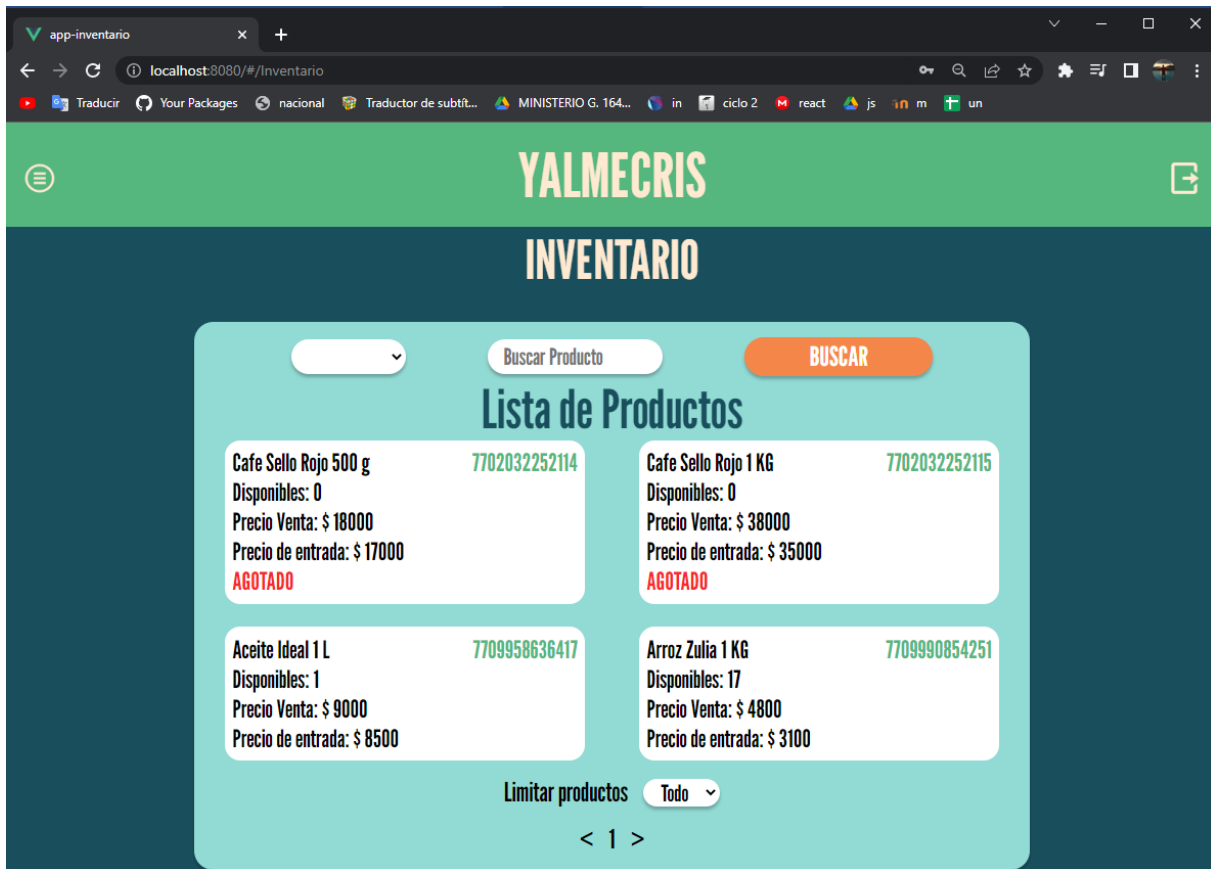
The screenshot shows a web browser window with the URL `localhost:8080/#/RegistrarSaldo?id_venta=22&total_venta=9600&cedula_cli=1192230212`. The application header is green with the logo 'YALMECRIS' and a menu icon. The main section is dark teal with the title 'Añadir Saldo' in white. A light blue rounded rectangle contains the form:

NOMBRE	CODIGO	TELEFONO
Juan Darío Ramírez Pallares	1192230212	3204853715

Below the table, there is an 'Abono:' label and a text input field containing '0'. To the left of the input field, the text 'Total: \$9600' is displayed twice. At the bottom of the form is an orange button labeled 'Registrar Saldo'.

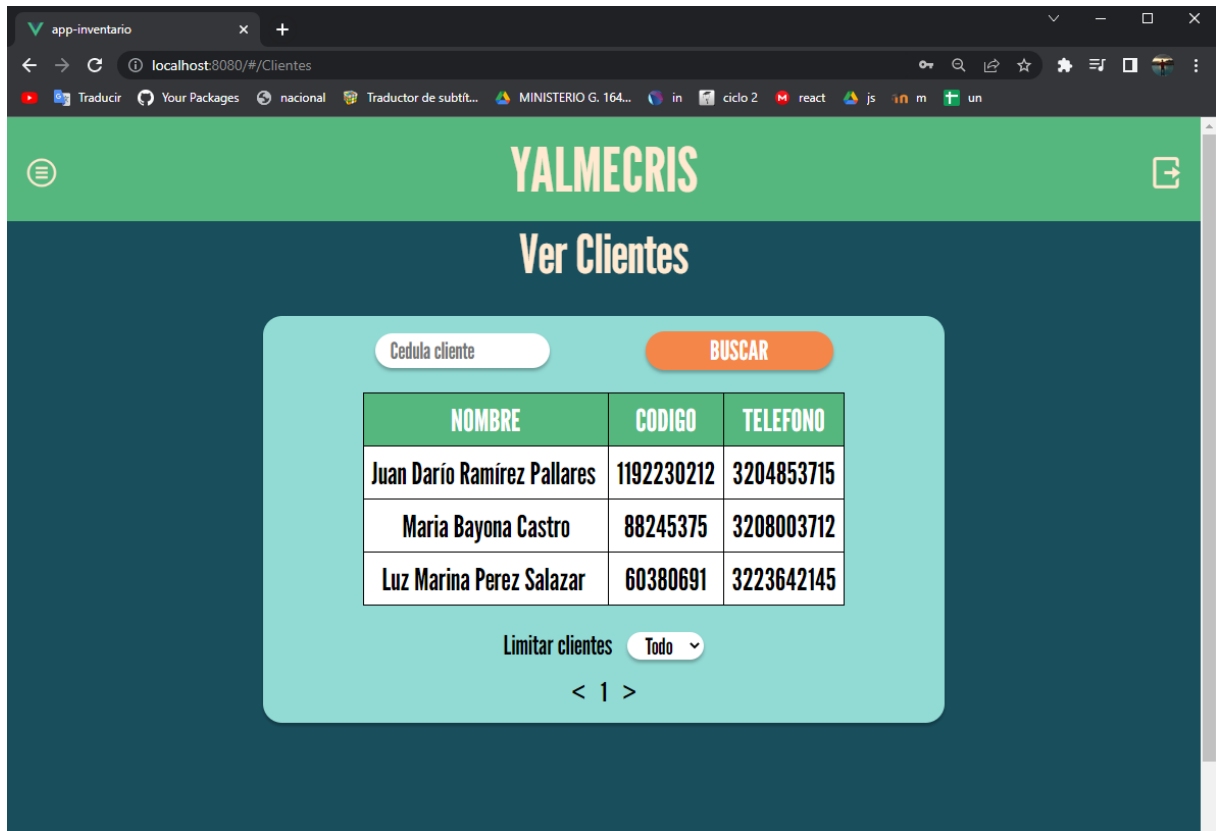
Esta es la interfaz que se va a visualizar en caso de que en el registro de la venta se haya decidido por abrir un saldo se mostrara la vista de la imagen anterior en donde vera los datos del cliente y si el usuario puede hacer un abono del total de la venta y finalmente se debe seleccionar registrar saldo y finaliza el proceso.

11. Ver Inventario



Esta es la interfaz de ver inventario en esta vista se va a ver todos los productos que están registrados en el sistema, se puede hacer uso de unos filtros de búsqueda ya sea por nombre, código y por cantidad menor o igual a cierta cantidad la cual debe ser escrita en el campo superior, además de cada producto se muestran los detalles de este y en rojo se muestra si ese producto se encuentra agotado.

12. Ver Clientes



En esta interfaz el usuario puede observar el listado de los diferentes clientes que están registrados en el sistema los detalles que salen de los clientes son su nombre, código y su teléfono.

13. Ver proveedores

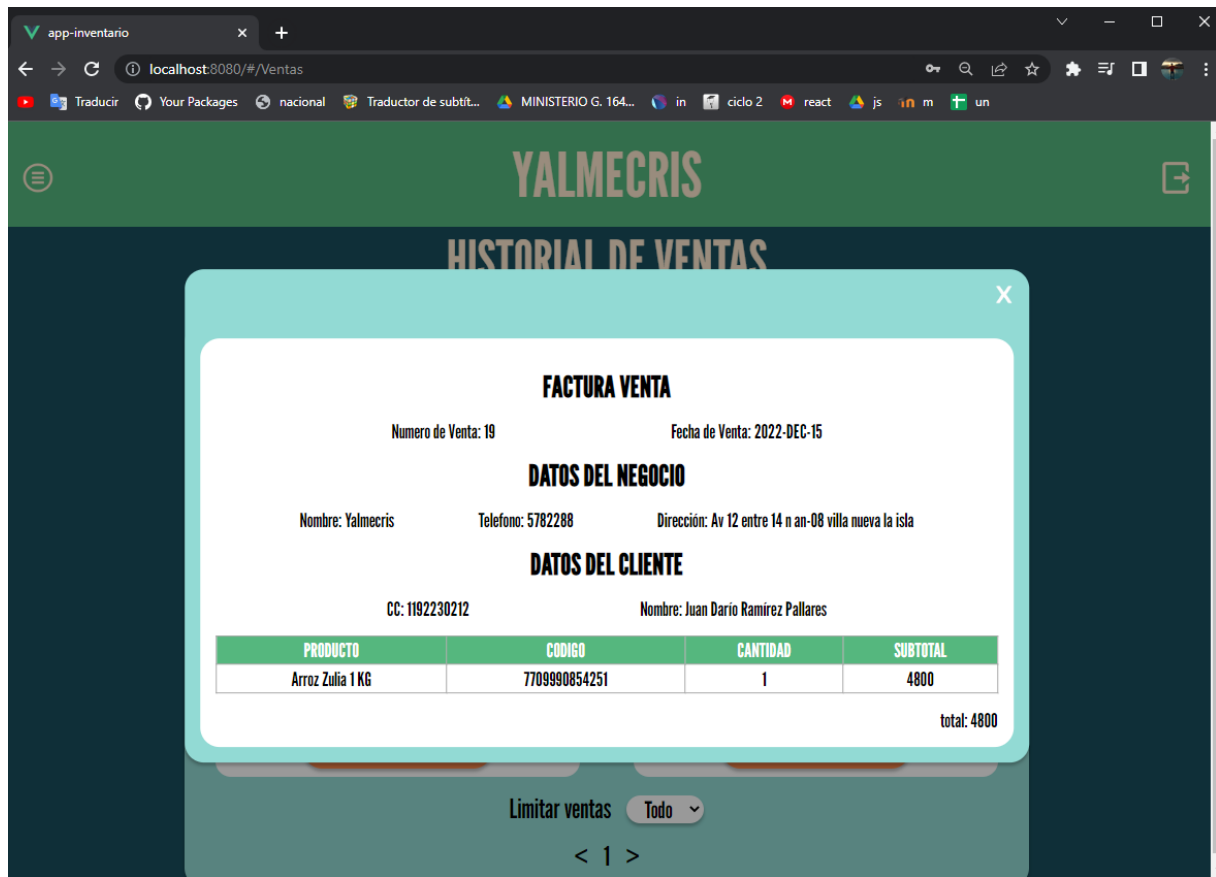


En esta interfaz el usuario puede observar el listado de los diferentes proveedores que están registrados en el sistema los detalles que salen de los proveedores son su nombre, código, teléfono y su correo.

14. Historial de ventas



En esta interfaz el usuario puede observar el listado de las diferentes ventas que están registradas en el sistema y poder ver los detalles de estas. Además de hacer un filtro de búsqueda.

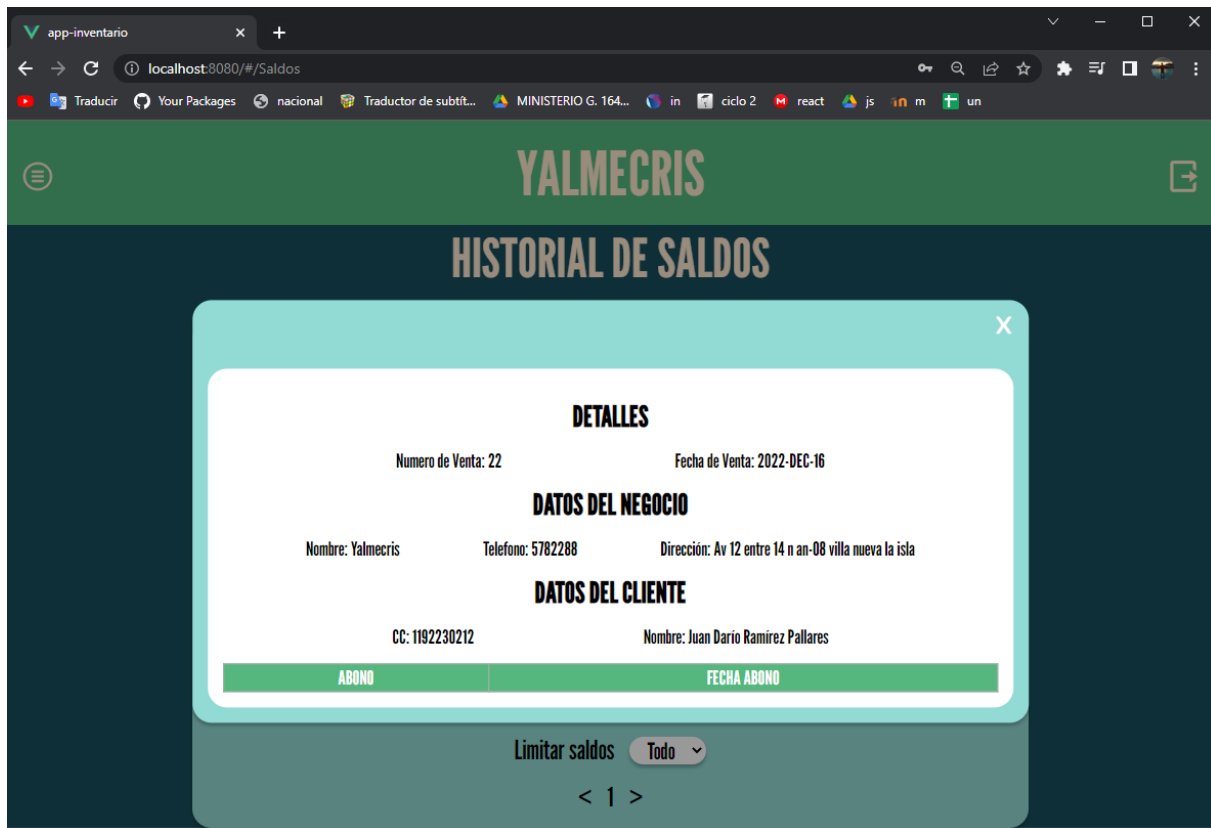


Si se selecciona en ver detalles se vera una vista con los datos relacionados a una venta en especifico con una tabla que muestra el producto que se compro el código del producto la cantidad el subtotal y el total de la venta.

15. Historial de saldos



En esta interfaz el usuario puede observar el listado de los saldos que están registrados en el sistema y poder hacer un filtro de búsqueda según el estado en el que se encuentre el saldo.



Al presionar sobre ver detalles en la vista de ver historial de saldos se puede visualizar algunos de los detalles sobre el saldo y una tabla con la cantidad de abono que se ha hecho y su fecha en que se realizó.