APLICACIÓN WEB PARA EL SEGUIMIENTO DE DEUDAS DEL CLIENTE Y GESTION DE INVENTARIO

FREDY STIVEN ANGARITA VELASCO

ERICK ENRIQUE CHAPARRO MARTINEZ

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE TECNOLOGÍA

PROYECTO CURRICULAR TECNOLOGÍA EN SISTEMATIZACIÓN DE DATOS

BOGOTÁ D.C, COLOMBIA

2025

APLICACIÓN WEB PARA EL SEGUIMIENTO DE DEUDAS DEL CLIENTE Y GESTION DE INVENTARIO

FREDY STIVEN ANGARITA VELASCO

20211578053

ERICK ENRIQUE CHAPARRO MARTINEZ

20211578057

TRABAJO DE GRADO EN MODALIDAD DE MONOGRAFÍA PARA OPTAR POR EL TITULO DE TECNÓLOGO EN SISTEMATIZACION DE DATOS

TUTOR:

JORGE EDUARDO HERNANDEZ RODRIGUEZ

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE TECNOLOGÍA

PROYECTO CURRICULAR TECNOLOGÍA EN SISTEMATIZACIÓN DE DATOS

BOGOTÁ D.C, COLOMBIA

2025

Tabla de Contenido

1. Introducción 16
2. Planeación 18
2.1. Titulo 18
2.2. Planteamiento 18
2.3. Formulación 19
3. Objetivos 20
3.1. Objetivo General 20
3.2. Objetivos Específicos 20
4. Alcance – Limitaciones 20
4.1. Alcances 20
4.2. Limitaciones 22
4.2.1. Roles del Sistema 22
5. Justificación 23
Marco Teórico 24
5.1. Marco Referencial 24
5.1.1. Morosidad 24
5.1.2. Gestión de Cobranza 25
6. Antecedentes 26
6.1. Odoo 26
6.2. Shopify 26
6.3. Zoho 26
6.4. Moonflow 26
6.5. Intiza 26
6.6. Duemint 27
7. Marco Conceptual 28
7.1. Lenguaje de programación 28
7.2. TypeScript 28

7.3. Java....28
7.4. Framework....28
7.5. NUXT....28
7.6. Spring Framework....28
7.7. Microservicios....29
7.8. Base de datos....29
7.9. SQL....29
7.10. PostgreSQL....29
7.11. Arquitectura Hexagonal....29
7.12. API Rest....29
7.13. JSON....30
7.14. Json Web Token (JWT)....30
8. Impactos....31
8.1. Impacto social....31
8.2. Impacto económico....31
8.3. Impacto tecnológico....31
8.4. Impacto organizacional....32
9. Metodología....32
9.1. Metodología de Desarrollo....32
9.2. Implementación.....33
10. Presupuesto y fuentes de financiación....35
10.1. Tabla de costos de recursos humanos....35
10.2. Factibilidad de Recursos Técnicos....35
10.3. Factibilidad de Otros Recursos....36
10.4. Factibilidad Económica Costo Total....36
11. Decisiones de diseño....37
11.1. Tecnologías....37

11.2. Modelo de datos 39
11.2.1. Modelo entidad relación 39
11.2.2. Modelo Relacional 40
11.3. Diagrama de Clases Backend 40
11.3.1. Apigateway 41
11.3.2. Microservicio de Stock 41
11.3.3. Microservicio de Payment 42
11.3.4. Microservicio de Transaction 43
11.3.5. Microservicio de User 44
11.4. Diagrama de Clases Frontend 45
11.5. Diagrama General de Casos de Uso 47
11.6. Diagramas de Secuencia 48
11.6.1. Secuencia de inicio de sesión 48
11.6.2. Secuencia de registro de usuario 48
11.6.3. Secuencias modulo inventario 49
11.6.4. Secuencias modulo transacción 59
11.6.5. Secuencias de deudores 65
11.7. Diagrama de Arquitectura de Software 67
11.8. Diagrama de Componentes 68
12. Desarrollo 69
12.1. Backlog 69
12.2. Análisis 71
12.2.1. Historias de usuario backend 71
12.2.2. Historias de usuario frontend 83
12.3. Implementación. 93
12.3.1. Módulo de autenticación. 93
12.3.2. Módulo de inventario 95
12.3.3. Módulo de transacciones. 97
12.3.4. Módulo de deudores. 99
13. Pruebas 100
13.1. Autenticación 100
13.2. Inventario 103

13.3. Transacción. 107
13.4. Deudores. 110
14. Resultados. 111
15. Conclusiones. 111
15.1. Aspecto organizacional 111
15.2. Aspecto tecnológico 112
15.3. Aspecto funcional 112
16. Recomendaciones 113
17. Bibliografía 114

Índice de Tablas

Tabla 1. Factibilidad de Recursos Humanos _____ 35
Tabla 2. Factibilidad de Recursos Técnicos_____ 36
Tabla 3. Factibilidad de Otros Recursos _____ 36
Tabla 4. Factibilidad Económica Costo Total _____ 36
Tabla 5. Comparación tecnologías Frontend. _____ 37
Tabla 6. Comparación tecnologías Backend _____ 39
Tabla 7. Backlog _____ 71
Tabla 8. Historia de Usuario 01 _____ 71
Tabla 9. Historia de Usuario 02 _____ 72
Tabla 10. Historia de usuario 3_____ 72
Tabla 11. Historia de usuario 4_____ 72
Tabla 12. Historia de usuario 5_____ 73
Tabla 13. Historia de usuario 6_____ 73
Tabla 14. Historia de usuario 7_____ 74
Tabla 15. Historia de usuario 8_____ 74
Tabla 16. Historia de usuario 9_____ 74
Tabla 17. Historia de usuario 10_____ 75
Tabla 18. Historia de usuario 11_____ 75
Tabla 19. Historia de usuario 12_____ 75
Tabla 20. Historia de usuario 13_____ 76
Tabla 21. Historia de usuario 15_____ 76
Tabla 22. Historia de usuario 16_____ 77
Tabla 23. Historia de usuario 17_____ 77
Tabla 24. Historia de usuario 17_____ 78
Tabla 25. Historia de usuario 18_____ 78

Tabla 26. Historia de usuario 19 79
Tabla 27. Historia de usuario 20 79
Tabla 28. Historia de usuario 21 79
Tabla 29. Historia de usuario 22 80
Tabla 30. Historia de usuario 23 81
Tabla 31. Historia de usuario 24 81
Tabla 32. Historia de usuario 25 81
Tabla 33. Historia de usuario 26 82
Tabla 34. Historia de usuario 27 82
Tabla 35. Historia de usuario 28 83
Tabla 36. Historia de usuario 29 83
Tabla 37. Historia de usuario 30 84
Tabla 38. Historia de usuario 31 84
Tabla 39. Historia de usuario 32 84
Tabla 40. Historia de usuario 33 85
Tabla 41. Historia de usuario 34 85
Tabla 42. Historia de usuario 35 85
Tabla 43. Historia de usuario 36 86
Tabla 44. Historia de usuario 37 86
Tabla 45. Historia de usuario 38 86
Tabla 46. Historia de usuario 39 87
Tabla 47. Historia de usuario 40 87
Tabla 48. Historia de usuario 41 87
Tabla 49. Historia de usuario 42 88
Tabla 50. Historia de usuario 43. 88
Tabla 51. Historia de usuario 44. 88
Tabla 52. Historia de usuario 45 89

Tabla 53. Historia de usuario 46 89
Tabla 54. Historia de usuario 47 89
Tabla 55. Historia de usuario 48 90
Tabla 56. Historia de usuario 49 90
Tabla 57. Historia de usuario 50 91
Tabla 58. Historia de usuario 51 91
Tabla 59. Historia de usuario 52 91
Tabla 60. Historia de usuario 53 92

Índice de Imágenes

Imagen 1. Tablero Scrumbam ____ 34
Imagen 2. Modelo entidad relación ____ 39
Imagen 3. Modelo relacional ____ 40
Imagen 4. Diagrama de clases Apigateway ____ 41
Imagen 5. Diagrama de clases Stock ____ 42
Imagen 6. Diagrama de clases Payment ____ 43
Imagen 7. Diagrama de clases Transaction ____ 44
Imagen 8. Diagrama de clases User ____ 45
Imagen 9. Diagrama de Clases Frontend. ____ 46
Imagen 10. Diagrama general de casos de uso ____ 47
Imagen 11. Diagrama de secuencia inicio de sesión ____ 48
Imagen 12. Diagrama de secuencia de registro de usuario ____ 49
Imagen 13. Diagrama de secuencia Traer categorías ____ 50
Imagen 14. Diagrama de secuencia Eliminar categoría ____ 50
Imagen 15. Diagrama de secuencia Crear categoría ____ 51
Imagen 16. Diagrama de secuencia Editar producto ____ 52
Imagen 17. Diagrama de secuencia Crear producto ____ 53
Imagen 18. Diagrama de secuencia Editar producto ____ 55
Imagen 19. Diagrama de secuencia Eliminar producto ____ 55
Imagen 20. Diagrama de secuencia obtener productos ____ 55
Imagen 21. Diagrama de secuencia Crear categoría ____ 56
Imagen 22. Diagrama de secuencia Editar subcategoría ____ 57
Imagen 23. Diagrama de secuencia Eliminar subcategoría ____ 58
Imagen 24. Diagrama de secuencia Obtener subcategorías ____ 59
Imagen 25. Diagrama de secuencia Crea Transacción de Compra ____ 60
Imagen 26. Diagrama de secuencia Crear Transacción de Venta ____ 61

Imagen 27. Diagrama de secuencia de Crear Transacción de Crédito _____ 62
Imagen 28. Diagrama de secuencia de Traer detalle de la transacción ____ 63
Imagen 29. Diagrama de secuencia de Traer transacciones_____ 63
Imagen 30. Diagrama de secuencia de Eliminar transacciones _____ 65
Imagen 31. Diagrama de secuencia de Obtener deudores _____ 65
Imagen 32. Diagrama de secuencia de abonar a la deuda _____ 66
Imagen 33. Diagrama de arquitectura de Software _____ 67
Imagen 34. Diagrama de Componentes _____ 68
Imagen 35. Inició de sesión_____ 93
Imagen 36. Registro. _____ 94
Imagen 37. Aprobación de solicitudes_____ 94
Imagen 38. Modulo Inventario _____ 95
Imagen 39. Editar productos _____ 95
Imagen 40. Crear Productos._____ 96
Imagen 41. Edición y creación de categorías y subcategorías._____ 97
Imagen 42. Creación de transacciones. _____ 97
Imagen 43. Selección de productos. _____ 97
Imagen 44. Historial de transacciones. _____ 98
Imagen 45. Detalle de una transacción. _____ 98
Imagen 46. Módulo de deudores. _____ 99
Imagen 47. Pagos efectuados. _____ 99
Imagen 48. Pruebas inicio de sesión. _____ 100
Imagen 49. Pruebas registro de usuarios._____ 101
Imagen 50. Pruebas Aprobación de nuevos usuarios._____ 102
Imagen 51. Pruebas inventario._____ 103
Imagen 52. Pruebas nuevo producto. _____ 104
Imagen 53. Pruebas edición de productos. _____ 105

Imagen 54. Prueba formularios de subcategorías y categorías. _____ 106
Imagen 55. Pruebas creación de transacciones._____ 107
Imagen 56. Pruebas creación y edición de usuarios. _____ 108
Imagen 57. Prueba historial de transacciones. _____ 109
Imagen 58. Pruebas página de deudores._____ 110

AGRADECIMIENTOS

Erick Chaparro

Agradezco profundamente a mi madre y padre quienes fueron testigos de toda mi formación académica y profesional, quienes a lo largo de mi vida me han apoyado en lo que me he propuesto a realizar.

A mi abuela, que lastimosamente no pudo estar en la etapa final de mi proceso como estudiante en la universidad, pero siempre me daba ánimos para seguir adelante, sin importar la dificultad presentada, y las enseñanzas que me dejó a lo largo de mi vida siempre las recordaré.

A mis compañeros y amigos que hicieron este proceso mucho más ameno con su manera de ayudar y compañía.

Y a mi tutor, por su orientación, apoyo y dedicación en cada etapa de este proceso.

Fredy Angarita

Agradezco a toda mi familia, que ha sido parte fundamental de mi proceso como estudiante y como persona, gracias por el apoyo incondicional que me han brindado durante todas las etapas de mi vida.

También a mis amigos que han sido parte de mi proceso universitario, gracias por su ayuda y acompañamiento durante estos últimos años.

Finalmente agradezco a todos los maestros que han influido en mi vida, no solo como profesional sino como persona.

RESUMEN

Las deudas por parte del cliente es un factor común dentro de todos los negocios de Colombia y el mundo, después de todo a la hora prestar un servicio o vender algún producto, las empresas buscan crear lazos de confianza con sus compradores, para que sean preferidos sobre la competencia.

Sin embargo, lo anterior, también puede generar conflictos con la clientela ya que, si no existe un control exhaustivo sobre las deudas del consumidor, donde se especifiquen exactamente donde y cuando se creó tal movimiento, la perdida de mercancía podría desestabilizar el funcionamiento de la empresa, resultando en problemas a corto y largo plazo.

ABSTRACT

The debt by the customer is a common factor in all businesses in Colombia and the world, after all when providing a service or selling a product, companies seek to create bonds of trust with their buyers, to be preferred over the competition.

However, this can also generate conflicts with customers because, if there is no exhaustive control over consumer debts, where they specify exactly where and when such movement was created, the loss of merchandise could destabilize the operation of the company, resulting in problems in the short and long term.

1. INTRODUCCIÓN

En Colombia, es común el uso de la palabra “fiar” lo que se traduce en términos generales como “deuda”. Esta acción es un acuerdo mayoritariamente verbal entre el comprador y el vendedor; el comprador adquiere un bien o un servicio sin realizar el pago inmediato, en la perspectiva del vendedor, esto genera un compromiso de un pago a futuro, el cual puede estar sujeto a plazos o condiciones especificadas con el cliente al cual se le genera la deuda. Para el vendedor esto puede ser un riesgo financiero, pues se entrega el bien o servicio confiando en que el comprador cumplirá el acuerdo verbal establecido.

Adicionalmente, la deuda para los vendedores presenta un desafío, pues sin un seguimiento adecuado de las deudas, existe un gran riesgo de morosidad, afectado directamente a la liquidez del negocio, así como la disminución en la rentabilidad, el riesgo de crédito y la inestabilidad financiera del negocio [1].

Con lo anterior mencionado, se observa que el seguimiento de la deuda se convierte en una herramienta crucial para cualquier tipo de negocio, sin importar si es grande, mediano o pequeño, y que incluso, este seguimiento de deuda es usada por los bancos, los cuales aparte de mirar en la base de datos, consultan en registros nacionales [2]. Un cliente moroso puede ser algo significativo, así que para prevenir esto, se debe tener un control riguroso de las cuentas pendientes para una mejor gestión del riesgo crediticio y garantizar un equilibrio entre las relaciones que se tiene entre vendedor y comprador, además de que ayuda a optimizar los procesos financieros [3].

Así mismo para gestionar de manera adecuada los cobros de los clientes se deben adoptar políticas claras y procesos eficaces para asegurar un pago oportuno, para ello es importante ofrecer diferentes formas para saldar la deuda pendiente, como por ejemplo un sistema de abonos o generar espacios de diálogos directos con el cliente para discutir opciones para solventar la deuda, esto dará paso a generar y mantener relaciones de confianza con el cliente, evitando así la posible perdidas de clientes potenciales [4].

En el caso del emprendimiento Surtidora y Mayorista LIAN, el seguimiento de las deudas de sus clientes se ha convertido en un problema, ya que el registro de los saldos pendientes se realiza de manera manual por medio de cuadernillos, este método no solo es susceptible a errores humanos sino también limita la capacidad del negocio de llevar un control riguroso y actualizado a la hora de obtener las cuentas a cobrar.

El registro manual implica que el proceso de consulta y actualización es lento y mucho más laborioso que un registro sistematizado, generando almacenamiento inseguro, dificultad para encontrar registros y poca vida útil de estos [5], así mismo, la página de sistemas de gestión OnREP respalda lo expuesto con la siguiente declaración “Un buen software de gestión nos permite analizar y recoger información para ayudar a la toma de decisiones, aumentando así la competitividad de nuestra empresa. De este modo mejora la productividad y nos permite realizar los ajustes necesarios para mejorar en la producción.” [6]. A su vez el hecho de hacer un registro manual por medio de cuadernos provoca que no se pueda obtener información más detallada acerca del inventario perdido por culpa de una deuda no pagada de un cliente, cabe resaltar que la gestión de inventario es un problema complejo debido a la logística que debe tener y que las empresas recurren a las herramientas de software para esta gestión, “Una de las estrategias para enfrentar dicha complejidad consiste en la implementación de herramientas informáticas de software” [6], pero si se escoge un software comercial puede conllevar a fallas, puesto que el entorno o contexto para que estas fueron creadas es diferente a la que requiere el emprendimiento. Por estas razones es importante que el emprendimiento logre adoptar soluciones tecnológicas que permitan sistematizar el seguimiento de las deudas, optimizar procesos de cobro y mejorar la visibilidad de información financiera, esto no solo agilizará el cobro de deudas, sino también, podrá tomar decisiones más informadas al próximo cliente que deseé generar una deuda con un acuerdo verbal.

2. PLANEACIÓN

2.1. Titulo

Aplicación web para el seguimiento de deudas del cliente y gestión de inventario para el emprendimiento Distribuidora Lian

2.2. Planteamiento

En Colombia, es común el uso de la palabra “fiar” lo que se traduce en términos generales como “deuda”. Esta acción es un acuerdo mayoritariamente verbal entre el comprador y el vendedor; el comprador adquiere un bien o un servicio sin realizar el pago inmediato, en la perspectiva del vendedor, esto genera un compromiso de un pago a futuro, el cual puede estar sujeto a plazos o condiciones especificadas con el cliente al cual se le genera la deuda. Para el vendedor esto puede ser un riesgo financiero, pues se entrega el bien o servicio confiando en que el comprador cumplirá el acuerdo verbal establecido.

Adicionalmente, la deuda para los vendedores presenta un desafío, pues sin un seguimiento adecuado de las deudas, existe un gran riesgo de morosidad, afectado directamente a la liquidez del negocio, así como la disminución en la rentabilidad, el riesgo de crédito y la inestabilidad financiera del negocio [1].

Con lo anterior mencionado, se observa que el seguimiento de la deuda se convierte en una herramienta crucial para cualquier tipo de negocio, sin importar si es grande, mediano o pequeño, y que incluso, este seguimiento de deuda es usada por los bancos, los cuales aparte de mirar en la base de datos, consultan en registros nacionales [2]. Un cliente moroso puede ser algo significativo, así que para prevenir esto, se debe tener un control riguroso de las cuentas pendientes para una mejor gestión del riesgo crediticio y garantizar un equilibrio entre las relaciones que se tiene entre vendedor y comprador, además de que ayuda a optimizar los procesos financieros [3].

Así mismo para gestionar de manera adecuada los cobros de los clientes se deben adoptar políticas claras y procesos eficaces para asegurar un pago oportuno, para ello es importante ofrecer diferentes formas para saldar la deuda pendiente, como por ejemplo un sistema de abonos o generar espacios de diálogos directos con el cliente para discutir opciones para solventar la deuda, esto dará paso a generar y

mantener relaciones de confianza con el cliente, evitando así la posible perdidas de clientes potenciales [4].

En el caso del emprendimiento Surtidora y Mayorista LIAN, el seguimiento de las deudas de sus clientes se ha convertido en un problema, ya que el registro de los saldos pendientes se realiza de manera manual por medio de cuadernillos, este método no solo es susceptible a errores humanos sino también limita la capacidad del negocio de llevar un control riguroso y actualizado a la hora de obtener las cuentas a cobrar.

El registro manual implica que el proceso de consulta y actualización es lento y mucho más laborioso que un registro sistematizado, generando almacenamiento inseguro, dificultad para encontrar registros y poca vida útil de estos [5], así mismo, la página de sistemas de gestión OnREP respalda lo expuesto con la siguiente declaración “Un buen software de gestión nos permite analizar y recoger información para ayudar a la toma de decisiones, aumentando así la competitividad de nuestra empresa. De este modo mejora la productividad y nos permite realizar los ajustes necesarios para mejorar en la producción.” [6]. A su vez el hecho de hacer un registro manual por medio de cuadernos provoca que no se pueda obtener información más detallada acerca del inventario perdido por culpa de una deuda no pagada de un cliente, cabe resaltar que la gestión de inventario es un problema complejo debido a la logística que debe tener y que las empresas recurren a las herramientas de software para esta gestión, “Una de las estrategias para enfrentar dicha complejidad consiste en la implementación de herramientas informáticas de software” [6], pero si se escoge un software comercial puede conllevar a fallas, puesto que el entorno o contexto para que estas fueron creadas es diferente a la que requiere el emprendimiento. Por estas razones es importante que el emprendimiento logre adoptar soluciones tecnológicas que permitan sistematizar el seguimiento de las deudas, optimizar procesos de cobro y mejorar la visibilidad de información financiera, esto no solo agilizará el cobro de deudas, sino también, podrá tomar decisiones más informadas al próximo cliente que deseé generar una deuda con un acuerdo verbal.

2.3. Formulación

Teniendo en cuenta los inconvenientes a los que se enfrenta el emprendimiento Distribuidora LIAN a la hora de vender sus productos, y dando la opción de adquirir productos a crédito (fiados), es de vital importancia que el negocio tenga registro

de cuando, quién, cuanto y que productos debe el cliente; así como los productos que debe suplir nuevamente para contar con existencias de los mismos, ¿Cómo podría implementarse una herramienta de software que permita llevar a cabo la recolección de la información referente a las deudas de sus clientes y la correspondiente gestión de inventario según las diferencias transacciones?

3. OBJETIVOS

3.1. Objetivo General

Desarrollar una aplicación web para el seguimiento de deudas por parte de los clientes del emprendimiento Distribuidora LIAN.

3.2. Objetivos Específicos

- Elaborar los módulos de creación de transacciones, seguimiento de deudas por parte de los clientes y gestión de inventario.
- Implementar un sistema de autenticación y autorización para garantizar la seguridad y el control del acceso a la aplicación permitiendo que solo los usuarios autorizados puedan acceder al sistema.
- Diseñar la arquitectura del sistema, definiendo la estructura y comunicación de los componentes, el modelo de datos y las tecnologías a emplear para garantizar la mantenibilidad.
- Realizar pruebas funcionales donde se evidencie el comportamiento esperado de los diferentes procesos que se realizaran en los diferentes módulos

4. ALCANCE – LIMITACIONES

4.1. Alcances

- Diseñar un sistema que optimice los procesos de registro de transacciones (compra, venta o crédito), haciendo uso de tecnologías web como Nuxt para

el Front-End; SpringMVC y Spring Webflux para Back-End. Adcional, implementar un sistema de seguridad basado en JWT que permitirá respaldar la autenticidad del usuario.

- El módulo de inventario tendrá como objetivo poder administrar el stock existente en el negocio Surtidora y Mayorista LIAN, donde se podrán hacer operaciones como editar y eliminar.
- El apartado de editar del módulo de inventario tendrá la responsabilidad de actualizar los precios del producto y cambiar el stock, según sea necesario puesto en ocasiones se necesitará eliminar existencias de este o agregarlas.
- En esta sección del módulo de inventario, se podrá eliminar los productos que no se piensan volver a vender en el puesto de venta de Distribuidora LIAN.
- En el módulo de movimientos se podrá filtrar los movimientos por fecha y tipo, y tener la opción de ver el detalle de la transacción.
- La sección de ver movimiento dará la opción al usuario de observar el detalle de la transacción.
- La opción de crear pago posibilitará al usuario crear abonos de una deuda pendiente de un cliente, y solo se habilitará si el movimiento es una deuda.
- El módulo de deudas será el responsable de almacenar todos los saldos pendientes y agruparlos por cliente.
- La acción de mostrar deudas se encargará de agrupar las transacciones en los que un cliente tiene un saldo pendiente y en caso de querer abonar una cantidad de dinero a la deuda, se podrá redirigir a la opción de ver movimiento y crear un pago.
- El módulo transacción se encargará de crear los movimientos que se ven reflejados en los anteriores apartados y cambiara según el tipo de transacción, ya que difiere si es un pago de contado o una deuda. Estos registros se podrán visualizar en un Excel.
- En el caso de necesitar crear un movimiento de tipo deuda, se desplegará un formulario con campos adicionales como el nombre del cliente, el teléfono y si deja abona alguna cantidad de dinero se podrá registrar en este mismo.
- En caso de hacer una compra para rellenar el stock existente, la opción de “añadir compra” se podrá usar para registrar que producto, cantidad y el precio por el que se compró.

4.2. Limitaciones

- El proyecto al centrarse en solventar los problemas de Distribuidora Lian los principales beneficiados de esta implementación tecnológica serán las personas que administran este negocio.
- El aplicativo solamente estará desarrollado para un entorno web
- El aplicativo no contará con facturación electrónica para el cliente

4.2.1. Roles del Sistema

4.2.1.1. Colaborador

- El colaborador podrá registrarse con su correo y contraseña para acceder al portal de administración
- El colaborador tiene la opción de crear transacciones, bien sea de compra de productos, venta o crédito/fiado al cliente
- El colaborador podrá exportar las transacciones en formato EXCEL en un rango de fechas.
- El colaborador tiene la posibilidad de eliminar alguna transacción en caso de haber cometido algún error o simplemente se devolvieran las existencias.
- El colaborador podrá crear productos con sus respetivas categorías y subcategorías, teniendo en cuenta el precio de compra y venta del producto.
- El colaborador podrá subir una imagen del producto en formato .jpg o .png que no exceda un límite de peso.
- El colaborador puede eliminar existencias de producto del inventario
- El colaborador tiene la posibilidad de editar una categoría o subcategoría
- El colaborador puede editar la información relacionada con el producto
- El colaborador puede acceder al módulo de deudas en el que podrá observar las deudas activas de los clientes
- El colaborador puede ver el detalle de la deuda activa del cliente, en el que se mostrará los productos llevados, el precio de cada uno, el total y el restante de la deuda
- El colaborador podrá registrar un abono que realiza el cliente a la deuda activa

4.2.1.2. Administrador.

- El administrador puede realizar todas las acciones de un colaborador con la diferencia de que tiene acceso al panel de administración de cuentas, en el que podrá gestionar todas las solicitudes de registro.

- El administrador podrá aceptar o rechazar una solicitud.
- El administrador podrá ver el historial de solicitudes previamente respondidas.

5. JUSTIFICACIÓN

La implementación de un sitio web que solvente los problemas del emprendimiento Distribuidora LIAN alusivos al seguimiento de las deudas de sus clientes contribuirá a:

- **Persistencia adecuada:** El sitio web podrá evitar errores de digitación en los registros de las deudas, de la misma manera ayudara a tener una información más confiable, puesto que se solucionará el problema de que se pierdan datos.
- **Información detallada:** El uso de una herramienta software dará paso a que la recolección de los datos sea más detallada sobre los clientes que deben algún dinero al negocio, dando paso a hacer un seguimiento más exhaustivo.
- Presentación de los datos: A diferencia del uso común de métodos de recolección manual como son los cuadernillos, los datos serán fácilmente accesibles para el personal, haciendo mucho más fácil encontrar datos específicos.
- **Minimizar riesgos:** El hecho de “fiar” a un cliente siempre trae consigo una posible pérdida de dinero, por ello tener un historial detallado de deudas de un comprador posibilitara evitar prestar productos a un deudor que ya debe mucho, o que tenga una deuda previa muy antigua, posibilitando así minimizar los riegos de perdidas.
- **Detallar posibles pérdidas:** En el caso de que un deudor no se haga responsable con los pagos, se podrá detallar que productos se perdieron y a su vez la suma monetaria de la perdida.
- **Solución especializada:** El software al ser pensado para las necesidades específicas del emprendimiento Surtidora y Mayorista LIAN, se adaptará a los requerimientos concretos del negocio.

MARCO TEÓRICO

5.1. Marco Referencial

5.1.1. Morosidad

La morosidad se define como el hecho de incumplimiento de un cliente o entidad sobre alguna responsabilidad financiera con otra compañía o persona [8]. Generalmente cuando el tiempo establecido para el pago es excedido, se considerará al deudor como “moroso”, sin embargo, eso no significa que este último no vaya a saldar su deuda, sino que ha incumplido un plazo previamente establecido [9].

Si bien la morosidad afecta a sobre todo a las compañías que prestan algún producto o servicio por medio de un pago a futuro ya que en términos generales pueden llegar a perder dinero, esto también puede generar problemas para el deudor que incumple, puesto que, con el crecimiento de la tecnología y el intercambio de información entre compañías, se han generado métodos para advertir a otros negocios que una persona no es de confianza, estas son las denominadas listas de morosos [9], donde normalmente si alguien incurrió en esta práctica pueden ser fácilmente agregadas y traerle consigo consecuencias como por ejemplo, la negación de préstamos.

Así mismo algunas categorías con las que se suelen clasificar a los deudores son:

Morosos Intencionales

Los morosos intencionales, son aquellos que utilizan diferentes estrategias para incumplir de manera intencionada con los pagos de una deuda, puesto que, a pesar de tener el dinero para saldarla, no tienen la intención de cumplir con su responsabilidad financiera [10][11].

Morosos Circunstanciales

Se les denomina como morosos circunstanciales a aquellas personas que suelen tener un historial de pagos a tiempo y de manera responsable, pero por culpa de una situación temporal se les imposibilito realizar el pago correspondiente, generalmente las empresas suelen negociar un plan de pagos, con el cual el deudor pueda estar más cómodo [11].

Morosos desordenados

Los morosos desordenados son deudores que no tienen un control adecuado sobre sus deudas, aunque en la mayoría de los casos suelen retrasarse ya que no tienen claridad sobre las fechas limites, pero muestran interés en saldar la deuda [11][12].

5.1.2. Gestión de Cobranza

Se denomina gestión de cobranza al proceso de administrar y controlar la recuperación de los pagos pendientes de los clientes, siendo fundamental para asegurar la liquidez de la empresa [13]. Generalmente la definición de políticas para organizar este tipo de procesos internos de la empresa es fundamental no solo para poder recuperar el dinero, sino que a su vez permite generar una cultura de pago responsable entre los clientes.

De igual manera, dentro de las practicas más comunes están:

Cobranza Preventiva

La cobranza preventiva tiene el objetivo evitar que un cliente llegue a la morosidad, para ello, las empresas suelen utilizar métodos de comunicación para recodar a sus clientes las fechas límites del pago, así mismo se encargan de los reclamos y quejas debido a que gran parte de los casos de morosidad suelen ser por alguna inconformidad con el producto o servicio prestado [13][14].

Segmentación de clientes

La segmentación de clientes es una de las practicas comunes dentro de las compañías, si bien depende directamente de la información recopilada previamente sobre sus clientes, la gestión de este tipo de datos da a lugar a identificar posibles deudores morosos o riesgosos [14].

Formas de pagos

Las compañías suelen ofrecer a los clientes diferentes formas de pago cómodas, evitando que el pago sea tedioso y difícil de realizar, logran que el deudor pueda cumplir con sus obligaciones financieras, o en su defecto evitando que sea una excusa para el incumplimiento [15].

Por otra parte, las fechas límites del pago dependerán directamente con el tipo de negocio, ya que dependiendo del ciclo de este se podrán adoptar diferentes fechas a las que normalmente se acostumbra como quincenales o mensuales y que sean mutuamente beneficiosas. [15].

6. ANTECEDENTES

6.1. Odoo

Odoo es una aplicación gratis, con micro transacciones, cuenta con un sistema de inventario para agilizar las operaciones y rutas y visibilidad en tiempo real, es posible automatizar los recordatorios de los proveedores, puede activar los pedidos de compra automáticamente, enviar mensajes sobre los pedidos de compra, confirmar su recepción días antes. Es posible gestionar el inventario en tiempo real, con varias funciones como números de series, lotes, empaquetado, estrategias de almacenamiento, recuentos cíclicos, entre otras más estrategias [16].

6.2. Shopify

Es una plataforma de comercio electrónico que permite a los usuarios publicar sus productos, pero también puede ayudar a gestionar los listados de productos de una forma eficiente, proporcionando un seguimiento de inventario, este contiene una prueba gratis de 14 días y después se debe pagar dependiendo del plan que se requiera. También Shopify genera una comisión por cada transacción de un 2%. Es bastante útil si se requiere trasladar el negocio a una plataforma en línea [17].

6.3. Zoho

Es un software de gestión de inventario que ofrece una gama completa de funciones de inventario, como el ciclo de vida de los clientes, relación con los proveedores, integraciones con CRM, automatización, entre otros más. Su panel de control permite acceder a los datos de stock e inventario de diferentes almacenes, tiene un diseño minimalista que facilita la visión general de la actividad de las ventas o el detalle de otros productos [18].

6.4. Moonflow

Moonflow es una plataforma que ofrece productos software para la gestión de cobranzas enfocada en las necesidades específicas de diferentes sectores comerciales, entre sus principales características tenemos la integración con software interno como ERP, diferentes canales de comunicación con los clientes, automatización de envíos de recordatorios y otras tareas, y generación de reportes automáticos [19].

6.5. Intiza

Intiza ofrece soluciones software para la gestión de finanzas centrándose en la administración de cobranzas y proveedores, ofreciendo así un análisis de

información por medio de tableros, reportes e indicadores en tiempo real, segmentación de clientes según criterios específicos, seguimiento de clientes por medio de historiales personalizados [20].

6.6. Duemint

Duemint es una plataforma que ofrece varias soluciones para la gestión financiera de una empresa, como la conciliación bancaria, automatización de cobranzas, seguimiento del flujo de caja y pago a proveedores, dentro de sus características más destacables tenemos la integración con RMP, flexibilización de métodos de pagos en línea, carga de documentos para reducir el ingreso de manual de datos, y automatización de la comunicación con clientes por medio de flujos personalizados [21].

7. MARCO CONCEPTUAL

7.1. Lenguaje de programación

Es un conjunto de reglas y sintaxis que permite dar instrucciones a una computadora para realizar tareas específicas, actuando como puente entre humano y máquina, de esta manera facilita la creación de programas informáticos que ejecutan las instrucciones definidas en el código escrito [22].

7.2. TypeScript

TypeScript es un lenguaje de programación fuertemente tipado que se basa en JavaScript, agregando sintaxis adicional de JavaScript esto con el fin de que se detecten errores en forma temprana en el editor [23].

7.3. Java

Java es un lenguaje de programación ampliamente utilizado para codificar aplicaciones web, siendo rápido, seguro y confiable, altamente tipado para codificar aplicaciones, siendo útil para la creación de aplicaciones móviles Android y tecnologías del servidor [24].

7.4. Framework

Es un marco de trabajo o esquema que ofrece una estructura base para elaborar un proyecto con objetivos específicos, es más como una plantilla que sirve como un punto de partida para el desarrollo de software, permitiendo acelerar el desarrollo, reducir errores y obtener un resultado de mayor calidad [25].

7.5. NUXT

Nuxt es un framework open source de desarrollo frontend basado en VueJS, que integra herramientas y funcionalidades avanzadas, como el ruteo automático, importación automática de componentes, utilidades de obtención de datos, entre otros.

7.6. Spring Framework

Es un framework más popular para Java empresarial para código de alto rendimiento, liviano y reutilizable, pues, su finalidad es garantizar la agilidad y resolución de problemas que puedan surgir en el desarrollo. Spring ofrece como elemento principal, el soporte de infraestructura a nivel de aplicación, otorgando así, un modelo para configuración como para la programación de aplicaciones web [27].

7.7. Microservicios

Los microservicios son tipo de arquitectura como un modo de crear software, con esta estrategia, las aplicaciones se dividen en elementos más pequeños e independientes entre sí, es decir que cada uno de los elementos o procesos que conforman la aplicación, es un microservicio (microservicio para pagos, inventario, registro o login), con esto se consigue distribuir el software con mayor rapidez [28].

7.8. Base de datos

Una base de datos en una recopilación de datos sistemática y almacenada electrónicamente, pudiendo contener cualquier tipo de datos, palabras, números, imágenes, archivos e incluso vídeos (dependiendo de la base de datos), puede usar un software denominado “sistema de administración de base de datos (DBMS por sus siglas en inglés)” para facilitar las operaciones básicas: almacenar, recuperar, editar y borrar [29].

7.9. SQL

Es un lenguaje de programación para almacenar y procesar información en base de datos relacionales; una base de datos relacional almacena información en forma de tablas, con filas y columnas que pueden representar diferentes atributos de datos y diferentes relaciones entre sí [28]. Con instrucciones SQL es posible realizar las operaciones básicas descritas en el punto anterior.

7.10. PostgreSQL

Es un sistema de base de datos que tiene una buena reputación en el mundo empresarial por su flexibilidad, fiabilidad y alta estabilidad. Proporciona soporte a diferentes funciones de SQL y añadiendo características nuevas que otros motores de base de datos no poseen [30].

7.11. Arquitectura Hexagonal

También es conocida como arquitectura de puertos y adaptadores, este patrón de diseño busca separar la lógica de negocio de la lógica del sistema, con este enfoque, ayuda a garantizar la calidad del software y su escalabilidad. Se basa en que las aplicaciones deben ser independientes de la tecnología subyacente y, por tanto, fácilmente intercambiables [31].

7.12. API Rest

Es una interfaz de programación de aplicaciones que se sigue los principios de diseño del estilo de la arquitectura REST, esto significa transferencia de estado

representacional y consiste en conjunto de reglas y recomendaciones para diseñar una API web, el intercambio de información se maneja mediante formato JSON [32].

7.13. JSON

“Es un formato de texto que forma parte del sistema de JavaScript y que se deriva de su sintaxis, pero no tiene como objetivo la creación de programas, sino el acceso, almacenamiento e intercambio de datos” [33].

7.14. Json Web Token (JWT)

Es un estándar del documento RFC 7519 donde se define un mecanismo para poder propagar entre dos partes, y de formas segura, la identidad del usuario que está ingresando al sistema, cuenta con una serie de privilegios o “claims”, los cuales son codificados en un objeto de tipo JSON, que son inyectados en el cuerpo del mensaje que se envía al servidor [34].

8. IMPACTOS

La correcta implementación del sistema web para el emprendimiento Distribuidora Lian podría tener impactos positivos en múltiples aspectos tanto para el emprendimiento en cuestión, como para los diferentes negocios que estén empezando a emerger en Colombia, estos impactos se muestran a continuación:

8.1. Impacto social

El sistema facilita el acceso a productos incluso bajo la modalidad de crédito (fiado), lo cual representa un apoyo para clientes que no siempre cuentan con la liquidez inmediata, de esta manera, es posible fortalecer la confianza entre vendedor-cliente, contribuyendo a la digitalización de los procesos en negocios que están surgiendo.

Este sistema puede servir como modelo o ejemplo para otros emprendimientos en Colombia, el cual impulsa la modernización, la inclusión tecnológica e incluso la competitividad.

8.2. Impacto económico

Este sistema representa un impulso significativo para el emprendimiento en cuestión, aportando al crecimiento y sostenibilidad, al optimizar procesos de venta y control de deuda, se reducen gastos operativos y se mejora el tiempo de administración de recursos, lo que se traduce, en mayor rentabilidad para el negocio.

Por otro lado, al digitalizar las necesidades del emprendimiento, se obtiene una mayor precisión financiera, donde se reducen los riesgos o el error en el registro de compras o ventas realizadas; adicional, la implementación de tecnologías escalables permite proyectar el crecimiento del negocio, sin necesidad de que se incurran en gastos excesivos de infraestructura, siendo un software económicamente sostenible a largo plazo.

8.3. Impacto tecnológico

De la misma forma, el uso de bases de datos permite asegurar la permanencia de la información a lo largo del tiempo y posibilita la generación de copias de seguridad de manera ágil y masiva. Así mismo la realización de operaciones como búsqueda o filtrado se hacen de manera más eficiente, superando con gran diferencia al registro en papel.

Por último, la arquitectura basada en microservicios brinda una escalabilidad y seguridad, lo que permite que sea adaptable al crecimiento del negocio y posible futuras integraciones con pasarelas de pago u otras plataformas de e-commerce.

8.4. Impacto organizacional

Adicionalmente, la sistematización del control de inventario y el registro de ventas reduce errores y pérdidas, permitiendo una administración más eficiente de los recursos, el cual ofrece un modelo que puede ser replicable para los demás emprendimientos en Colombia que buscan optimizar sus procesos.

9. METODOLOGÍA

9.1. Metodología de Desarrollo

Scrumban.

La metodología elegida para la realización del proyecto es Scrumban. Dicha elección se basa en las ventajas que brinda sobre la gestión de tareas, flujos de trabajos, limitación del WIP, manejo de historias de usuario y planeación, entre otras características destacadas frente a otras prácticas de desarrollo ágil. Además, su enfoque de licitación de requerimientos basada en metodologías agiles [36], permitirá garantizar no solo una visualización constante sobre el progreso del proyecto, sino a su vez obtener retroalimentaciones sin necesidad de esperar un tiempo específico, ofreciendo la oportunidad de generar valor al proyecto constantemente [44].

Las principales características de Scrumban son:

- En la metodología no se definen roles específicos, puesto que, al aplicarla, los roles existentes se mantienen igual, solo cambia la gestión de las tareas [38][39].

- Uso del tablero es una de las piedras angulares de la metodología, ya que en este se asignarán las tareas, y cambiarán el estado de estas, según sea el progreso existente [37].

- El flujo de trabajo en Scrumban son las etapas por las que pasan las tareas antes de ser terminadas, el cual es implementado según sean las necesidades del equipo de trabajo [38][39].

● La limitación del WIP es una característica esencial que promueve la finalización de actividades, a su vez evita la acumulación de tareas en una sola fase del flujo de trabajo [37][39].

● Se manejan historias de usuario, que son una descripción sobre la funcionalidad deseada por el usuario final por cada parte del sistema [38].

● Se hacen reuniones cortas para hablar sobre el progreso del flujo del trabajo, con el fin de resolver problemas que no permitan un avance continuo de las tareas por las diferentes etapas del tablero [37].

● Se usa el Product Backlog que es la lista organizada de funcionalidades que se esperan del software, se actualiza constantemente para conocer exactamente que hace falta por realizar [37][38].

● Manejo de sprints donde se suele entregar un avance de cierta cantidad de funcionalidades [37].

9.2. Implementación.

Para la implementación de Scrumban como metodología de desarrollo se hizo uso de la herramienta Github Projects la cual aplica las características más importantes de Scrumban, permitiendo:

- Visualizar el progreso del proyecto en tiempo real.
- Organizar tareas en fases dentro de un tablero Kanban.
- Administrar los estados del tablero, según las necesidades del proyecto.
- Gestionar la creación de tareas, el cambio de estado.

Github Projects fue elegida por su simplicidad e integración con los repositorios de Github, lo cual permitió clasificar el trabajo asignado a cada integrante según el repositorio de origen, ofreciendo como resultado un contexto más amplio sobre la tarea.

Por otro lado, para la asignación de tareas se realizó un análisis sobre cada módulo necesario para cumplir con los objetivos propuestos, obteniendo como resultado varias historias de usuario.

De la misma forma, para la limitación de trabajo en cada estado del tablero, se tomó la decisión que cada tarea se realizaba hasta finalizarla, esto con el fin de agilizar el desarrollo evitando que existan conflictos en los cambios realizados en el repositorio.

Finalmente se realizaron reuniones ocasionales, con el objetivo de coordinar el desarrollo de funcionalidades, la corrección de errores y discutir sobre el progreso del proyecto.

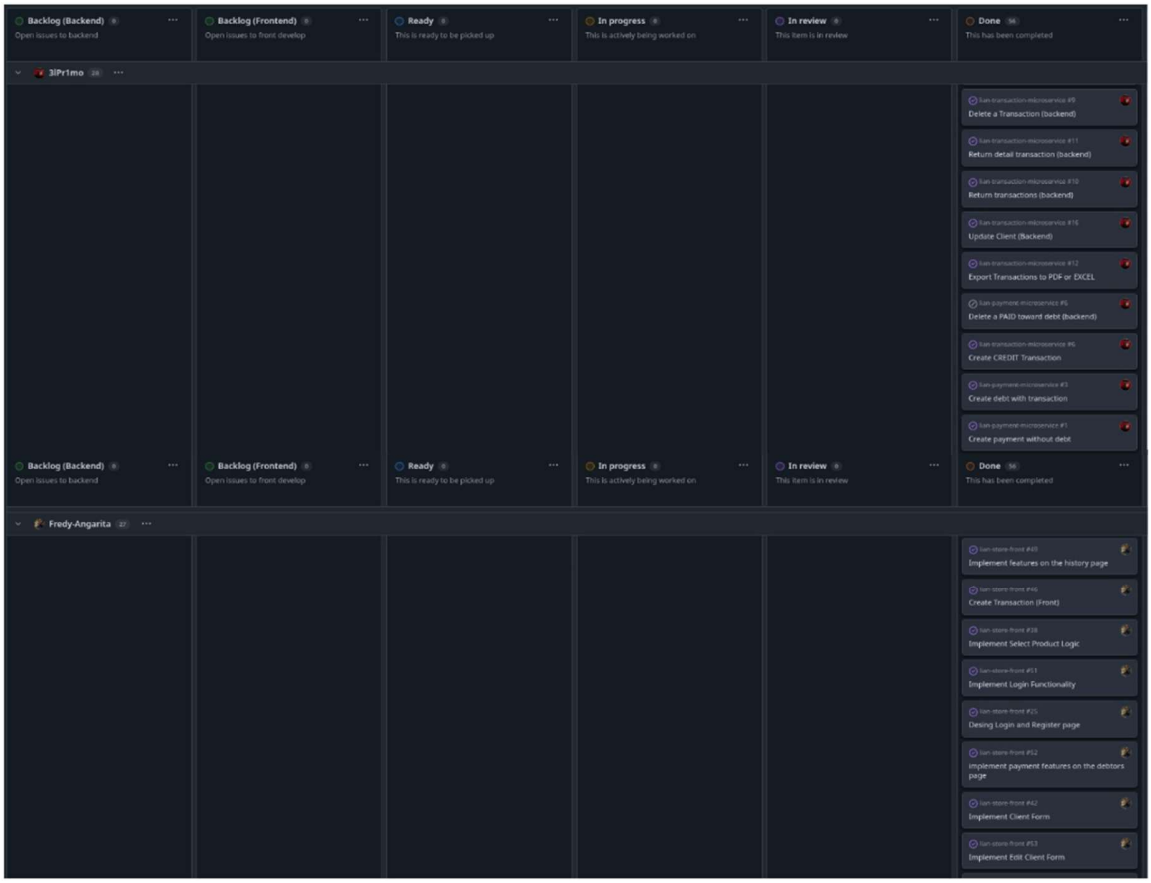


Imagen 1. *Tablero Scrumbam*

10. PRESUPUESTO Y FUENTES DE FINANCIACIÓN

La factibilidad económica del proyecto es alta, pues en términos financieros se incluyen dos equipos de trabajo, asesorías del tutor del proyecto, acceso a Internet, licencia de office para poder realizar el aplicativo web. En las tablas que se presentan a continuación se describe la factibilidad económica, identificando hardware, software y recursos humanos necesarios para realización del proyecto que se propone.

10.1. Tabla de costos de recursos humanos

Tipo	Descripción	Valor * hora	Cantidad mensual	Total * mes	Total 6 meses
Tutor	Personal para el asesoramiento y revisión periódica del proyecto.	$40.000	10 horas	$400.000	$2.400.000
Desarrollador	Personal que se encargara del desarrollo del aplicativo web.	$12.000	100 horas	$1.200.000	$7.200.000
TOTAL					$9.600.000

Tabla 1. *Factibilidad de Recursos Humanos*

10.2. Factibilidad de Recursos Técnicos

Tipo	Descripción	Valor * hora	Cantidad mensual	Total * mes	Total 6 meses
Tutor	Personal para el asesoramiento y revisión periódica del proyecto.	$40.000	10 horas	$400.000	$2.400.000

Desarrollador	Personal que se encargara del desarrollo del aplicativo web.	$12.000	100 horas	$1.200.000	$7.200.000
TOTAL					$9.600.000

Tabla 2. *Factibilidad de Recursos Técnicos*

10.3. Factibilidad de Otros Recursos

Tipo	Descripción	Valor unitario	Cantidad (meses)	Total
Internet	Necesario para la comunicación efectiva entre la empresa y el desarrollador.	$100.000	6	$600.000
TOTAL				$600.000

Tabla 3. *Factibilidad de Otros Recursos*

10.4. Factibilidad Económica Costo Total

Recurso	Valor
Costos totales del recurso humano.	$9.600.000
Costos totales del recurso técnico.	$4.720.000
Costos totales otros recursos.	$600.000
Total	$14.920.000

Tabla 4. *Factibilidad Económica Costo Total*

Siendo el presupuesto final para el proyecto de: **$14.920.000**

11. DECISIONES DE DISEÑO

11.1. Tecnologías.

Con el objetivo de lograr hacer la implementación propuesta, se deben elegir tecnologías tanto para el backend como frontend, las cuales permitan hacer un desarrollo ágil, robusto y escalable.

Por lo anterior se realizó un cuadro comparativo entre tecnologías, destacando características claves para el desarrollo del proyecto.

Tecnologías	Estructura de componentes.	Integración de librerías.	Sistema de enrutamiento integrado	Integración con Typescript.
Angular	Se declara separando lógica, estilos y template (HTML), todo dentro de una misma carpeta con el nombre del componente.	Se debe realizar configuraciones explicitas para el funcionamiento de la librería.	SI	SI
React	Se usa dos archivos, uno para los estilos y otro para la lógica y el HTML	Se debe realizar una configuración manual para el funcionamiento de la librería.	NO	SI
Vue	Se definen en un solo archivo, usando la estructura SFC (Single File Component)	Se debe realizar una configuración manual, pero el sistema de plugins de Vue facilita algunas de estas.	NO	SI
Nuxt	Nuxt al ser un framework basado en Vue, la misma estructura SFC.	Cuenta con su ecosistema de módulos, el cual permite integraciones directas y sin configuraciones adicionales.	SI	SI

Tabla 5. *Comparación tecnologías Frontend.*

Se opto por Nuxt, debido a su gran integración de librerías externas, permitiendo agregar bibliotecas de componentes, iconos y estilos de manera sencilla y rápida, así mismo, el sistema de enrutamiento automático es de vital importancia ya que

permite la separación de funcionalidades en rutas, sin hacer instalaciones o configuraciones extras.

Por otro lado, Nuxt al tener una estructura de directorios bien definida donde separa componentes, paginas, layouts, middlewards y composables, posibilita la importación automática de estos elementos y al mismo tiempo fomenta una organización del proyecto clara y escalable.

Para la parte del Back-End, se tenían algunas tecnologías candidatas, teniendo en cuenta los criterios que se muestran en la siguiente tabla:

Criterio	**Spring**	**NestJS**	**Django**
Lenguaje	Java	TypeScript	Python
Escalabilidad	Altamente escalable, aunque es diseñado principalmente para entornos empresariales	Escalabilidad horizontal, aunque para su optimización, puede requerir mayor esfuerzo	Escalable para aplicaciones medianas
Integración con otros servicios	Excelente integración con Kafka, RabbitMQ, Docker, AWS, Hibernate, Eureka, entre otros.	Buena integración con tecnologías modernas como GraphQL, WebSockets.	Tiene una buena integración, pero su enfoque no es para entornos empresariales
Seguridad	Hace uso de Spring Security (modulo nativo) el cual es un apartado robusto y posee una buena integración con JWT, OAuth2 y Roles	Provee algunos módulos, aunque su seguridad depende de librerías externas	Se incluye Django Auth, el cual ofrece seguridad, pero poca flexibilidad.
Soporte para microservicios	Excelente soporte con herramientas como Spring Cloud, OpenFeign, Netflix OSS o Eureka	Buen soporte con Nest microservices module o gRPC, sin embargo, no ofrece una madurez acorde	Es posible con Django REST y algunas herramientas externas, sin embargo, no son módulos nativos

Tabla 6. *Comparación tecnologías Backend*

Luego de tener en cuenta estos aspectos, se optó por Spring, puesto que cumple con todos los requisitos de seguridad, fácil implementación y una escalabilidad prácticamente excelente, esto último es importante, pues si el emprendimiento llegase a expandirse, las necesidades de optimización de procesos se van a ampliar y con ello el software implementado también.

11.2. Modelo de datos

El modelo de datos en un aplicativo de software es importante, puesto que es la representación gráfica de la estructura que se va a emplear en la base de datos, ayudando a entender los datos a manejar y tomar las decisiones tecnológicas de manera correcta para su almacenaje y tratamiento respectivo.

11.2.1. Modelo entidad relación

El modelo entidad relación nos ayuda a entender a grandes rasgos como va a estar conformado la base de datos, que entidades entran en la lógica de negocio y como se relacionan.

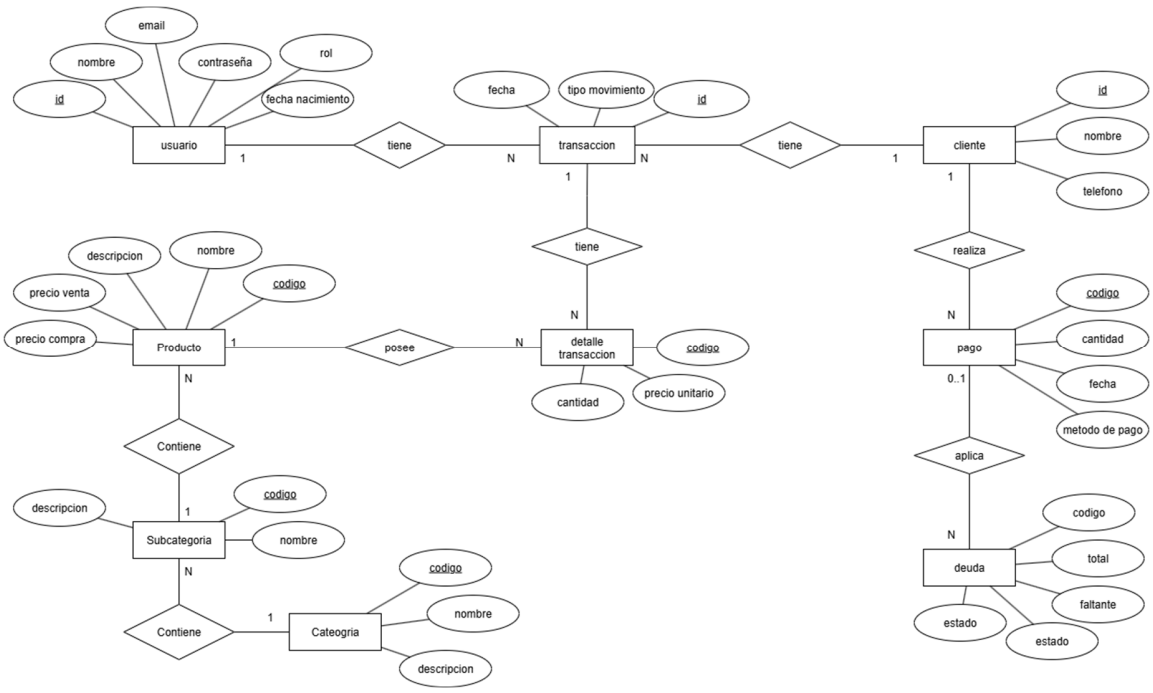


Imagen 2. *Modelo entidad relación*

11.2.2. Modelo Relacional

En el modelo relacional, se muestra las tablas a implementar con su tipo de dato en cada columna, las relaciones entre las tablas, sus llaves primarias y foráneas. Se optó por utilizar una base de datos relacional (SQL) para un mejor ordenamiento y almacenaje de datos.

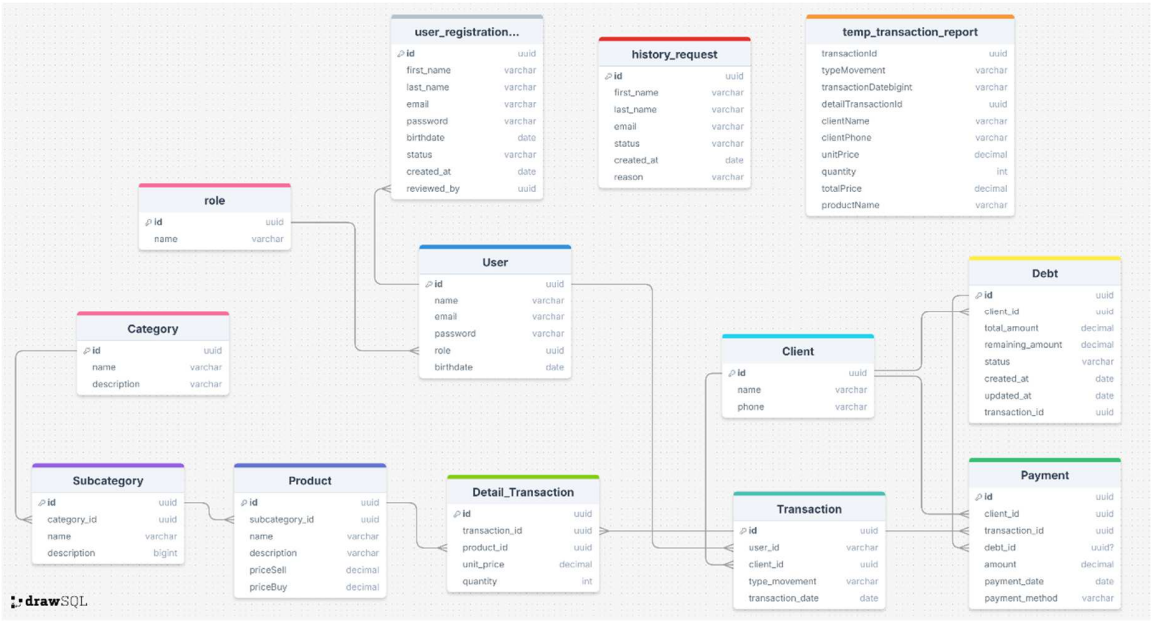


Imagen 3. *Modelo relacional*

11.3. Diagrama de Clases Backend

Este diagrama ofrece una representación visual de la estructura del sistema y sus microservicios, mostrando las clases que lo componen junto con sus atributos, métodos y relaciones entre ellas. Permitiendo visualizar su organización y conexión de la lógica de negocio dentro del Backend.

11.3.1. Apigateway

El diagrama de clases del apigateway, es importante, puesto que el apigateway al encargarse de toda la parte de enrutamiento, se debe saber, la interacción entre las clases que tiene y sus clases de configuración.

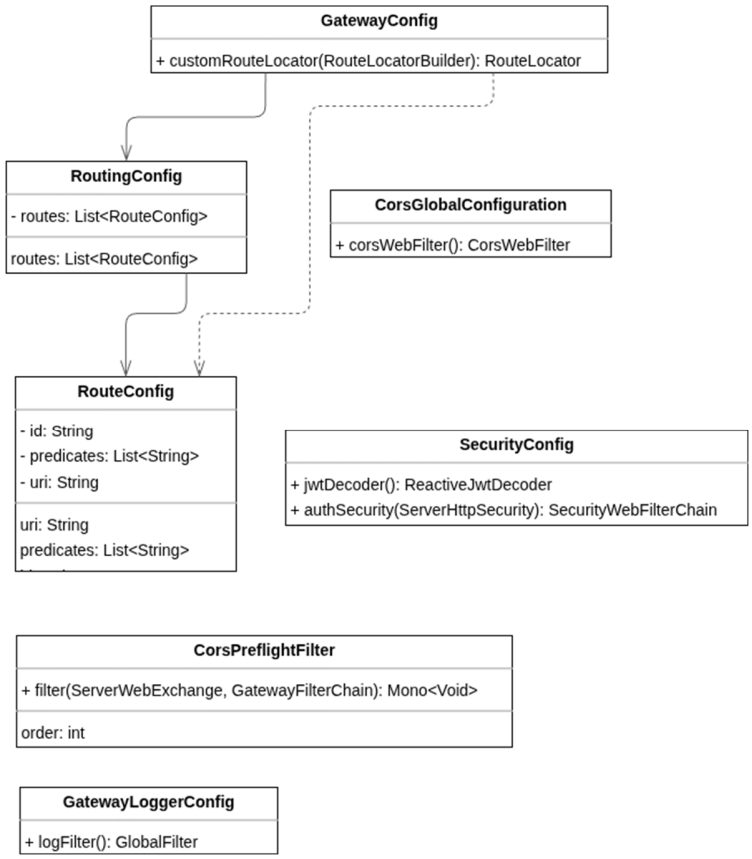


Imagen 4. *Diagrama de clases Apigateway*

11.3.2. Microservicio de Stock

En la siguiente imagen, se muestra la interacción entre las clases que componen el microservicio de stock para toda la lógica de negocio referente a la gestión de inventario.

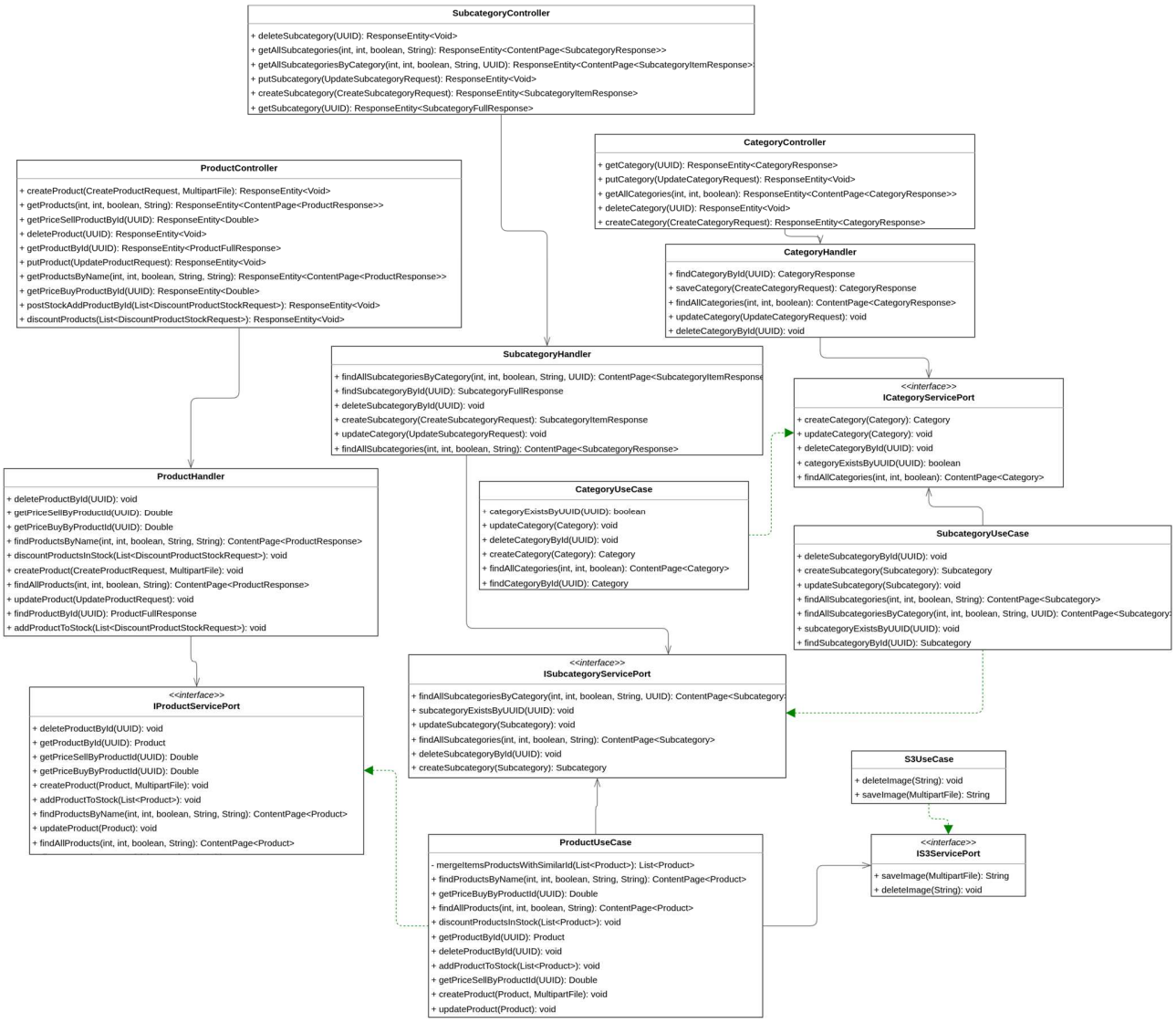


Imagen 5. *Diagrama de clases Stock*

11.3.3. Microservicio de Payment

Este diagrama, es la representación gráfica de la conexión entre diferentes clases para llevar a cabo los procesos de registro de pagos y trazabilidad de la deuda.

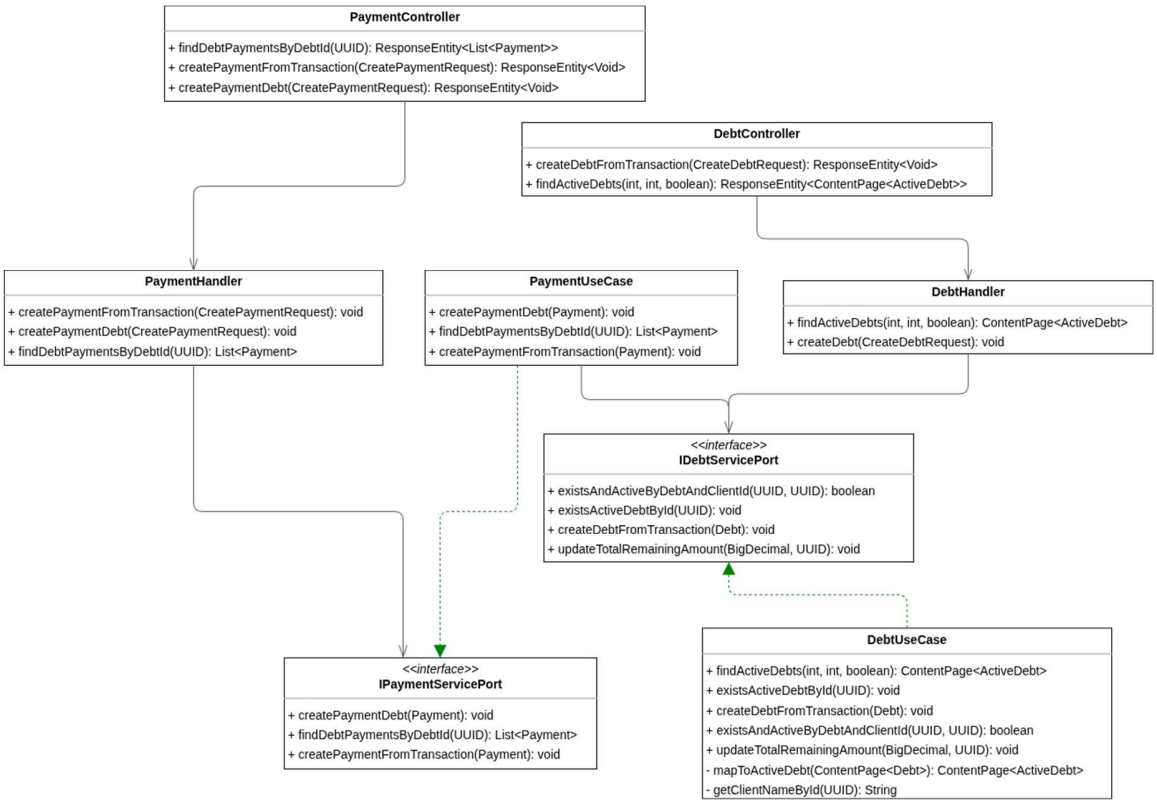


Imagen 6. *Diagrama de clases Payment*

11.3.4. Microservicio de Transaction

El diagrama de clases del microservicio de transacción es uno de los más importantes, pues como se ha mencionado a lo largo del documento, este actúa como orquestador. En el diagrama es posible observar cómo interactúan las clases entre sí y sus correspondientes métodos para la lógica de negocio, este cuenta con su clase principal “TransactionUseCase” en cual contiene toda la lógica de negocio cuando se realiza una transacción y al realizar una operación, dispara las demás acciones al resto de clases y a su vez, a los diferentes microservicios.

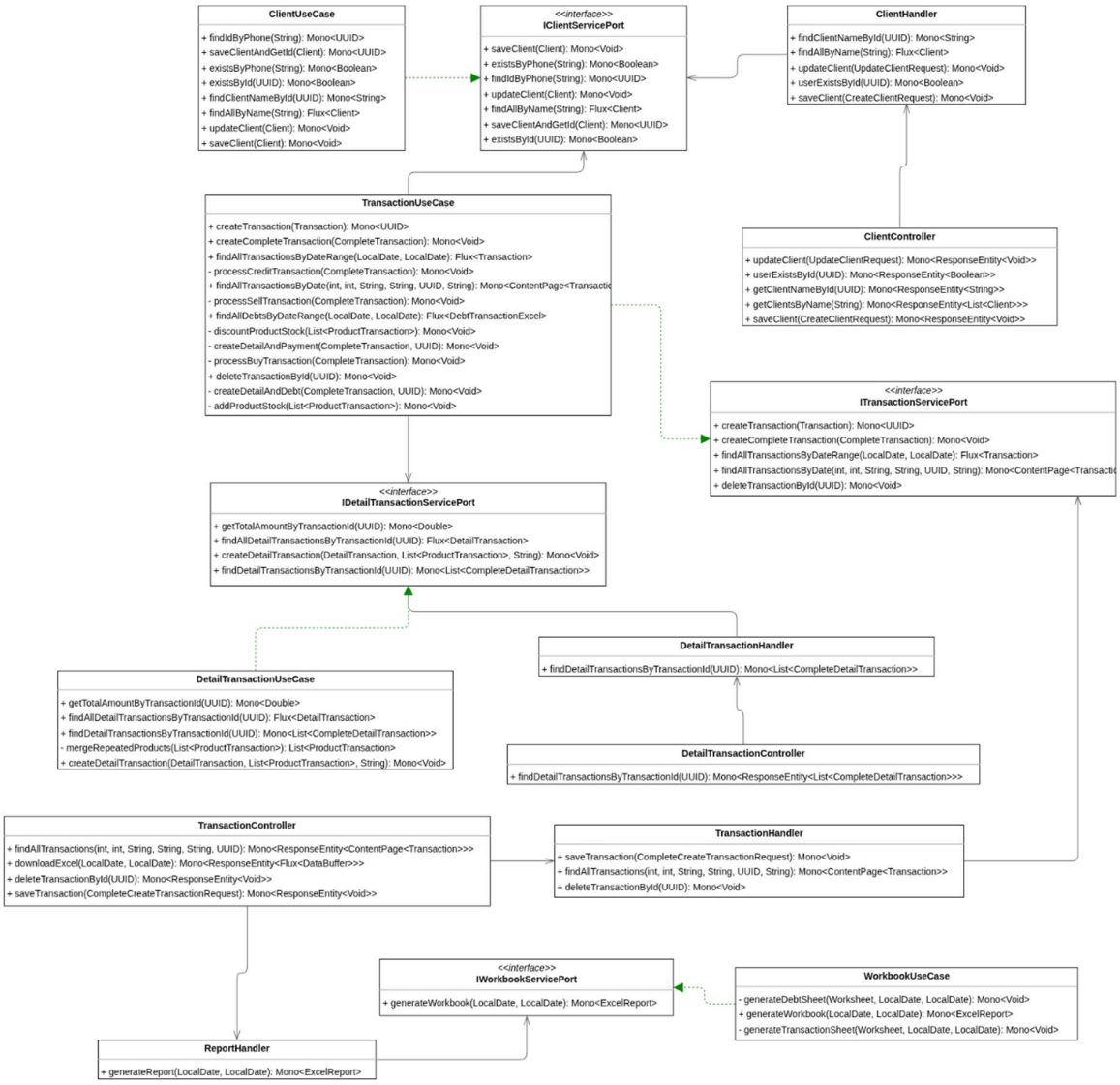


Imagen 7. *Diagrama de clases Transaction*

11.3.5. Microservicio de User

En este diagrama, se muestra cómo se encuentra organizada la lógica de negocio de todo lo relacionado con creación de usuarios, panel de administración, login, entre otros.

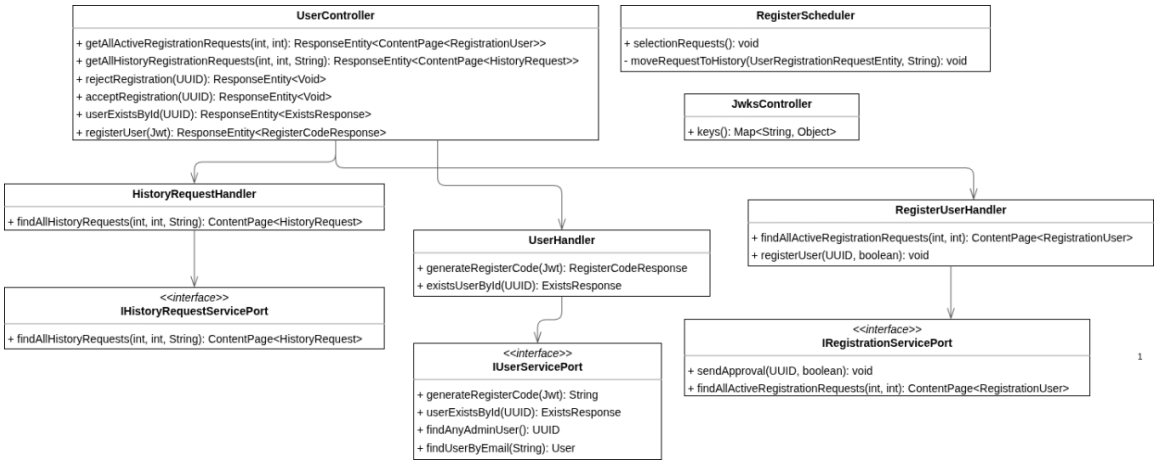


Imagen 8. *Diagrama de clases User*

11.4. Diagrama de Clases Frontend

En el siguiente diagrama se representa tanto los componentes junto a su función de ciclo de vida “OnMounted”, como los composables (funciones de VueJS para la reutilización de lógica), con los que interactúan para hacer las peticiones correspondientes al Backend.

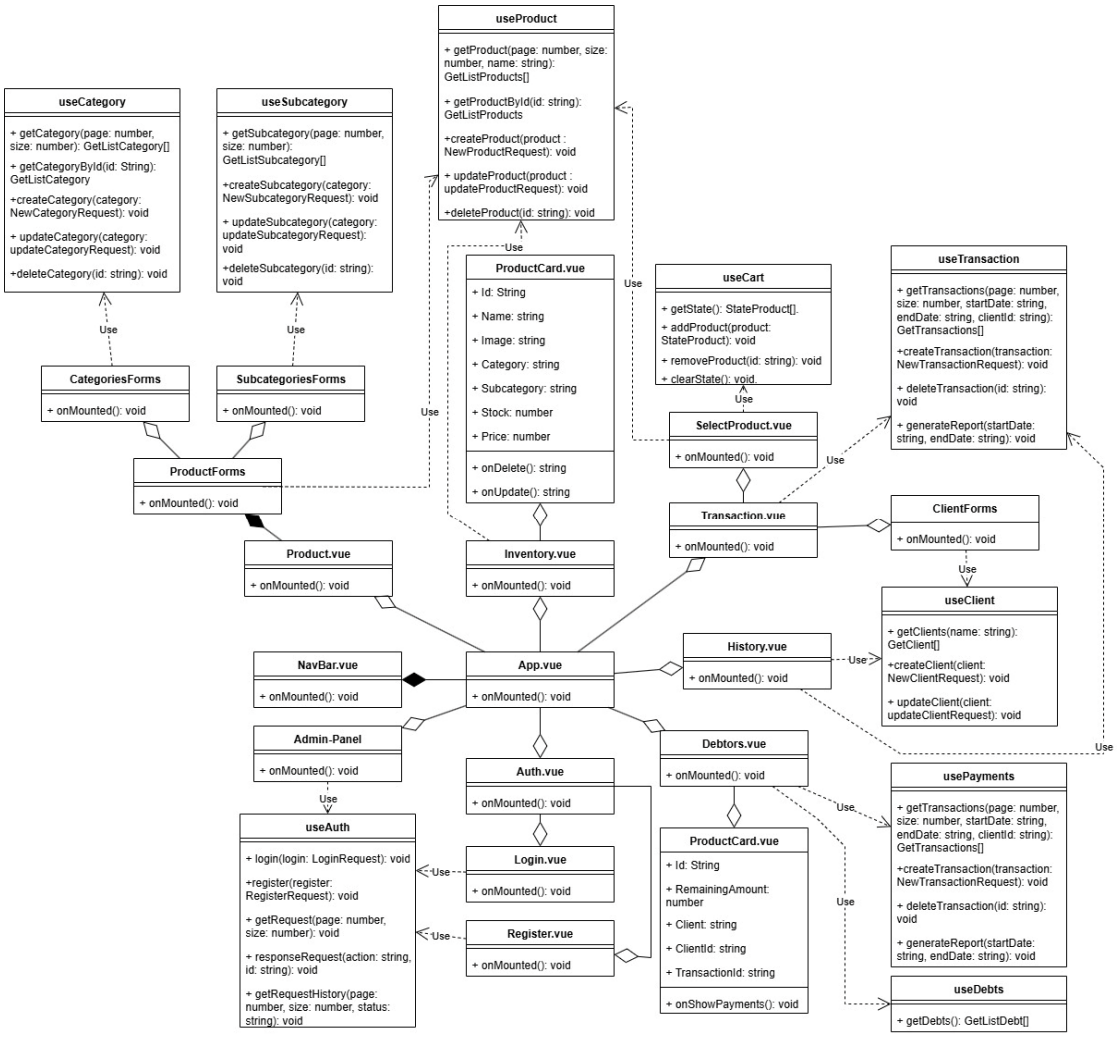


Imagen 9. *Diagrama de Clases Frontend.*

11.5. Diagrama General de Casos de Uso

Con el diagrama general de casos de uso se puede visualizar los diferentes escenarios de interacción que tienen los actores que utilizan el software. Donde se especifica que los actores de colaborador y administrador podrán interactuar con los módulos de inventario, transacción y deudores, así como el de autenticación con las opciones como registro e ingreso. La diferencia principal entre los dos roles se encuentra, en que solo el administrador podrá obtener las solicitudes activas y respondidas de registro de usuarios, así como aprobar o rechazar dicha petición.

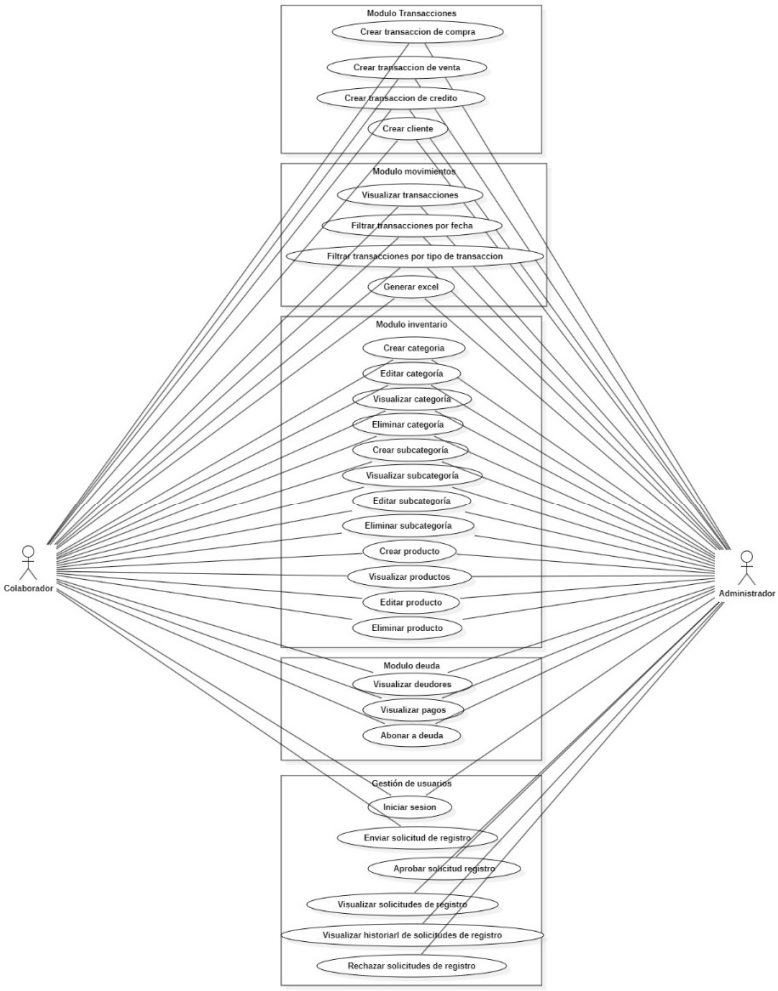


Imagen 10. *Diagrama general de casos de uso*

11.6. Diagramas de Secuencia

Se lleva a cabo un análisis detallado de los procesos que realizará el aplicativo web para saciar las necesidades del emprendimiento, divididas por módulos.

11.6.1. Secuencia de inicio de sesión

Esta secuencia demuestra la comunicación de los componentes tanto frontend como backend e interacción con el usuario para el proceso de inicio de sesión.

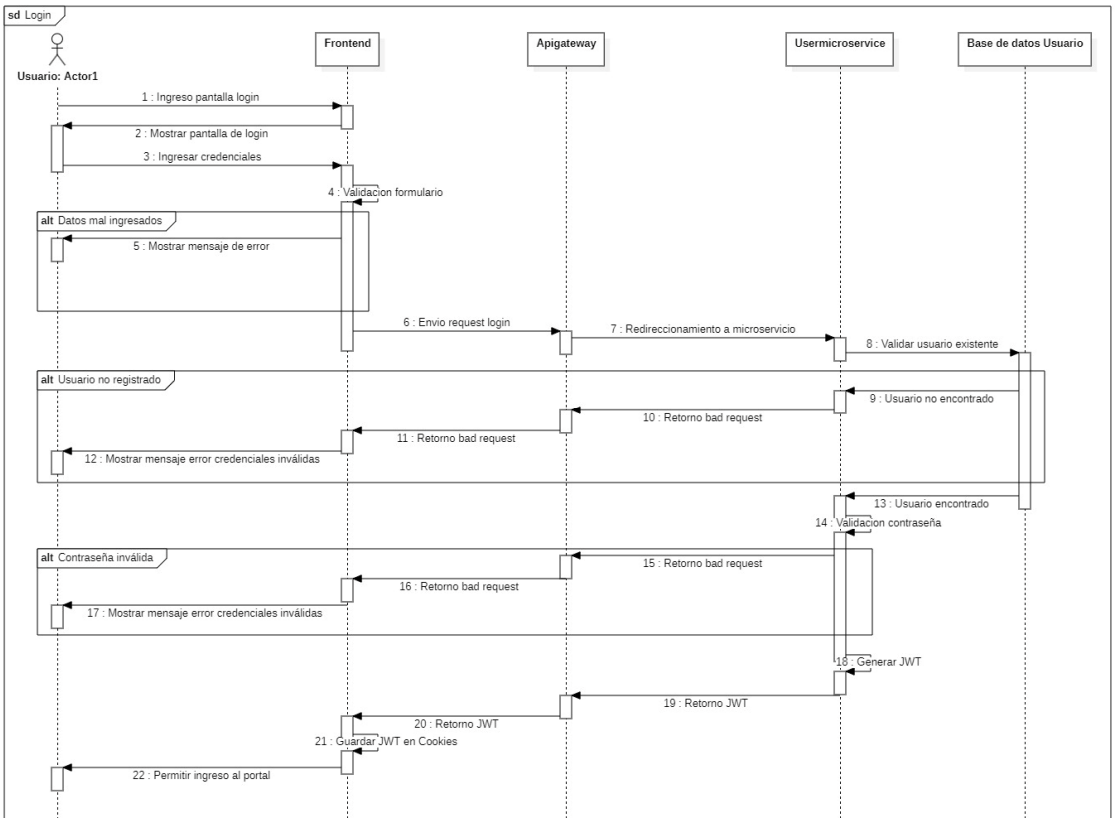


Imagen 11. *Diagrama de secuencia inicio de sesión*

11.6.2. Secuencia de registro de usuario

Esta secuencia implica el comportamiento que debe tener el sistema para registrar un usuario al portal.

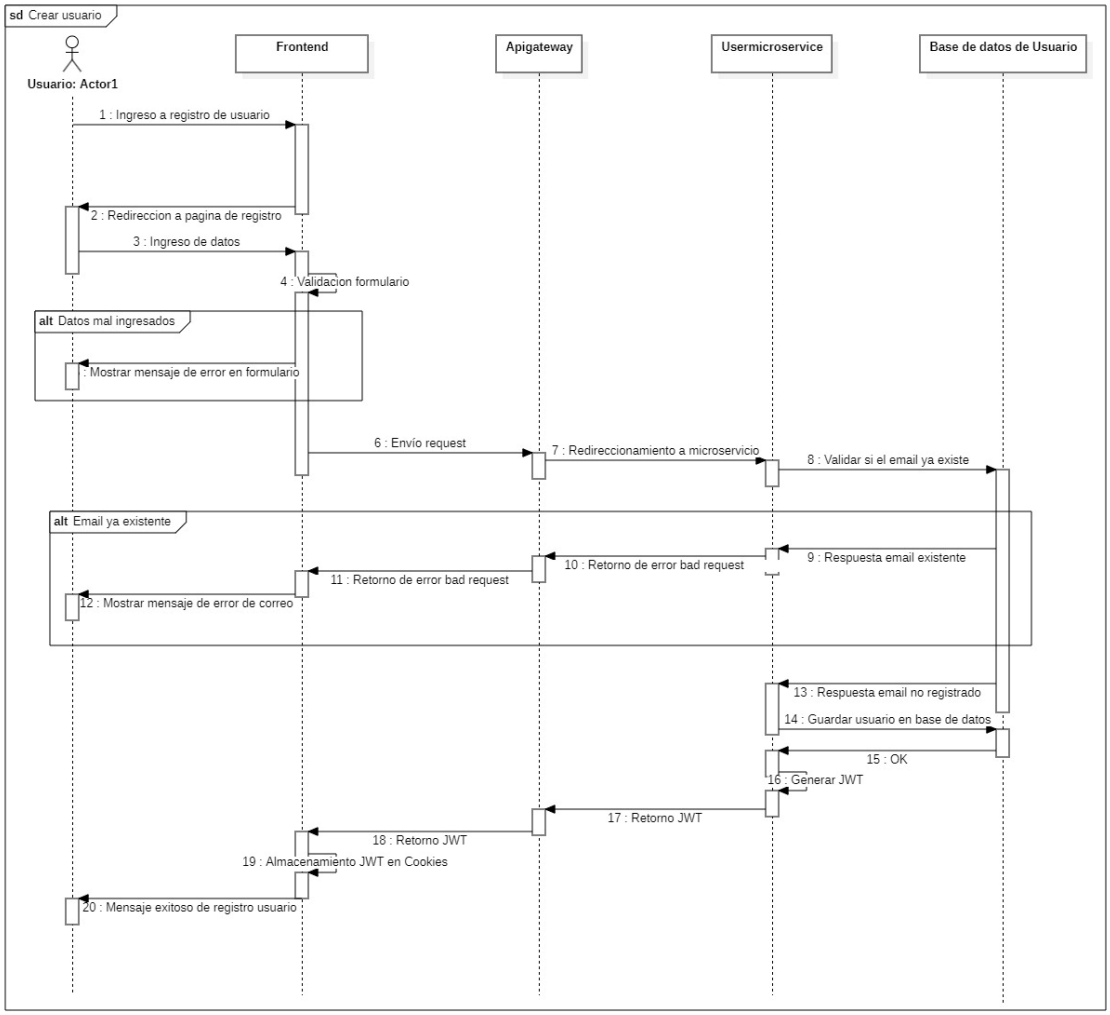


Imagen 12. *Diagrama de secuencia de registro de usuario*

11.6.3. Secuencias modulo inventario

Una vez el usuario haya iniciado sesión, tendrá habilitado los diferentes módulos para acceder y realizar diferentes operaciones, entre ellos el módulo de inventario que engloba toda la gestión que se debe tener para la administración de productos.

- **Obtener categorías**

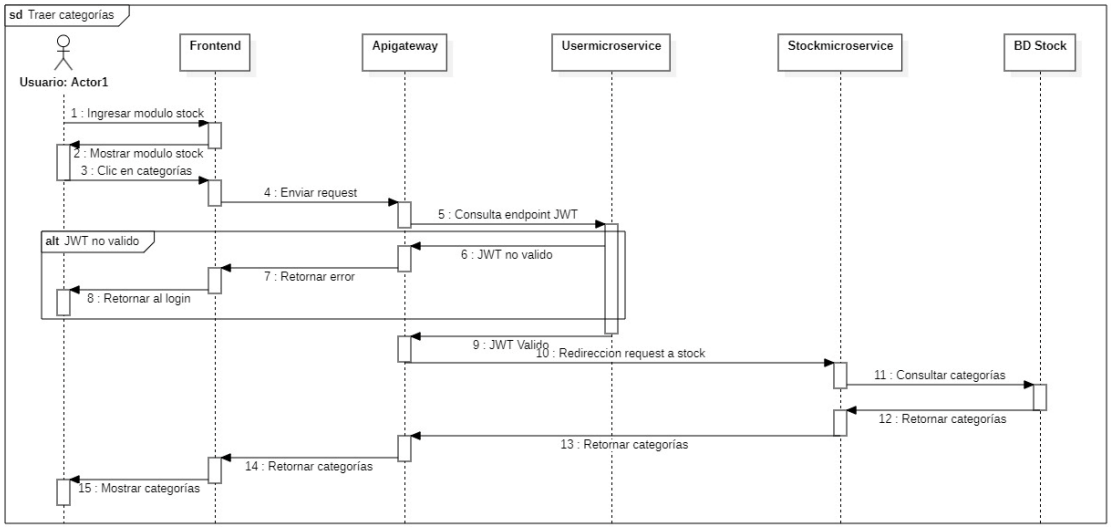


Imagen 13. *Diagrama de secuencia Traer categorías*

- **Eliminar categoría**

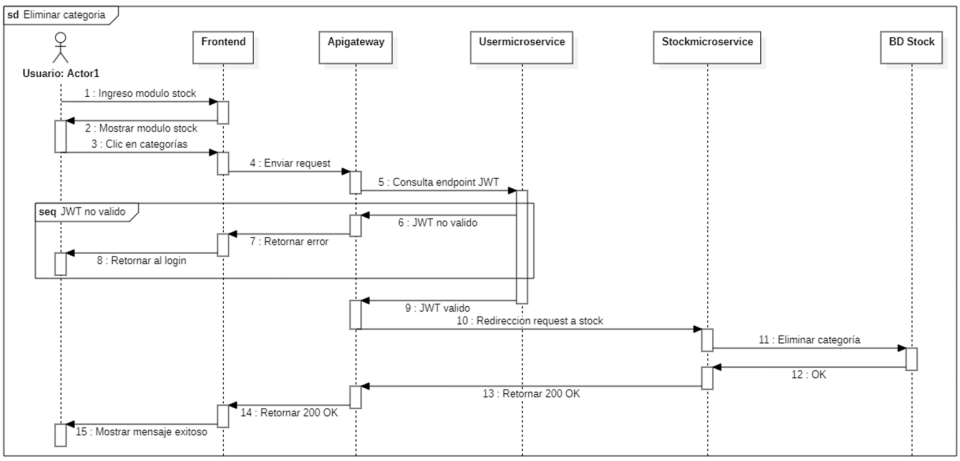


Imagen 14. *Diagrama de secuencia Eliminar categoría*

- **Crear categoría**

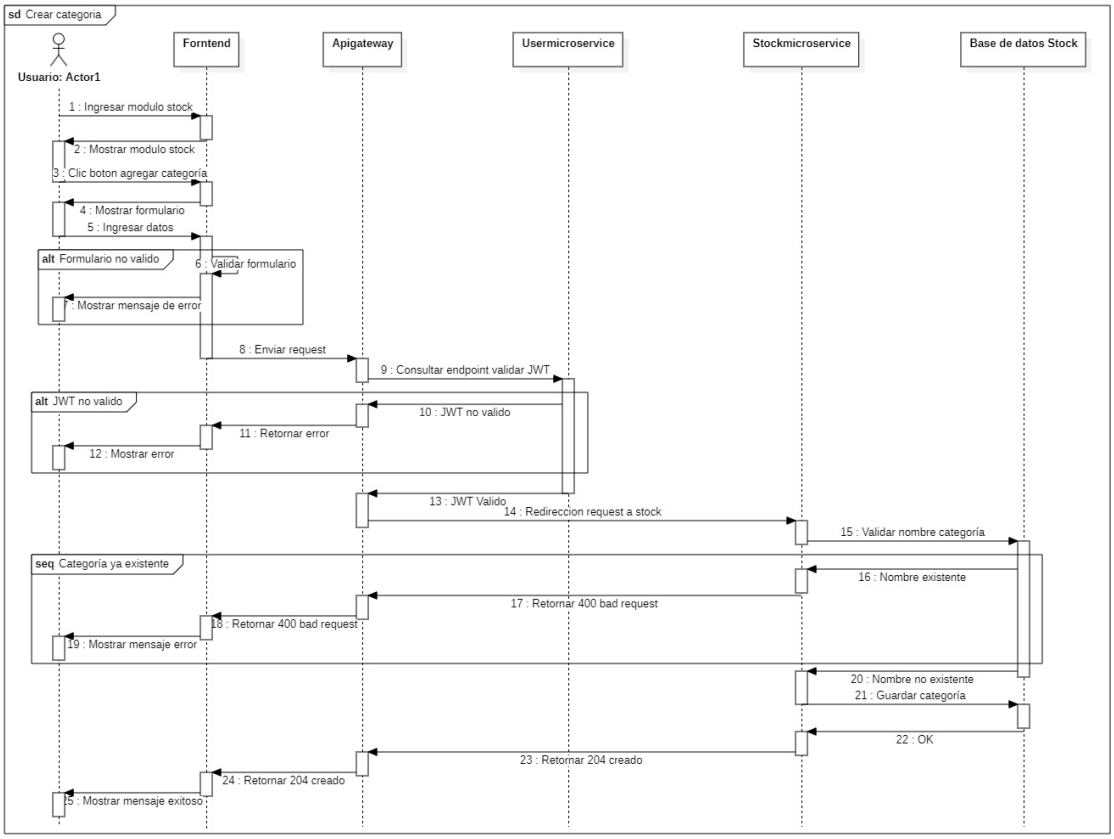


Imagen 15. *Diagrama de secuencia Crear categoría*

- **Actualizar categoría**

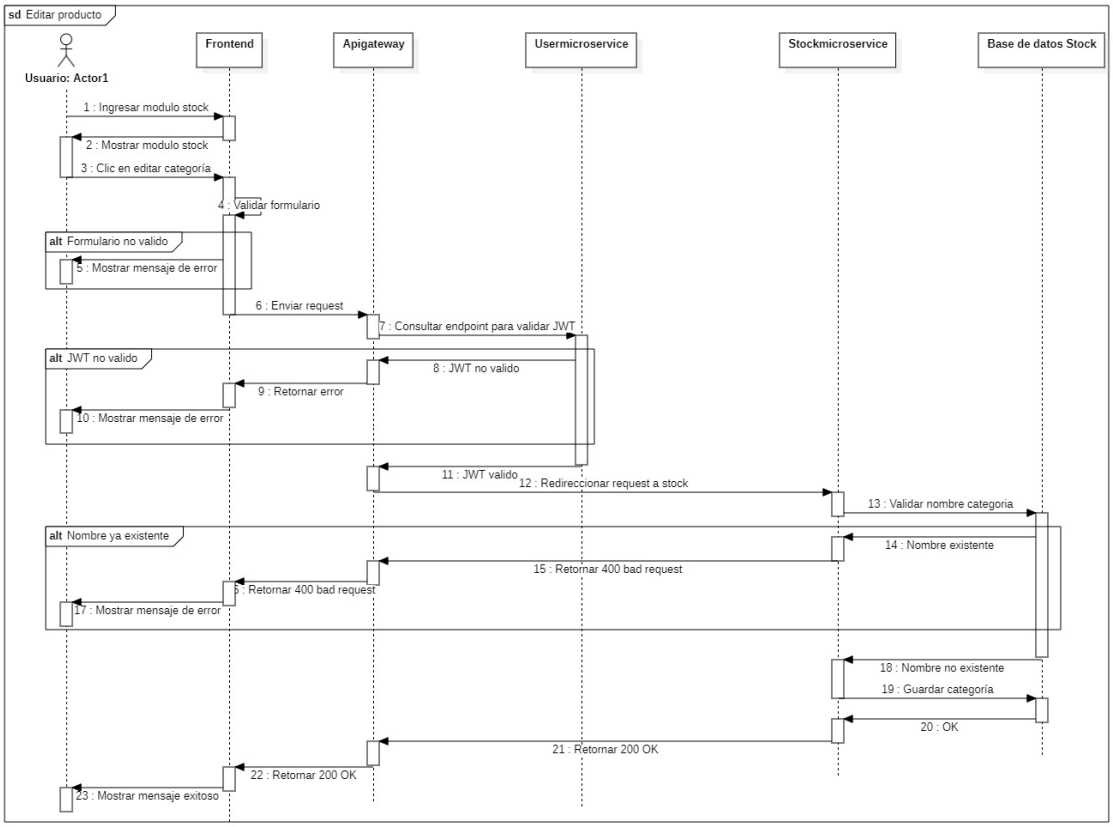


Imagen 16. *Diagrama de secuencia Editar producto*

- **Crear producto**

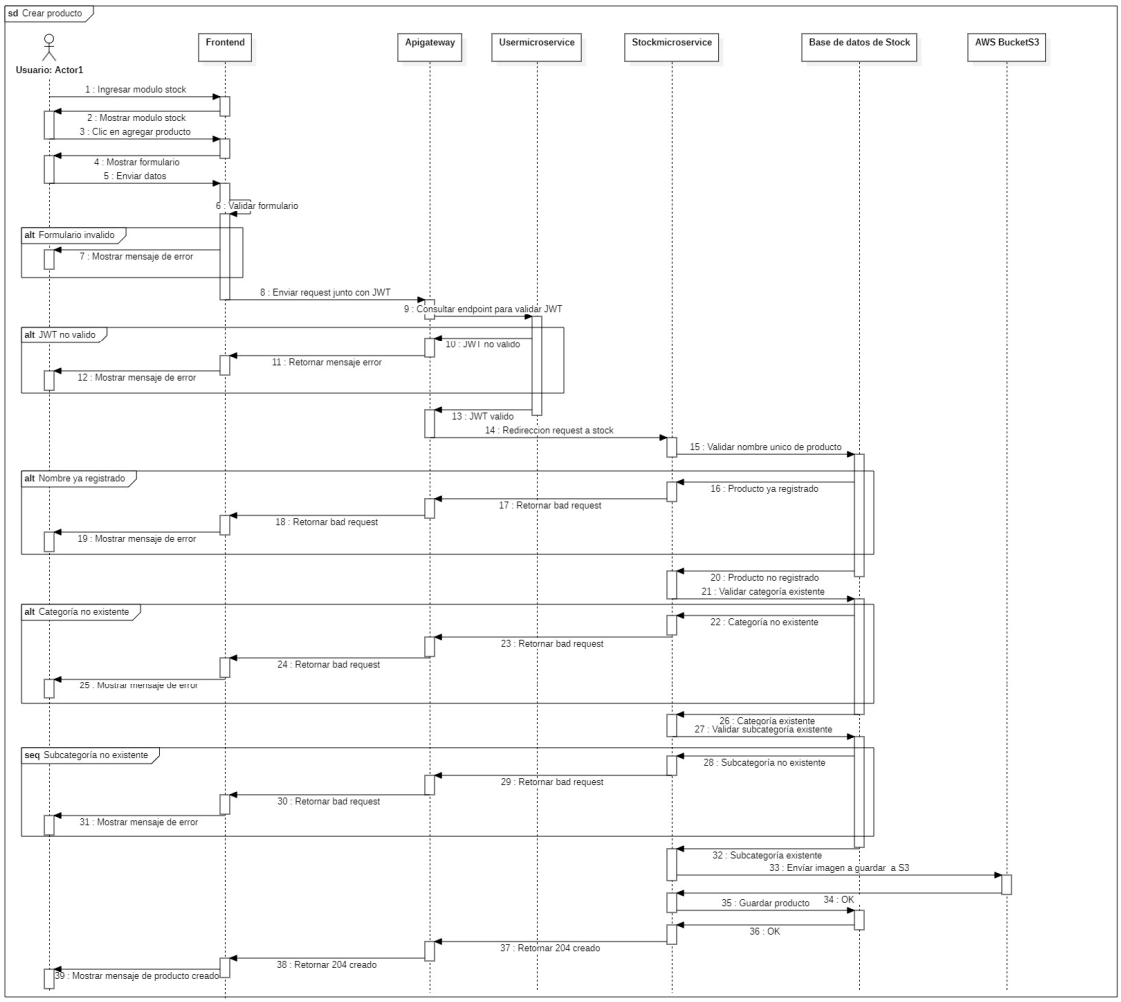


Imagen 17. *Diagrama de secuencia Crear producto*

- **Editar producto**

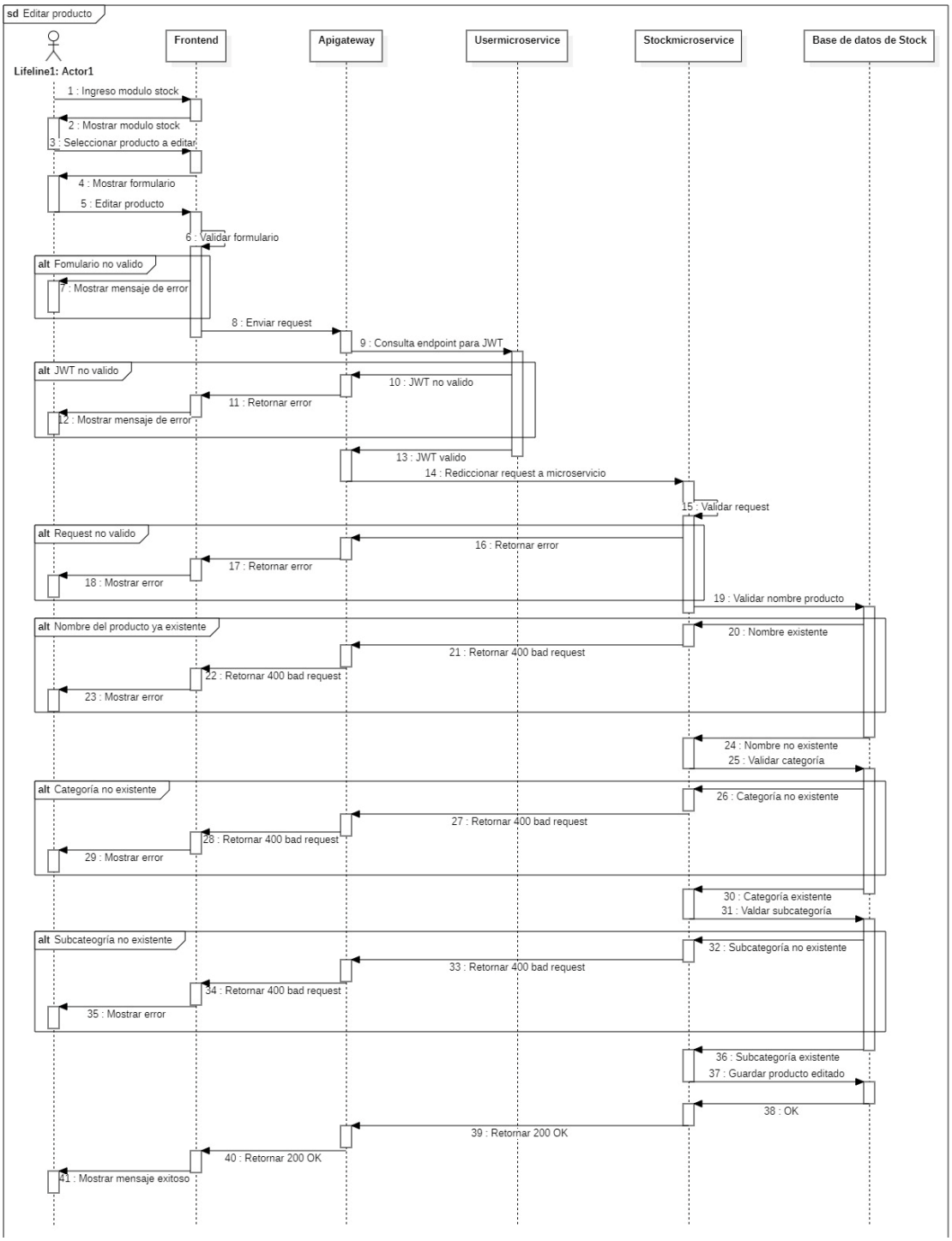


Imagen 18. *Diagrama de secuencia Editar producto*

- **Eliminar producto**

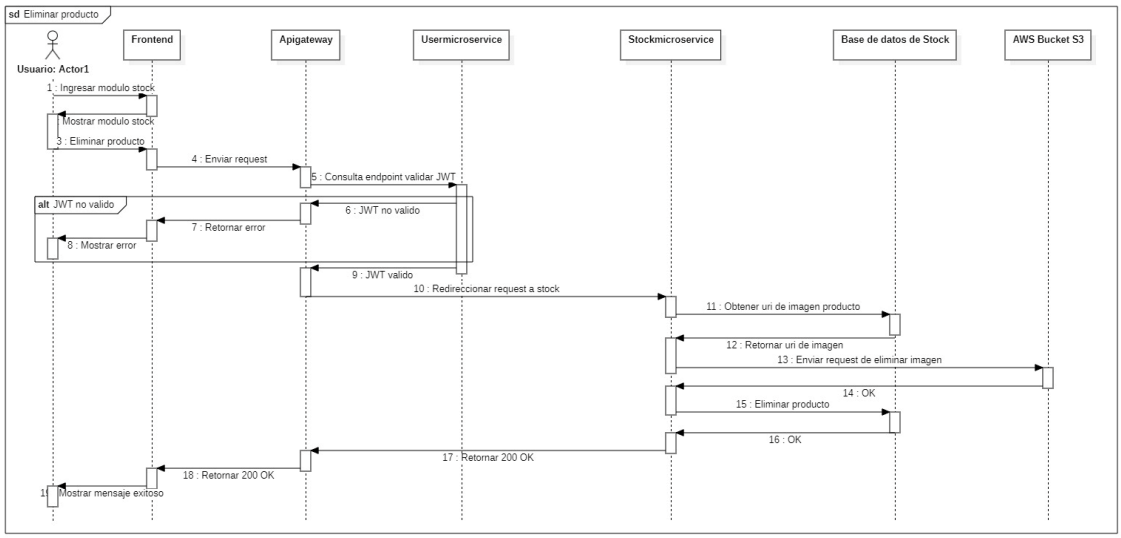


Imagen 19. *Diagrama de secuencia Eliminar producto*

- **Obtener productos**

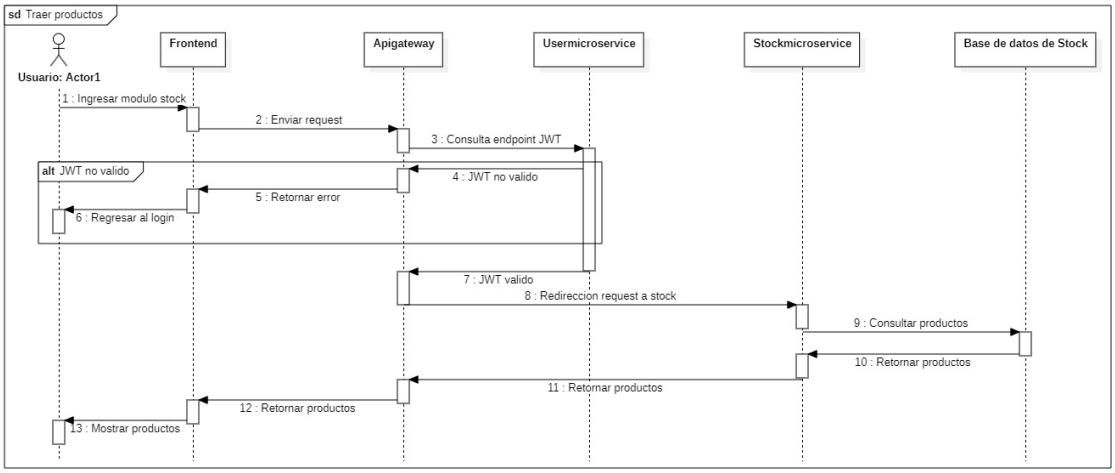


Imagen 20. *Diagrama de secuencia obtener productos*

- **Crear subcategoría**

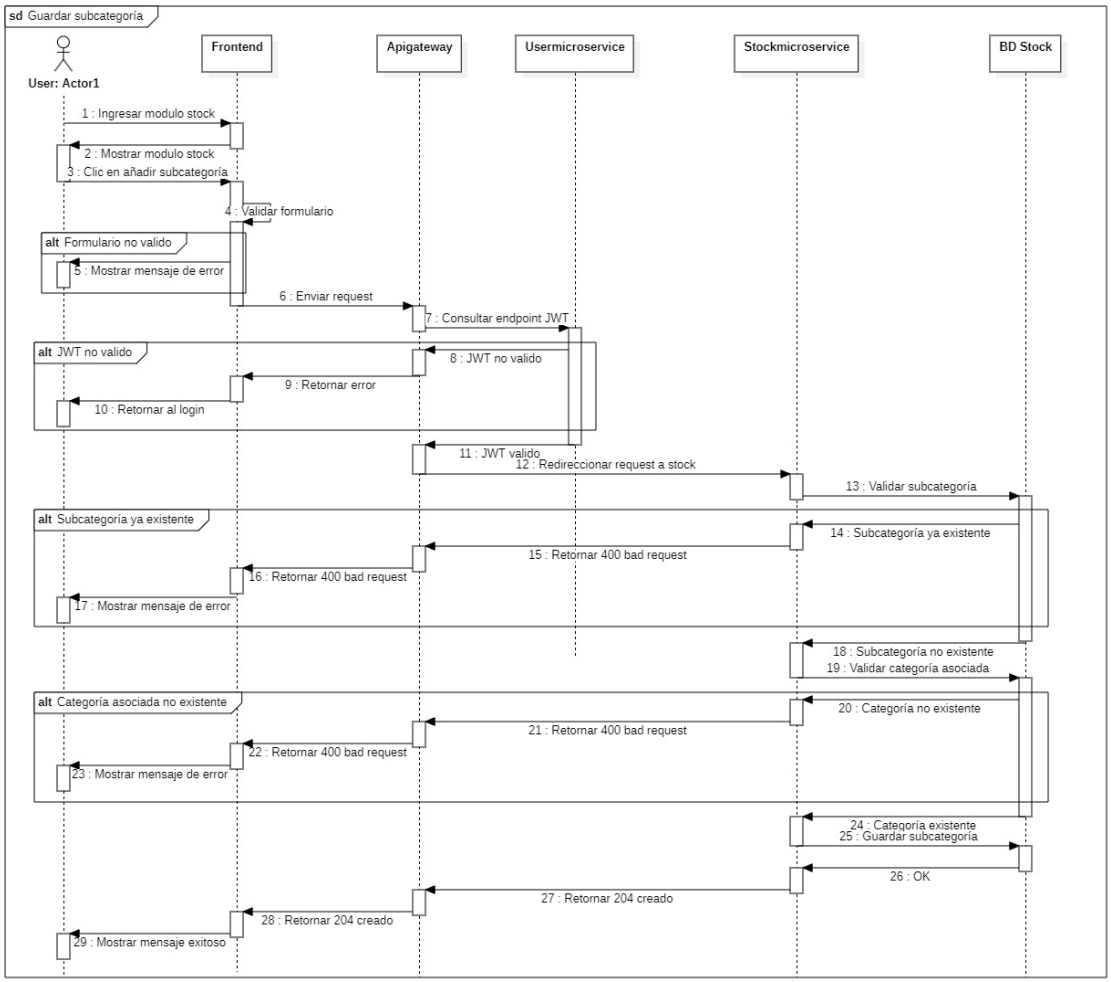


Imagen 21. *Diagrama de secuencia Crear categoría*

- **Editar subcategoría**

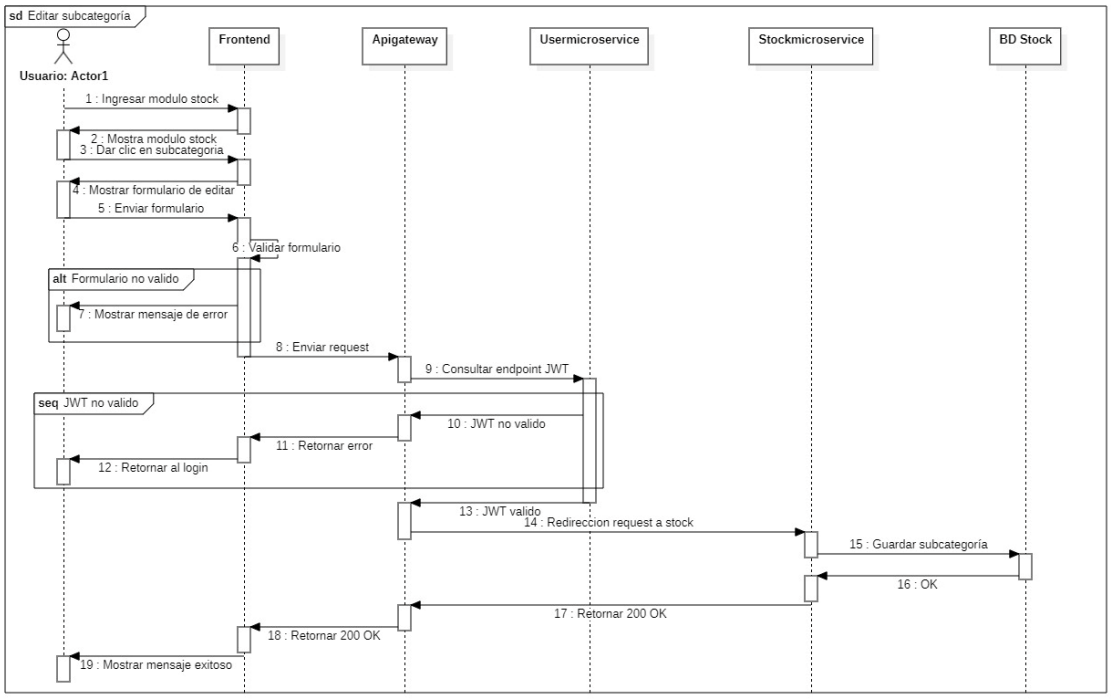


Imagen 22. *Diagrama de secuencia Editar subcategoría*

- **Eliminar subcategoría**

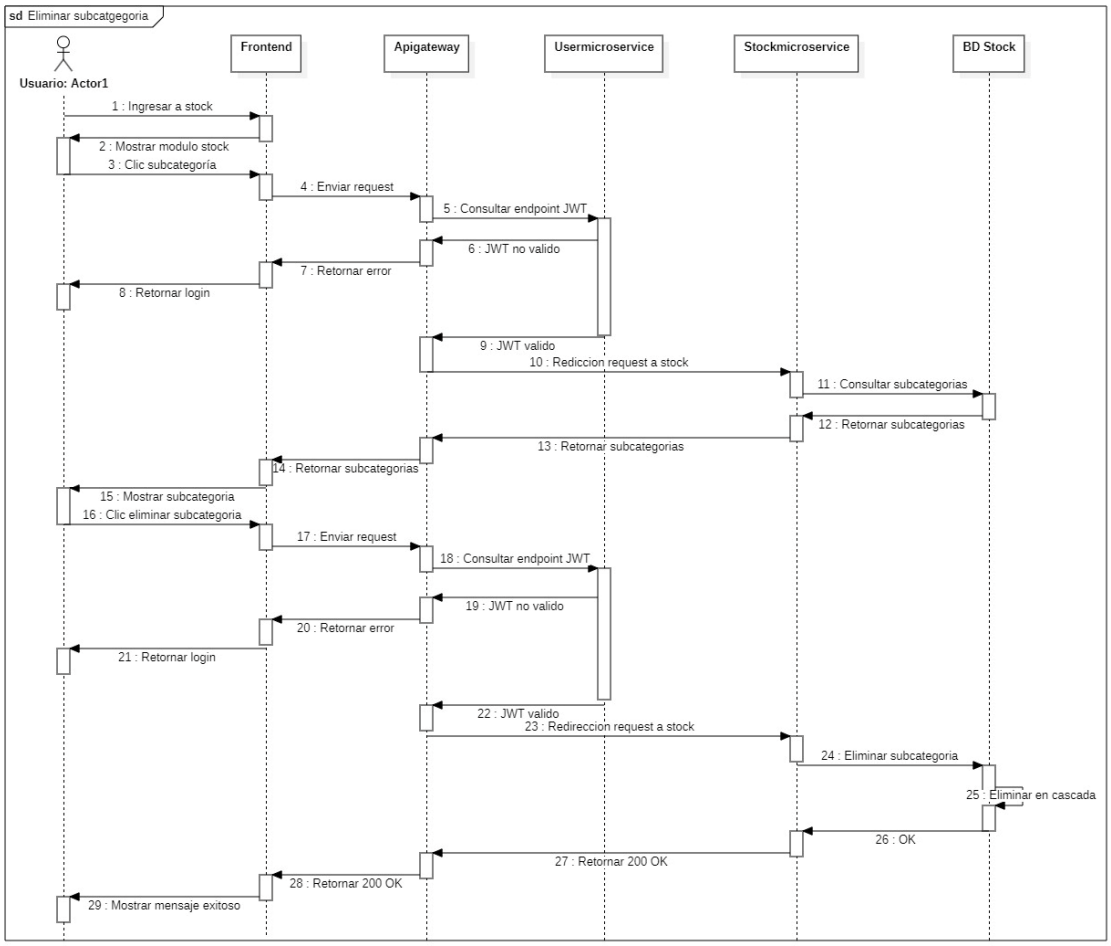


Imagen 23. *Diagrama de secuencia Eliminar subcategoría*

- **Obtener subcategorías**

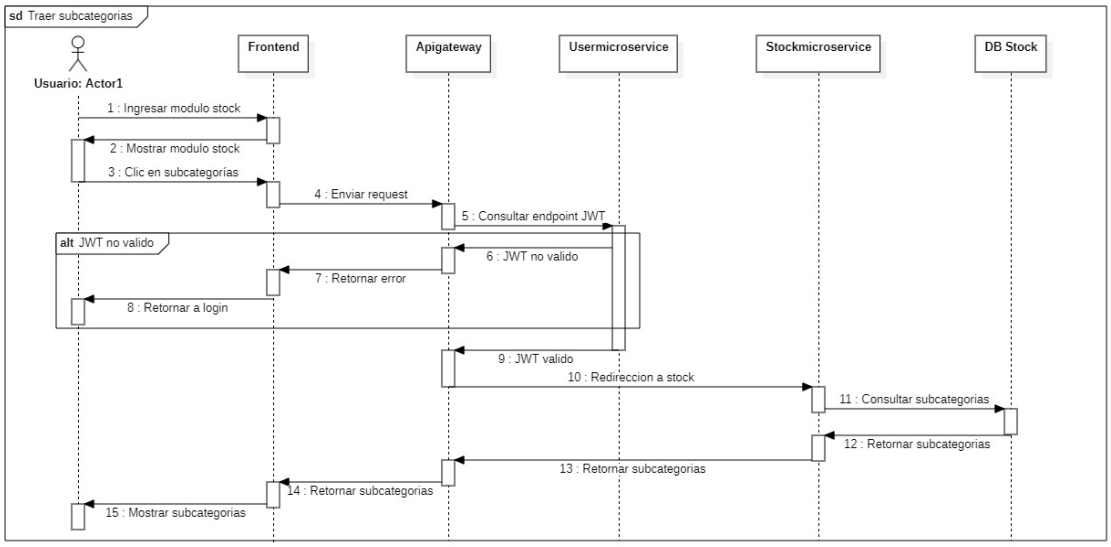


Imagen 24. *Diagrama de secuencia Obtener subcategorías*

11.6.4. Secuencias modulo transacción

El comportamiento de este módulo es vital en la aplicación web, puesto que va a actuar como orquestador entre los diferentes microservicios, esto se debe a que será quien dará la orden para registrar los pagos, deudas, aumentar o disminuir el stock.

- **Crear transacción de compra**

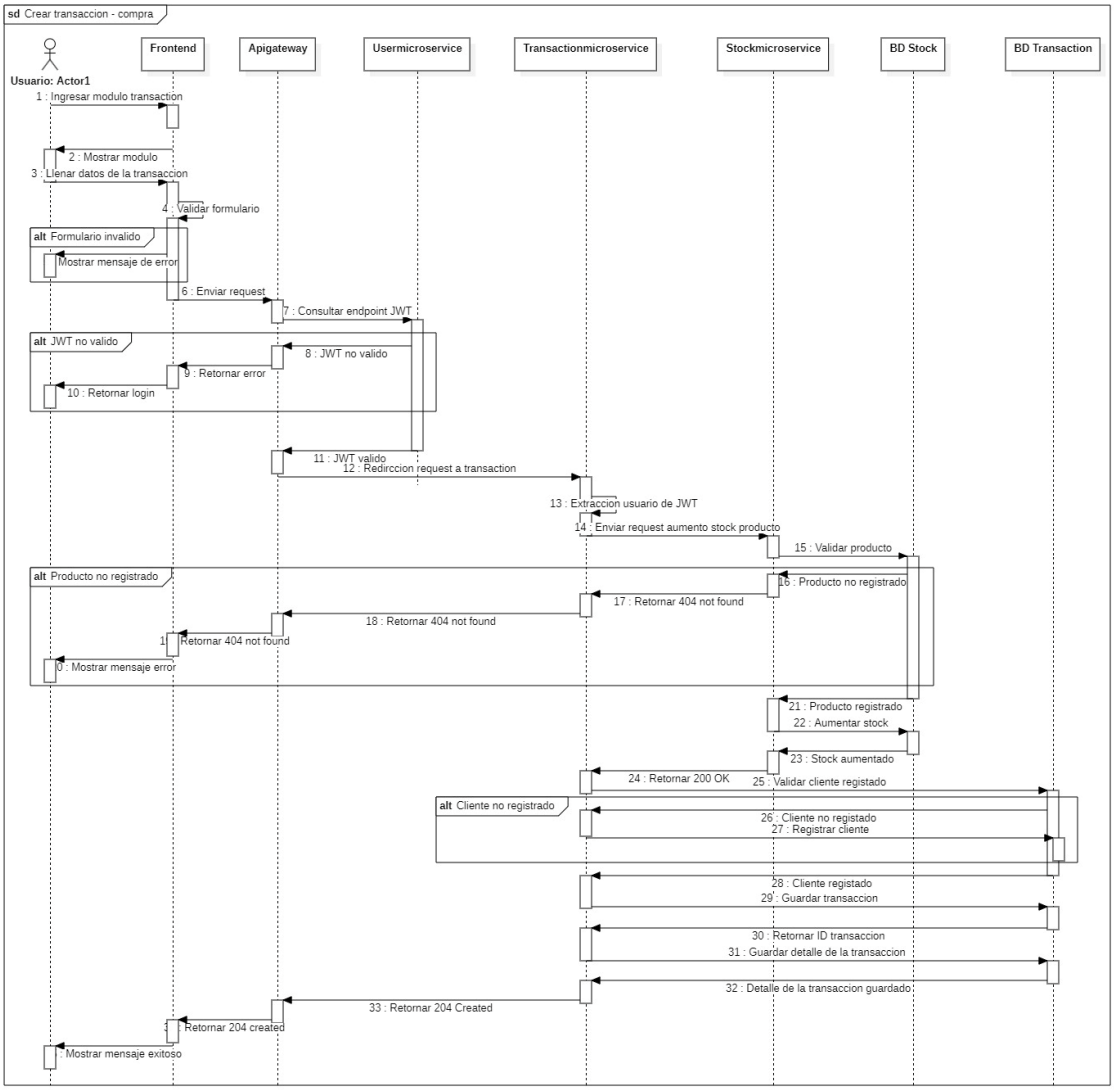


Imagen 25. *Diagrama de secuencia Crea Transacción de Compra*

- **Crear transacción de venta**

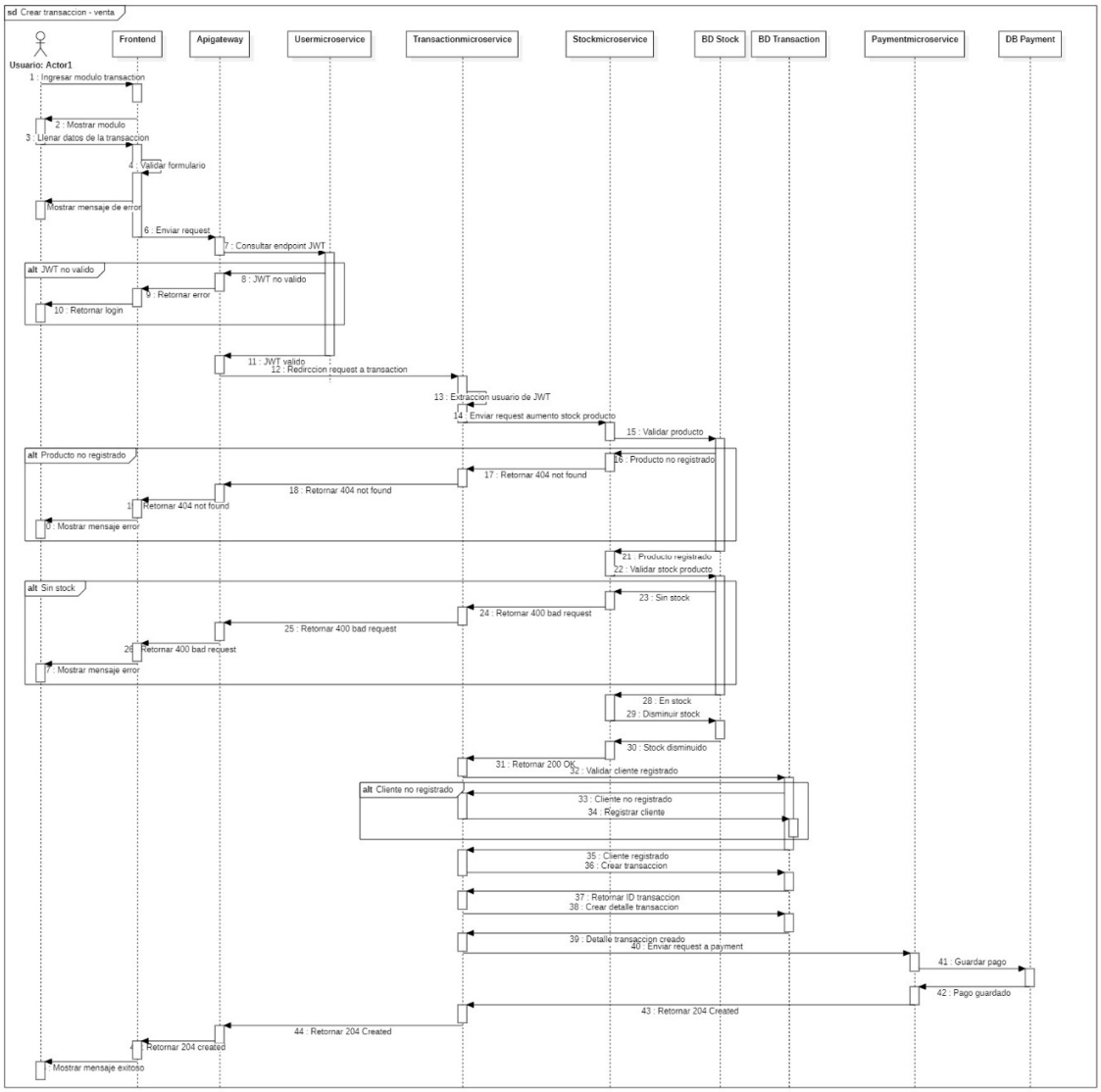


Imagen 26. *Diagrama de secuencia Crear Transacción de Venta*

- **Crear transacción de crédito (venta fiada)**

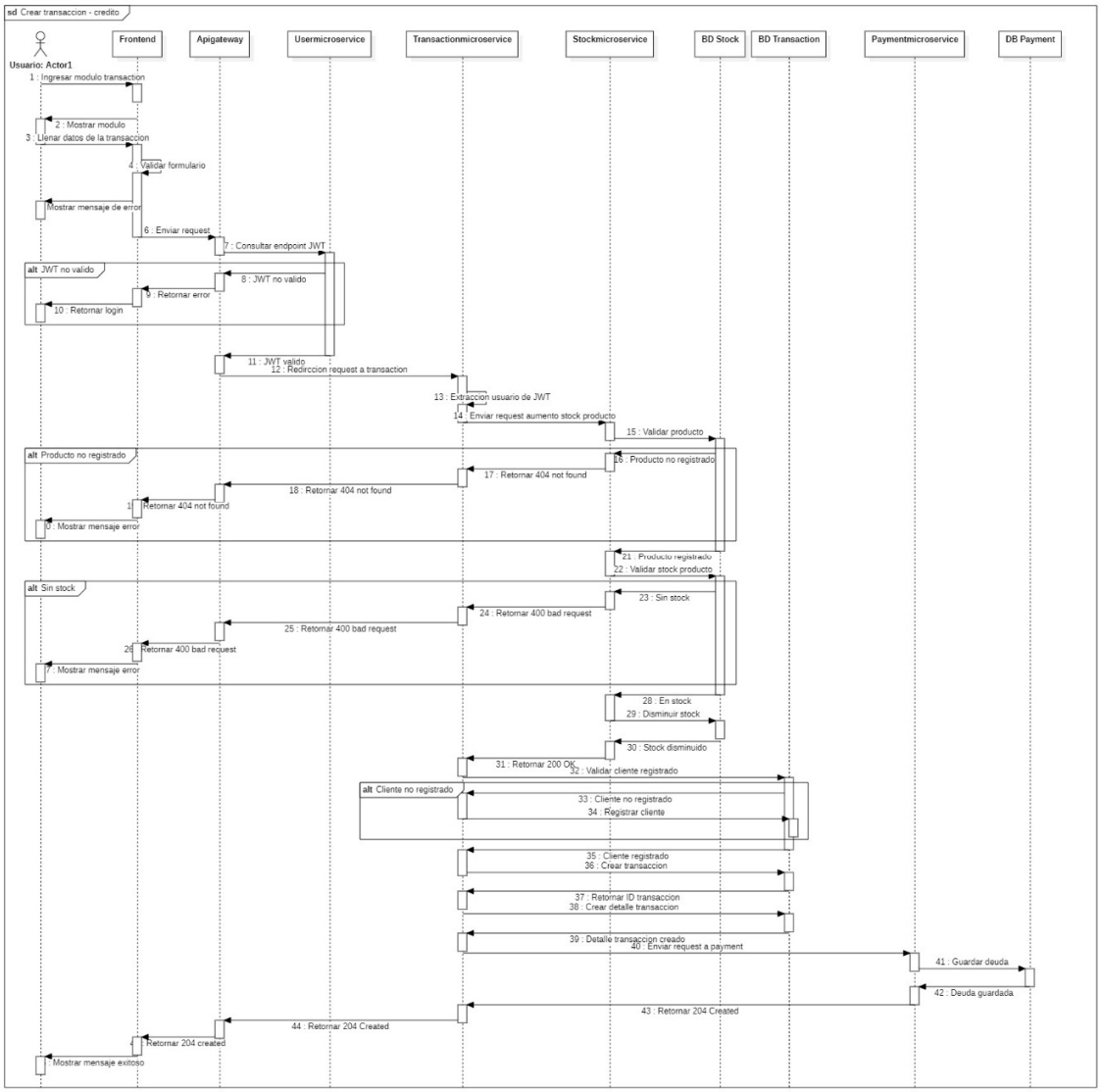


Imagen 27. *Diagrama de secuencia de Crear Transacción de Crédito*

- **Traer detalle de transacción**

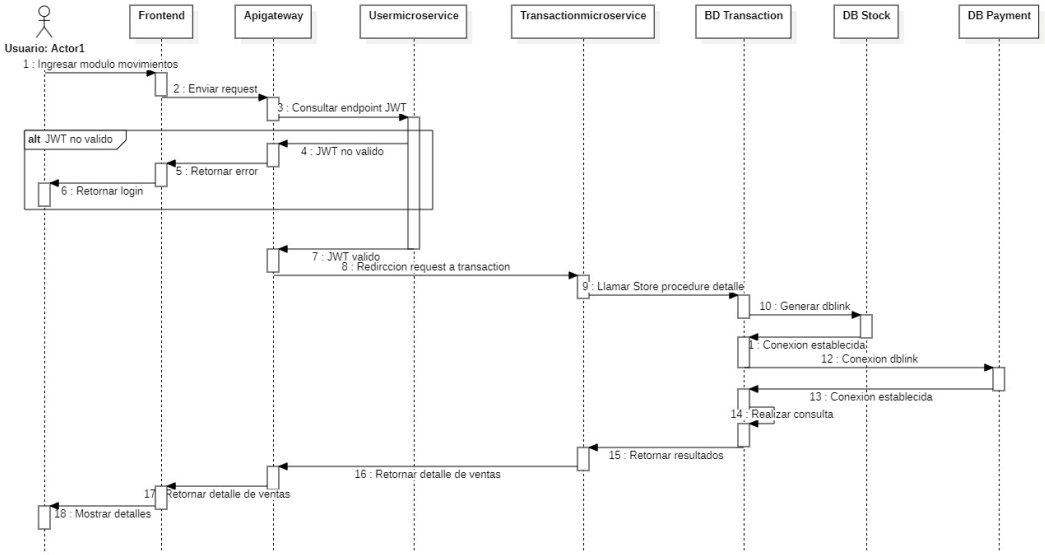


Imagen 28. *Diagrama de secuencia de Traer detalle de la transacción*

- **Traer transacciones**

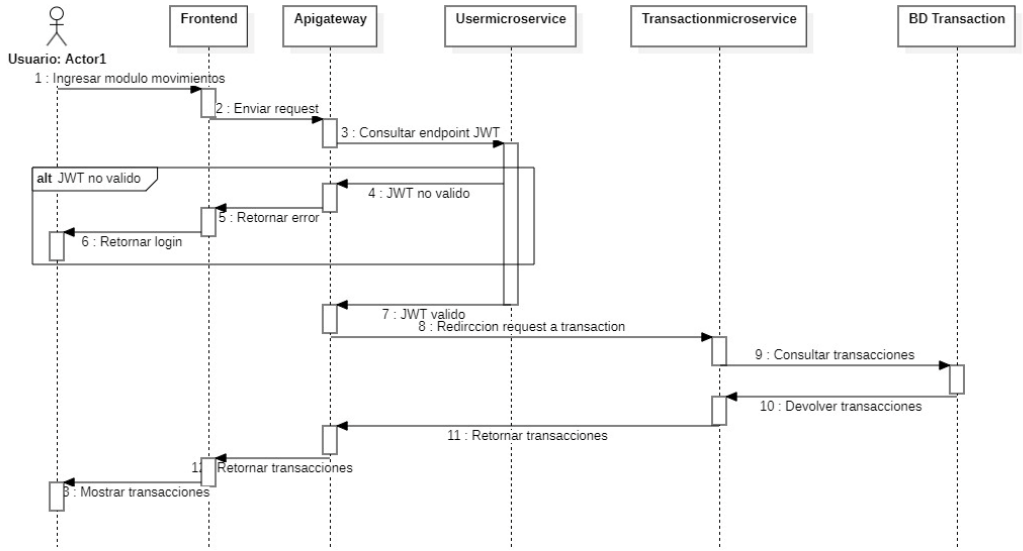


Imagen 29. *Diagrama de secuencia de Traer transacciones*

- **Eliminar transacciones**

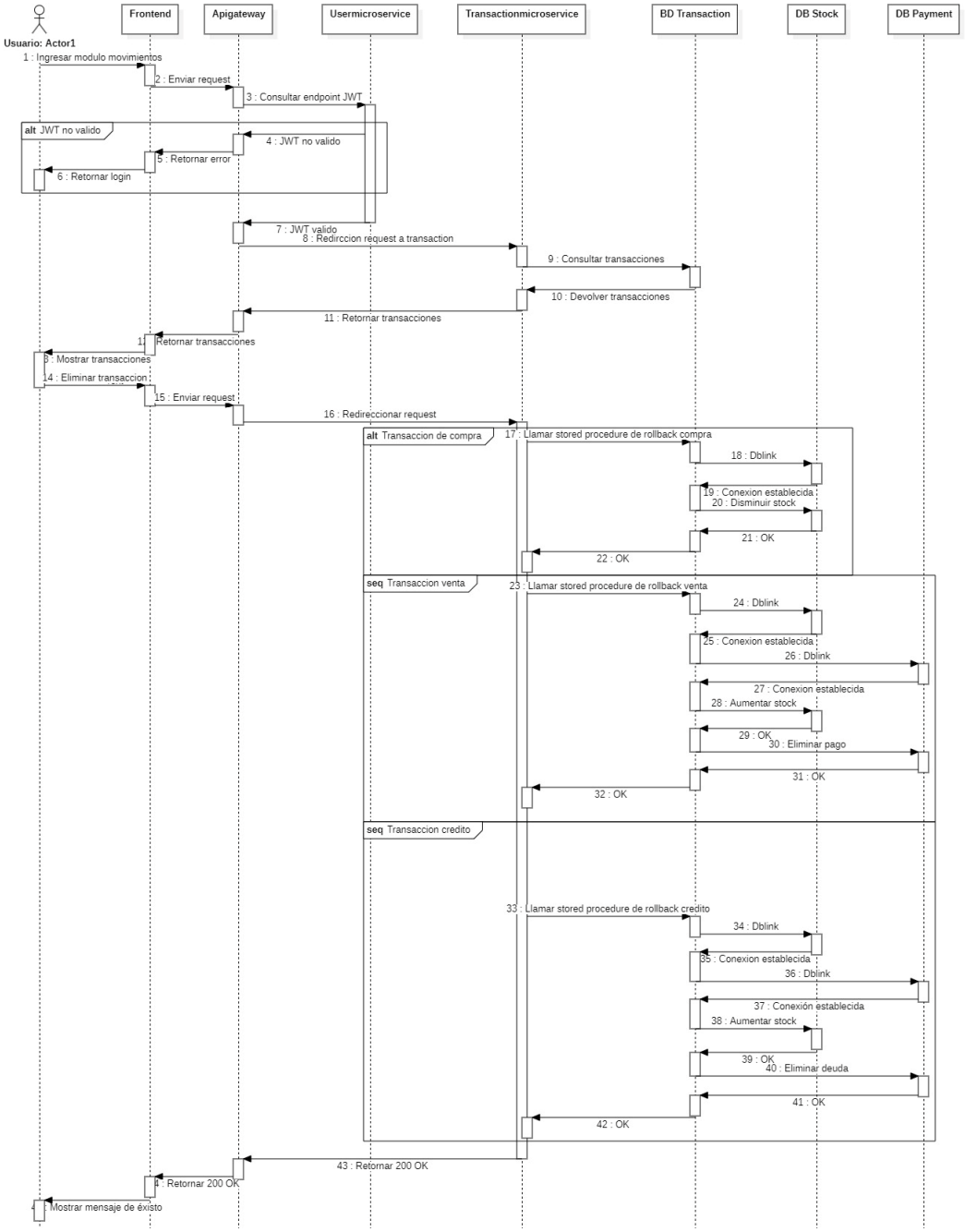


Imagen 30. *Diagrama de secuencia de Eliminar transacciones*

11.6.5. Secuencias de deudores

Este apartado contiene los deudores activos, solamente de lectura y abono a la deuda, si el usuario debe eliminar una deuda, se deberá eliminar la transacción y crear una nueva, puesto que va altamente ligado la transacción realizada con la deuda creada, de igual forma si desea actualizarla.

- **Obtener deudores activos**

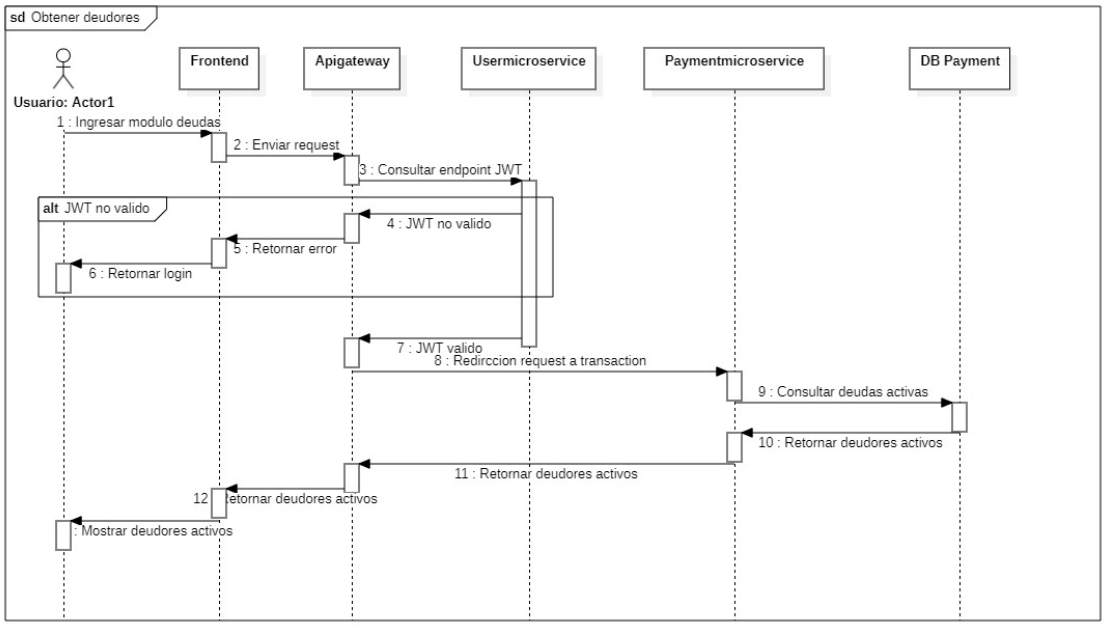


Imagen 31. *Diagrama de secuencia de Obtener deudores*

- **Abonar a la deuda**

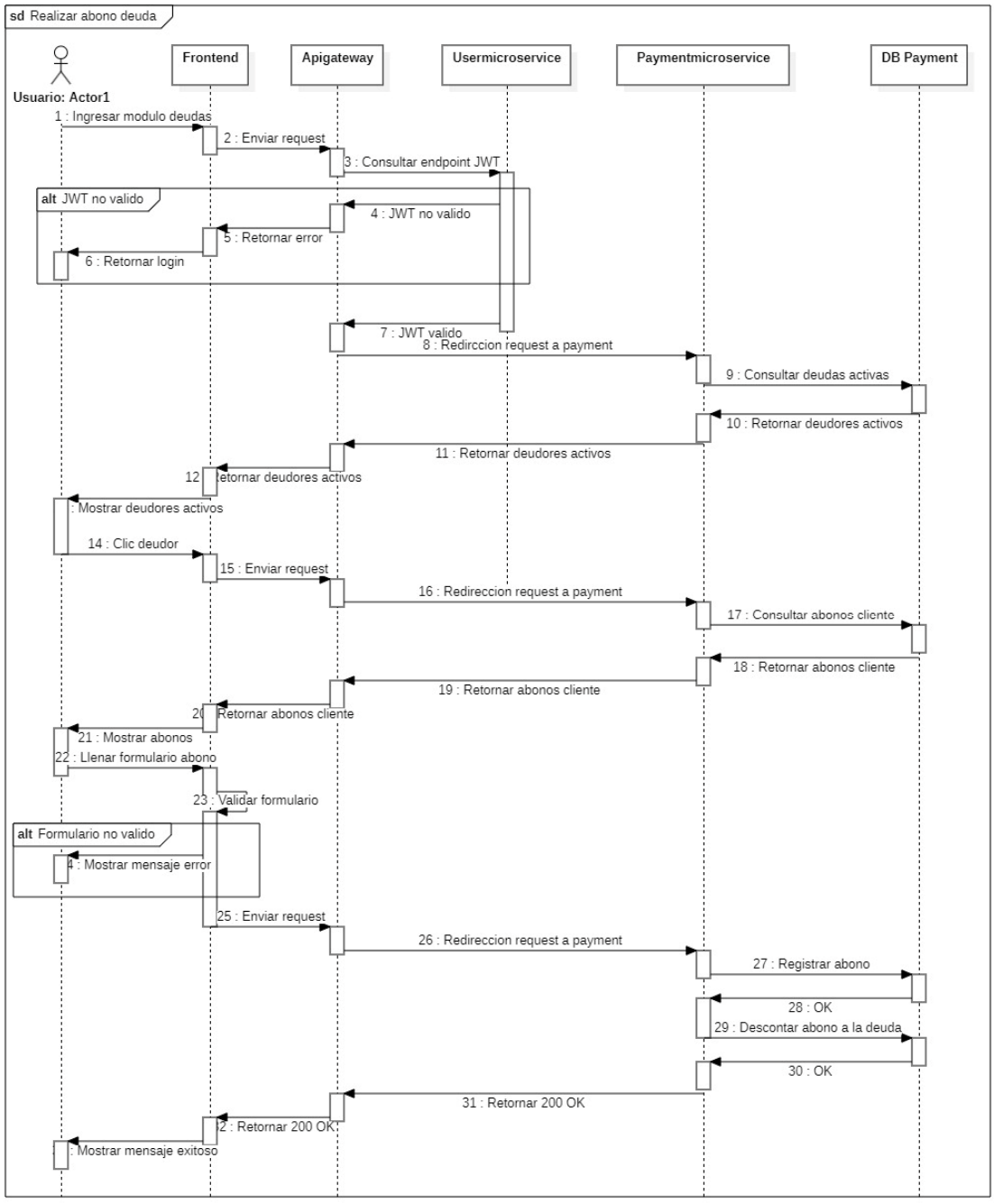


Imagen 32. *Diagrama de secuencia de abonar a la deuda*

11.7. Diagrama de Arquitectura de Software

Este diagrama es fundamental para la descripción visual de la estructura y organización de los microservicios del backend y su conexión con el frontend; ofreciendo una visión clara de la distribución, integración y el flujo de datos que se maneja.

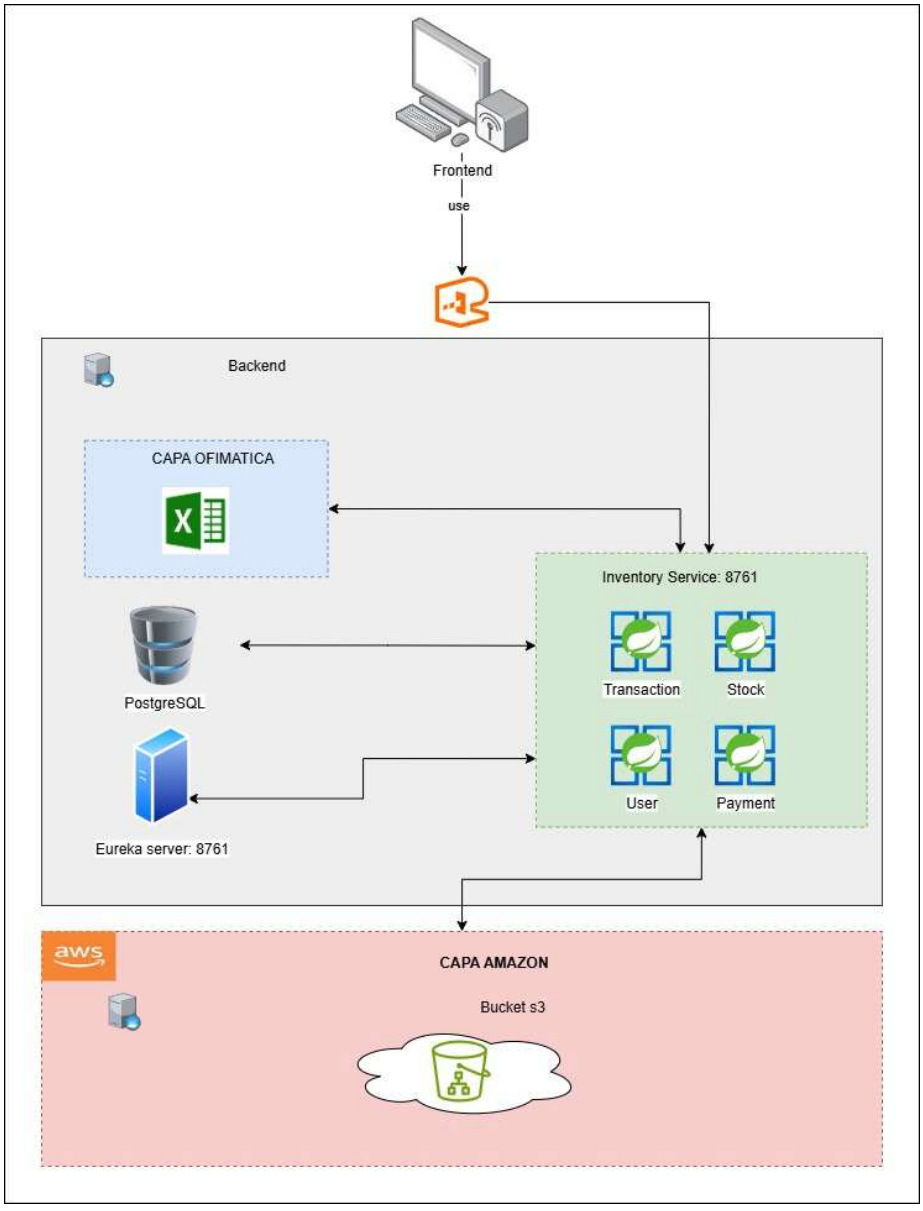


Imagen 33. *Diagrama de arquitectura de Software*

11.8. Diagrama de Componentes

Este diagrama muestra las relaciones entre los diferentes componentes del backend y su conexión con el frontend, así como su comunicación entre los diferentes módulos, cada uno conectado a su propio sistema de persistencia para asegurar la integración de datos.

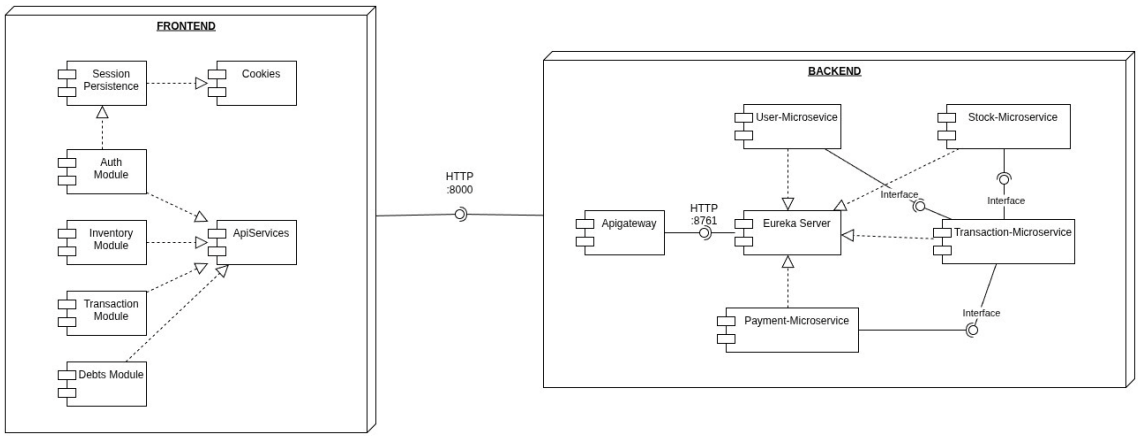


Imagen 34. *Diagrama de Componentes*

Como se evidencia en la anterior imagen, el software propuesto es bastante robusto y orientado a microservicios, integrando un “ApiGateway” para centrar las peticiones, que posteriormente se enrutaran a los respectivos microservicios, adicionalmente, se implementa un “Eureka Server” el cual actúa principalmente como directorio telefónico para el ApiGateway, es decir, que el ApiGateway le preguntará la ubicación o dirección de cada microservicio antes de enviar la solicitud. De esta forma, el Eureka Server permite la comunicación entre diferentes componentes o microservicios sin depender de direcciones IP fijas.

Cada microservicio al iniciar se registra en Eureka, informando su nombre, dirección y puerto, pero si un servicio se deshabilitara temporalmente, el Eureka actualizará su registro y evitará que el ApiGateway o los demás microservicios se intenten comunicar con él.

12. DESARROLLO

12.1. Backlog

ID	Historia de Usuario	Prioridad	Sprint	Responsable
01	Crear categoría	Alta	1	Erick Chaparro
02	Editar categoría	Media	1	
03	Eliminar categoría	Alta	1	
04	Traer categorías	Alta	1	
05	Crear subcategorías	Alta	1	
06	Editar subcategoría	Media	1	
07	Eliminar subcategoría	Alta	1	
08	Traer subcategoría	Alta	2	
09	Crear producto	Alta	2	
10	Editar producto	Media	2	
11	Eliminar producto	Alta	2	
12	Traer productos	Alta	2	
13	Traer productos por nombre	Alta	2	
14	Implementación de JWKS	Alta	3	
15	Inicio de sesión	Alta	3	
16	Registro de usuario	Alta	3	
17	Crear temporizador para revisión de solicitudes de registro	Alta	3	
18	Crear pago proveniente de transacción de venta	Alta	4	
19	Crear deuda proveniente de transacción de crédito	Alta	4	
20	Abonar a deuda	Alta	4	
21	Crear cliente	Alta	5	
22	Crear transacción de compra	Alta	5	
23	Crear transacción de venta	Alta	5	
24	Traer transacciones	Alta	5	
25	Traer detalle de la transacción	Alta	6	
26	Generar reporte Excel de transacciones y deudas	Medio	6	

27	Implementación de Eureka Server	Alta	6	
28	Implementación de un apigateway	Alta	7	
29	Diseñar la página de inventario	Alta	1	Fredy Angarita
30	Diseñar la página de deudores	Alta	1	
31	Diseñar la página de historial	Alta	1	
32	Diseñar la página de transacciones	Alta	1	
33	Implementar versión de inglés y español	Alta	1	
34	Obtener lista de productos	Media	2	
35	Implementar formulario de crear producto	Media	2	
36	Implementar formulario de crear categorías	Media	2	
37	Implementar formulario de subcategorías	Media	2	
38	Implementar formulario de editar subcategorías	Media	3	
39	Implementar el formulario de editar categorías.	Media	3	
40	Implementar la eliminación de subcategorías.	Media	3	
41	Implementar la eliminación de subcategorías.	Media	3	
42	Implementar el formulario de editar productos.	Media	3	
43	Implementar la eliminación de productos.	Media	4	
44	Implementar traducciones en los formularios.	Alta	4	

45	Obtener lista de deudores.	Media	4	
46	Implementar formulario de crear cliente.	Media	4	
47	Implementar formulario de actualizar cliente.	Media	5	
48	Implementar funcionalidades de pago en la página de deudores.	Media	5	
49	Diseñar la página de registro e inicio de sesión.	Alta	5	
50	Implementar la lógica de autenticación.	Alta	5	
51	Implementar panel de administrador.	Media	6	
52	Crear transacción	Media	7	
53	Implementar funcionalidades en la página de historial.	Media	7	

Tabla 7. *Backlog*

12.2. Análisis

12.2.1. Historias de usuario backend

Historia de usuario	
Número: 01	**Usuario:** Administrador
Nombre de la historia: Crear categoría	**Iteración asignada:** 1
Prioridad: Alta	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador necesito crear una categoría para relacionar y encontrar mis productos con mayor facilidad.	
Criterios de aceptación: • Cada categoría tiene 3 campos: ID, nombre y descripción. • El nombre de la categoría no se puede repetir. • Todas las categorías deben tener nombre y descripción. • El tamaño máximo del nombre debe ser de 50 caracteres. • El tamaño máximo de la descripción debe ser de 150 caracteres.	

Tabla 8. *Historia de Usuario 01*

Historia de usuario	
Número: 02	**Usuario:** Administrador
Nombre de la historia: Editar categoría	**Iteración asignada:** 1
Prioridad: Media	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador editar una categoría en caso de haber cometido un error.	
Criterios de aceptación: • Deberá respetar los caracteres máximos • No se deberán eliminar las subcategorías y productos asociados • Podrá editarse tanto nombre como descripción • No se permiten campos vacíos	

Tabla 9. *Historia de Usuario 02*

Historia de usuario	
Número: 03	**Usuario:** Administrador
Nombre de la historia: Eliminar categoría	**Iteración asignada:** 1
Prioridad: Media	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador quiero eliminar una categoría.	
Criterios de aceptación: • Deberá eliminarse las subcategorías y productos relacionados. • En caso de haber un error devolver el error.	

Tabla 10. *Historia de usuario 3*

Historia de usuario	
Número: 04	**Usuario:** Administrador
Nombre de la historia: Traer categoría	**Iteración asignada:** 1
Prioridad: Alta	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador quiero un servicio en el que me traiga las categorías guardadas.	
Criterios de aceptación: • El servicio debe estar paginado • Debe recibir tamaño y numero de página • Debe permitir devolver los resultados de forma ascendente o descendente dependiendo del nombre de la categoría	

Tabla 11. *Historia de usuario 4*

Historia de usuario

Número: 05	**Usuario:** Administrador
Nombre de la historia: Crear subcategoría	**Iteración asignada:** 1
Prioridad: Alta	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador quiero un servicio en el que pueda crear una subcategoría para relacionar de mejor manera los productos	
Criterios de aceptación: • Cada subcategoría tiene 4 campos: ID, nombre, descripción y el ID de la categoría a la que va relacionada • El nombre de la categoría no se puede repetir • No se permiten campos vacíos • La longitud máxima del nombre debe ser de 50 caracteres • La longitud máxima de la descripción debe ser de 150 caracteres • Se debe verificar si la categoría a la que va a estar relacionada existe	

Tabla 12. *Historia de usuario 5*

Historia de usuario	
Número: 06	**Usuario:** Administrador
Nombre de la historia: Editar subcategoría	**Iteración asignada:** 1
Prioridad: Media	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador quiero un servicio en el que pueda editar una subcategoría en caso de haber cometido un error	
Criterios de aceptación: • Cada subcategoría tiene 4 campos: ID, nombre, descripción y el ID de la categoría a la que va relacionada • El nombre de la categoría no se puede repetir • No se permiten campos vacíos • La longitud máxima del nombre debe ser de 50 caracteres • La longitud máxima de la descripción debe ser de 150 caracteres • Se debe verificar si la categoría a la que va a estar relacionada existe	

Tabla 13. *Historia de usuario 6*

Historia de usuario	
Número: 07	**Usuario:** Administrador
Nombre de la historia: Eliminar subcategoría	**Iteración asignada:** 1
Prioridad: Alta	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador quiero un servicio para eliminar una subcategoría	
Criterios de aceptación:	

• Al eliminar, se deben eliminar los productos relacionados a esa categoría • En caso de no existir la subcategoría se debe devolver un error

Tabla 14. *Historia de usuario 7*

Historia de usuario	
Número: 08	**Usuario:** Administrador
Nombre de la historia: Traer subcategoría	**Iteración asignada:** 1
Prioridad: Alta	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador quiero un servicio para obtener todas las subcategorías	
Criterios de aceptación: • El servicio debe estar paginado • Parametrizar si se desea las subcategorías en orden alfabético • Parametrizar si se desea las subcategorías en orden alfabético según la categoría relacionada • Parametrizar campo para obtener subcategorías según la categoría enviada	

Tabla 15. *Historia de usuario 8*

Historia de usuario	
Número: 09	**Usuario:** Administrador
Nombre de la historia: Crear producto	**Iteración asignada:** 2
Prioridad: Alta	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador quiero un servicio para crear un producto	
Criterios de aceptación: • Cada producto contiene 8 campos: ID, nombre, descripción, imagen, stock, precio de venta, precio de compra y la subcategoría a la que va a estar asociada • El nombre del producto no debe superar los 100 caracteres • La descripción no debe superar los 200 caracteres • El precio de venta no debe ser menor al precio de compra • Debe hacer uso del servicio de AWS BucketS3 para el almacenaje de imágenes	

Tabla 16. *Historia de usuario 9*

Historia de usuario

Número: 10	**Usuario:** Administrador
Nombre de la historia: Editar producto	**Iteración asignada:** 2
Prioridad: Media	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador quiero un servicio para editar un producto en caso de haber cometido algún error a la hora de la creación	
Criterios de aceptación: • Se debe poder editar en cualquiera de sus campos • Realizar validaciones correspondientes a la creación	

Tabla 17. *Historia de usuario 10*

Historia de usuario	
Número: 11	**Usuario:** Administrador
Nombre de la historia: Eliminar producto	**Iteración asignada:** 2
Prioridad: Alta	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador quiero un servicio para eliminar un producto	
Criterios de aceptación: • Se debe eliminar la imagen del servicio de AWS BucketS3	

Tabla 18. *Historia de usuario 11*

Historia de usuario	
Número: 12	**Usuario:** Administrador
Nombre de la historia: Traer productos	**Iteración asignada:** 2
Prioridad: Alta	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador quiero un servicio para obtener productos	
Criterios de aceptación: • Este servicio debe traer todos los productos sin importar la categoría o subcategoría • Debe retornar todos los datos del producto, adicional la subcategoría y categoría a la que pertenece • El servicio debe estar paginado	

Tabla 19. *Historia de usuario 12*

Historia de usuario	
Número: 13	**Usuario:** Administrador
Nombre de la historia: Traer productos por nombre	**Iteración asignada:** 2

Prioridad: Alta	**Riesgo de desarrollo:** Bajo
Descripción: Yo como administrador quiero un servicio para obtener productos por coincidencia de nombre	
Criterios de aceptación: • Este servicio debe traer todos los productos sin importar la categoría o subcategoría • Debe estar paginado • Debe retornar todos los datos del producto, adicional la subcategoría y categoría a la que pertenece • Deberá retornar los productos que coincidan con el nombre digitado	

Tabla 20. *Historia de usuario 13*

Historia de usuario	
Número: 14	**Usuario:** Administrador
Nombre de la historia: Implementación de JWKS	**Iteración asignada: 3**
Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Yo como administrador quiero un servicio que implemente JWKS para la autentificación y autorización mediante JWT, esto con el fin de que sea punto de entrada y verificación para los demás microservicios.	
Criterios de aceptación: • No debe afectarse las otras funcionalidades • Debe validar autorización y autentificación • Debe aplicar la estrategia de Oauth2.0 • Debe exponer un endpoint para que el apigateway lo consulte con la finalidad de verificar la autenticidad del token generado o enviado	

Tabla 21. *Historia de usuario 15*

Historia de usuario	
Número: 15	**Usuario:** Administrador
Nombre de la historia: Inicio de sesión	**Iteración asignada: 3**
Prioridad: Alta	**Riesgo de desarrollo:** Media
Descripción: Yo como administrador quiero un servicio que implemente el inicio de sesión de los usuarios registrados	
Criterios de aceptación: • Debe retornar el JWT • Debe verificar los datos enviados • Se debe retornar el rol del usuario	

Tabla 22. *Historia de usuario 16*

Historia de usuario	
Número: 16	**Usuario:** Administrador
Nombre de la historia: Registro de usuario	**Iteración asignada: 3**
Prioridad: Alta	**Riesgo de desarrollo:** Alto
Descripción: Yo como administrador quiero tener un servicio para que un futuro colaborador pueda registrarse, y yo poder aprobar el registro, esto con el fin de añadir una capa de seguridad y no cualquier persona se pueda registrar e ingresar al aplicativo web.	
Criterios de aceptación: • El formulario de registro deberá contener los siguientes campos: primer nombre, primer apellido, correo, contraseña y fecha de nacimiento • El usuario que desee registrarse no puede ser menor de edad • Las solicitudes iniciales deben tener un estado de "PENDIENTE" • Las solicitudes aprobadas deben estar ligadas al administrador para su posterior aprobación o rechazo de la solicitud • La solicitud aprobada quedará con estado "APROBADA" • La solicitud rechazada quedará con estado "RECHAZADA" • Al crear una solicitud de registro, este quedará con la fecha en la que se realizó la solicitud • Si la solicitud es aprobada el usuario solicitante quedará registrado en la base de datos con un rol de "USUARIO" y no podrá acceder al panel de aprobación • Todas las solicitudes una vez eliminadas o se cambien de estado, deberán quedar en un registro en el que se pueda consultar el historial de solicitudes con su último estado	

Tabla 23. *Historia de usuario 17*

Historia de usuario	
Número: 17	**Usuario:** Administrador
Nombre de la historia: Crear temporizador para revisión de solicitudes de registro	**Iteración asignada: 3**
Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Yo como administrador quiero que haya un temporizador en el que haga revisión de las solicitudes de registro.	
Criterios de aceptación: • Las solicitudes "ACEPTADA" pasarán a la tabla usuarios con sus credenciales ingresadas y deberán eliminarse y pasar a un historial	

• Las solicitudes “RECHAZDA” se deberán eliminar y pasarán a un historial • Las solicitudes “PENDIENTE” viejas, deberán eliminarse y pasar a un historial • El temporizador deberá ejecutarse cada 5 minutos

Tabla 24. *Historia de usuario 17*

Historia de usuario	
Número: 18	**Usuario:** Administrador
Nombre de la historia: Crear pago proveniente de transacción de venta	**Iteración asignada: 4**
Prioridad: Alta	**Riesgo de desarrollo:** Baja
Descripción: Yo como administrador quiero que, al registrar una transacción de venta, se genere un registro del pago realizado.	
Criterios de aceptación: • Debe exponer un endpoint para que el microservicio de transacción lo pueda consumir y generar el registro del pago • Deberá guardar los siguientes datos: ID, monto pagado, fecha de pago, cliente, transacción a la que va relacionada	

Tabla 25. *Historia de usuario 18*

Historia de usuario	
Número: 19	**Usuario:** Administrador
Nombre de la historia: Crear deuda proveniente de transacción de crédito	**Iteración asignada: 4**
Prioridad: Alta	**Riesgo de desarrollo:** Media
Descripción: Yo como administrador quiero que se registre la deuda proveniente de la transacción de crédito para su posterior monitoreo de la deuda	
Criterios de aceptación: • Los datos para registrar son: ID, monto total de la deuda, monto faltante de la deuda, estado, fecha de creación de la deuda, última fecha de abono a la deuda, cliente que tiene la deuda, transacción de la que proviene • Inicialmente, al crear la deuda, el monto faltante será el mismo del monto total de la deuda • Su estado inicial será PENDIENTE	

• Deberá exponer un endpoint para que el microservicio de transacciones lo pueda consumir al generar la transacción de crédito

Tabla 26. *Historia de usuario 19*

Historia de usuario	
Número: 20	**Usuario:** Administrador
Nombre de la historia: Abonar a deuda	**Iteración asignada: 4**
Prioridad: Alta	**Riesgo de desarrollo:** Media
Descripción: Yo como administrador quiero un servicio en el que pueda registrar lo abonado por el cliente a la deuda activa	
Criterios de aceptación: • El monto abonado no debe superar el monto faltante • Se debe registrar los abonos de la deuda en los pagos realizados para mantener un historial • El monto abonado se le restará al monto faltante • Una vez llegado el monto faltante a 0, el estado de la deuda deberá cambiar a "PAGADO"	

Tabla 27. *Historia de usuario 20*

Historia de usuario	
Número: 21	**Usuario:** Administrador
Nombre de la historia: Crear cliente	**Iteración asignada: 5**
Prioridad: Alta	**Riesgo de desarrollo:** Baja
Descripción: Yo como administrador quiero que se pueda registrar un cliente, esto con el fin de que, al realizar una transacción, sea posible su asociación al cliente en cuestión	
Criterios de aceptación: • Los datos para registrar del cliente son los siguientes: nombre y teléfono • Si no se envía el indicativo del cliente, deberá asignar el indicativo correspondiente al número de teléfono de manera automática • El número de teléfono debe ser único	

Tabla 28. *Historia de usuario 21*

Historia de usuario	
Número: 22	**Usuario:** Administrador
Nombre de la historia: Crear transacción de compra	**Iteración asignada: 5**

Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Yo como administrador quiero registrar una transacción de compra, esto con el fin de dejar registro los productos a los que se les aumenta el stock	
Criterios de aceptación: • Para registrar una transacción de compra se requieren los siguientes datos: productos y cantidad, método de pago, tipo de movimiento, cliente y fecha en la que se realizó la transacción • La compra debe estar asociada a un cliente genérico, pues no es relevante para el control de los clientes • Se debe crear el detalle de la operación de manera automática • Para registrar el detalle de la transacción son: precio por unidad del producto, cantidad, transacción a la que va asociada y producto al que va asociada la transacción. • Se debe consultar el precio de compra del producto en cuestión y registrarlo en el detalle de la transacción • Si se actualiza el precio de compra del producto posterior a la transacción, no se debe modificar el precio unitario del producto con el que quedó el detalle de la transacción • Al crear la transacción de compra, se debe aumentar el stock del producto en cuestión lo que la cantidad indique	

Tabla 29. *Historia de usuario 22*

Historia de usuario	
Número: 23	**Usuario:** Administrador
Nombre de la historia: Crear transacción de venta	**Iteración asignada: 5**
Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Yo como administrador quiero registrar una transacción de venta, esto con el fin de registrar lo vendido al cliente	
Criterios de aceptación: • Para registrar una transacción de venta se requieren los siguientes datos: productos y cantidad, método de pago, tipo de movimiento, cliente y fecha en la que se realizó la transacción • La venta puede o no está asociada a un cliente ya registrado • En caso de que no se registre cliente, se asignará la transacción a un cliente genérico • El detalle de la transacción se deberá creará automáticamente • Para registrar el detalle de la transacción son: precio por unidad del producto, cantidad, transacción a la que va asociada y producto al que va asociada la transacción.	

• Para el detalle de la transacción se debe almacenar el precio unitario de venta del producto en cuestión • Se debe disminuir el stock del producto en cuestión lo que indique la cantidad digitada • Se deberá registrar el pago de manera automática

Tabla 30. *Historia de usuario 23*

Historia de usuario	
Número: 24	**Usuario:** Administrador
Nombre de la historia: Traer transacciones	**Iteración asignada: 5**
Prioridad: Alta	**Riesgo de desarrollo:** Baja
Descripción: Yo como administrador quiero obtener las transacciones realizadas.	
Criterios de aceptación: • El servicio debe estar paginado • Debe filtrar por fecha y tipo de transacción • En caso de no diligenciar la fecha, deberá hacer una búsqueda total • En caso de no diligenciar el tipo de transacción, se deberá realizar una búsqueda de los 3 tipos de transacción (compra, venta y crédito) • Se debe retornar la información de la transacción	

Tabla 31. *Historia de usuario 24*

Historia de usuario	
Número: 25	**Usuario:** Administrador
Nombre de la historia: Traer detalle de la transacción	**Iteración asignada: 6**
Prioridad: Alta	**Riesgo de desarrollo:** Baja
Descripción: Yo como administrador quiero obtener el detalle de la transacción en cuestión.	
Criterios de aceptación: • Se debe retornar la información del detalle de la transacción en cuestión	

Tabla 32. *Historia de usuario 25*

Historia de usuario	
Número: 26	**Usuario:** Administrador
Nombre de la historia: Generar reporte Excel de transacciones y deudas	**Iteración asignada: 6**
Prioridad: Medio	**Riesgo de desarrollo:** Baja
Descripción: Yo como administrador quiero obtener un reporte en Excel (xlsx) en el que me muestre la información total referente de las transacciones y deudas en un periodo de tiempo	
Criterios de aceptación: • Debe realizar búsqueda por fecha inicial y fecha final • El nombre del reporte debe tener el siguiente formato: report_DDMMYYY-DDMMYYY que corresponden a la fecha inicial y fecha final del reporte • El reporte constará de 2 hojas, la primera de todo el detalle de las transacciones y la segunda sobre las deudas tanto activas como ya pagadas y su estado correspondiente	

Tabla 33. *Historia de usuario 26*

Historia de usuario	
Número: 27	**Usuario:** Administrador
Nombre de la historia: Implementación de Eureka Server	**Iteración asignada: 6**
Prioridad: Alta	**Riesgo de desarrollo:** Baja
Descripción: Yo como administrador quiero la implementación de un Eureka Server, esto con el fin de que funcione como un directorio telefónico para un futuro apigateway, adicional de proveer información sobre el estado de los microservicios	
Criterios de aceptación: • Las funcionalidades previas no deben verse afectadas • Cada microservicio debe estar suscrito al Eureka Server	

Tabla 34. *Historia de usuario 27*

Historia de usuario	
Número: 28	**Usuario:** Administrador
Nombre de la historia: Implementación de un apigateway	**Iteración asignada: 7**
Prioridad: Alta	**Riesgo de desarrollo:** Alta

Descripción: Yo como administrador quiero que todos los microservicios queden centralizados mediante un apigateway, esto con el fin de manejar solamente una URL a la hora de consumir los servicios
Criterios de aceptación: • Las funcionalidades previas no deben verse afectadas • El apigateway deberá consultar el endpoint de JWKS del microservicio de usuario con el fin de validar primero el JWT antes de redireccionar el request al microservicio correspondiente • El apigateway deberá usar el Eureka Server para enrutar el request enviado

Tabla 35. *Historia de usuario 28*

12.2.2. Historias de usuario frontend

Historia de usuario	
Número: 29	**Usuario:** Todos
Nombre de la historia: Diseñar la página de inventario.	**Iteración asignada: 1**
Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Crear la interfaz de usuario correspondiente al inventario, donde el usuario interactuara con toda la información de productos.	
Criterios de aceptación: • Usar de referencia el mockup previsto. • Implementar los componentes necesarios. • NO implementar funcionalidades. • Hacer uso de los componentes necesarios de PrimeVue.	

Tabla 36. *Historia de usuario 29*

Historia de usuario	
Número: 30	**Usuario:** Todos
Nombre de la historia: Diseñar la página de deudores.	**Iteración asignada: 1**
Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Maquetar la interfaz de la página de deudores, en la cual se encontrará toda la información referente a los clientes deudores.	
Criterios de aceptación: • Usar de referencia el mockup previsto. • Implementar los componentes necesarios. • NO implementar funcionalidades. • Hacer uso de los componentes necesarios de PrimeVue.	

Tabla 37. *Historia de usuario 30*

Historia de usuario	
Número: 31	**Usuario:** Todos
Nombre de la historia: Diseñar la página de historial.	**Iteración asignada: 1**
Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Construir la interfaz de la página de historial, donde se interactuará con todo el historial de ventas y compras.	
Criterios de aceptación: • Usar de referencia el mockup previsto. • Implementar los componentes necesarios. • NO implementar funcionalidades. • Hacer uso de los componentes necesarios de PrimeVue.	

Tabla 38. *Historia de usuario 31*

Historia de usuario	
Número: 32	**Usuario:** Todos
Nombre de la historia: Diseñar la página de transacciones.	**Iteración asignada: 1**
Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Crear la interfaz de transacciones, en la cual el usuario podrá hacer el registro de ventas y compras.	
Criterios de aceptación: • Usar de referencia el mockup previsto. • Implementar los componentes necesarios. • NO implementar funcionalidades. • Hacer uso de los componentes necesarios de PrimeVue.	

Tabla 39. *Historia de usuario 32*

Historia de usuario	
Número: 33	**Usuario:** Todos
Nombre de la historia: Implementar versión en inglés y español	**Iteración asignada: 1**
Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Implementar la funcionalidad de cambio de idiomas, enfocado principalmente en español e inglés.	
Criterios de aceptación:	

• Hacer uso de una librería de traducción. • Implementarlo en todas las opciones presentadas al usuario. • Se deberá agregar una opción para cambiar el idioma.

Tabla 40. *Historia de usuario 33*

Historia de usuario	
Número: 34	**Usuario:** Todos
Nombre de la historia: Obtener lista de productos	**Iteración asignada: 2**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Obtener la lista de productos guardados por el usuario.	
Criterios de aceptación: • El usuario debe ver reflejado los datos de los productos registrados al entrar a la página de inventario. • Cada producto mostrado tendrá las opciones de editar y eliminar. • La lista debe estar paginada.	

Tabla 41. *Historia de usuario 34*

Historia de usuario	
Número: 35	**Usuario:** Todos
Nombre de la historia: Implementar formulario de crear producto.	**Iteración asignada: 2**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar el formulario con el cual el usuario podrá crear un producto.	
Criterios de aceptación: • El usuario podrá crear un producto con imagen. • Deberá usar el endpoint específico para la creación del producto. • Deberá verificar las entradas del usuario en los diferentes campos. • Deberá mostrar el resultado de la operación al usuario.	

Tabla 42. *Historia de usuario 35*

Historia de usuario	
Número: 36	**Usuario:** Todos
Nombre de la historia: Implementar formulario de crear categorías.	**Iteración asignada: 2**
Prioridad: Media	**Riesgo de desarrollo:** Media

Descripción: Implementar el formulario con el cual el usuario podrá crear una categoría.
Criterios de aceptación: • Deberá usar el endpoint específico para la creación de categorías. • Deberá verificar las entradas del usuario en los diferentes campos. • Deberá mostrar el resultado de la operación al usuario.

Tabla 43. *Historia de usuario 36*

Historia de usuario	
Número: 37	**Usuario:** Todos
Nombre de la historia: Implementar formulario de crear subcategorías.	**Iteración asignada: 2**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar el formulario con el cual el usuario podrá crear una subcategoría.	
Criterios de aceptación: • Se debe asegurar que la subcategoría está asociada a una categoría. • Deberá usar el endpoint específico para la creación de subcategorías. • Deberá verificar las entradas del usuario en los diferentes campos. • Deberá mostrar el resultado de la operación al usuario.	

Tabla 44. *Historia de usuario 37*

Historia de usuario	
Número: 38	**Usuario:** Todos
Nombre de la historia: Implementar el formulario de editar subcategorías.	**Iteración asignada: 3**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar el formulario para editar una subcategoría.	
Criterios de aceptación: • El usuario podrá editar la información básica de la subcategoría. • Deberá usar el endpoint específico para la edición de la subcategoría. • Deberá verificar las entradas del usuario en los diferentes campos. • Deberá mostrar el resultado de la operación al usuario.	

Tabla 45. *Historia de usuario 38*

Historia de usuario	
Número: 39	**Usuario:** Todos
Nombre de la historia: Implementar el formulario de editar categorías.	**Iteración asignada: 3**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar el formulario para editar una categoría.	
Criterios de aceptación:	

• El usuario podrá editar la información básica de la categoría. • Deberá usar el endpoint específico para la edición de la categoría. • Deberá verificar las entradas del usuario en los diferentes campos. • Deberá mostrar el resultado de la operación al usuario.

Tabla 46. *Historia de usuario 39*

Historia de usuario	
Número: 40	**Usuario:** Todos
Nombre de la historia: Implementar la eliminación de subcategorías.	**Iteración asignada: 3**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar la funcionalidad de eliminar una subcategoría.	
Criterios de aceptación: • Deberá usar el endpoint específico para eliminar subcategorías. • El usuario deberá confirmar la eliminación de la subcategoría. • Deberá mostrar el resultado de la operación al usuario.	

Tabla 47. *Historia de usuario 40*

Historia de usuario	
Número: 41	**Usuario:** Todos
Nombre de la historia: Implementar la eliminación de subcategorías.	**Iteración asignada: 3**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar la funcionalidad de eliminar una categoría.	
Criterios de aceptación: • Deberá usar el endpoint específico para eliminar categorías. • El usuario deberá confirmar la eliminación de la categoría. • Deberá mostrar el resultado de la operación al usuario.	

Tabla 48. *Historia de usuario 41*

Historia de usuario	
Número: 42	**Usuario:** Todos
Nombre de la historia: Implementar el formulario de editar productos.	**Iteración asignada: 3**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar la funcionalidad de editar un producto.	
Criterios de aceptación: • El usuario podrá editar la información básica del producto. • Deberá usar el endpoint específico para la edición del producto. • Deberá verificar las entradas del usuario en los diferentes campos. • Deberá mostrar el resultado de la operación al usuario.	

Tabla 49. *Historia de usuario 42*

Historia de usuario	
Número: 43	**Usuario:** Todos
Nombre de la historia: Implementar la eliminación de productos.	**Iteración asignada: 4**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar la funcionalidad que permitirá eliminar productos al usuario.	
Criterios de aceptación: • Deberá usar el endpoint específico para eliminar productos. • El mensaje de confirmación deberá mostrar el nombre del producto a eliminar. • El usuario deberá confirmar la eliminación del producto. • Deberá mostrar el resultado de la operación al usuario.	

Tabla 50. *Historia de usuario 43.*

Historia de usuario	
Número: 44	**Usuario:** Todos
Nombre de la historia: Implementar traducciones en los formularios.	**Iteración asignada: 4**
Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Implementar las traducciones correspondientes a los errores de formulario, así como las respuestas de las operaciones.	
Criterios de aceptación: • Agregar las claves de traducción necesarias. • Mapear los errores enviados desde el backend. • Traducir las respuestas del envió de formularios.	

Tabla 51. *Historia de usuario 44.*

Historia de usuario	
Número: 45	**Usuario:** Todos
Nombre de la historia: Obtener lista de deudores.	**Iteración asignada: 4**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Obtener la lista de deudores activos.	
Criterios de aceptación: • El usuario debe ver reflejado las deudas activas al entrar a la página de deudores. • La lista mostrada al usuario debe estar paginada.	

Tabla 52. *Historia de usuario 45*

Historia de usuario	
Número: 46	**Usuario:** Todos
Nombre de la historia: Implementar formulario de crear cliente.	**Iteración asignada: 4**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar el formulario para crear un cliente.	
Criterios de aceptación: • Se deben verificar los datos ingresados por el usuario. • Debe mostrar el resultado de la operación al usuario. • Verificar que el teléfono contenga 10 caracteres y que contenga el +57. • Deberá mostrar el resultado de la operación al usuario.	

Tabla 53. *Historia de usuario 46*

Historia de usuario	
Número: 47	**Usuario:** Todos
Nombre de la historia: Implementar formulario de actualizar cliente.	**Iteración asignada: 5**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar el formulario para editar clientes.	
Criterios de aceptación: • Se debe comprobar que el cliente a actualizar no sea el cliente genérico. • El usuario debe recibir repuesta del resultado de la actualización. • Se deben comprobar los datos enviados por el usuario. • Verificar que el teléfono contenga 10 caracteres y que contenga el +57. • Deberá mostrar el resultado de la operación al usuario.	

Tabla 54. *Historia de usuario 47*

Historia de usuario	
Número: 48	**Usuario:** Todos
Nombre de la historia: Implementar funcionalidades de pago en la página de deudores.	**Iteración asignada: 5**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar las funcionalidades de obtener los pagos de una deuda activa, así como su creación.	
Criterios de aceptación: • El usuario debe poder observar los pagos previamente efectuados por el cliente para saldar su deuda.	

• El usuario debe tener la posibilidad de registrar los pagos de sus clientes. • Se deben verificar el monto ingresado por el usuario a la hora de realizar un pago. • Deberá mostrar el resultado de la operación al usuario.

Tabla 55. *Historia de usuario 48*

Historia de usuario	
Número: 49	**Usuario:** Todos
Nombre de la historia: Diseñar la página de registro e inicio de sesión.	**Iteración asignada: 5**
Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Maquetar la página donde los usuarios podrán ingresar y crear una cuenta nueva.	
Criterios de aceptación: • El usuario podrá cambiar entre formularios de registro e inicio de sesión. • Usar de referencia el mockup previsto. • Implementar los componentes necesarios. • NO implementar funcionalidades. • Hacer uso de los componentes necesarios de PrimeVue.	

Tabla 56. *Historia de usuario 49*

Historia de usuario	
Número: 50	**Usuario:** Todos
Nombre de la historia: Implementar la lógica de autenticación.	**Iteración asignada: 5**
Prioridad: Alta	**Riesgo de desarrollo:** Alta
Descripción: Implementar la funcionalidad correspondiente a la página de ingreso y registro, y la lógica de autenticación.	
Criterios de aceptación: • El usuario debe poder cambiar entre el formulario de registro e ingreso. • Al registrar una solicitud de nuevo usuario, se debe verificar la longitud de la contraseña, que no sean solo números o letras, y debe contener al menos un carácter especial, así mismo se debe verificar que la contraseña coincida con la verificación de esta. • Al registrar una solicitud de nuevo usuario se debe mostrar el resultado de la operación. • Una vez autenticado correctamente el usuario, debe ser redirigido a la página de inventario. • Se debe persistir el token de autenticación y rol del usuario. • Se debe verificar que el token exista para cualquier acción dentro de la página web.	

• Si el usuario no está autenticado, se debe redirigir a la página de login. • La sesión terminara cuando el usuario lo decida o cuando se vence el token.

Tabla 57. *Historia de usuario 50*

Historia de usuario	
Número: 51	**Usuario:** Todos
Nombre de la historia: Implementar panel de administrador.	**Iteración asignada: 6**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar el panel de administración, donde se dará la respuesta a las solicitudes de registro.	
Criterios de aceptación: • Se debe verificar que el rol del usuario sea tipo administrador, caso contrario se redirigirá a la página de inventario. • El administrador debe observar las solicitudes de registro activas y no activas. • El administrador podrá aceptar o rechazar solicitudes de nuevos usuarios. • Se debe mostrar el resultado de cada operación al administrador.	

Tabla 58. *Historia de usuario 51*

Historia de usuario	
Número: 52	**Usuario:** Todos
Nombre de la historia: Crear transacción.	**Iteración asignada: 7**
Prioridad: Media	**Riesgo de desarrollo:** Media
Descripción: Implementar la lógica referente a la creación de una transacción.	
Criterios de aceptación: • La información suministrada por el usuario debe ser verificada, si falta información, el usuario no podrá crear la transacción. • Se debe mostrar el resultado de la petición al usuario, ya sea un error o sea exitosa.	

Tabla 59. *Historia de usuario 52*

Historia de usuario	
Número: 53	**Usuario:** Todos
Nombre de la historia: Implementar funcionalidades en la página de historial.	**Iteración asignada: 7**
Prioridad: Media	**Riesgo de desarrollo:** Media

Descripción: Implementar la funcionalidad a la página de historial, donde se contempla la obtención de las transacciones creadas por el usuario, y la generación de reportes.
Criterios de aceptación: • El usuario debe observar la lista de transacciones al entrar a la página de historial. • La lista debe estar paginada. • La lista debe poder filtrarse por rango de fecha, tipo de transacción y cliente. • El usuario debe tener la opción de observar el detalle de la transacción. • El usuario podrá generar el reporte en un archivo Excel solo si se seleccionaron previamente un rango de fechas.

Tabla 60. *Historia de usuario 53*

12.3. Implementación.

Para mostrar el funcionamiento del aplicativo propuesto, se presentan las siguientes capturas de pantalla, de las vistas desarrollas para cumplir con los objetivos propuestos en el proyecto.

12.3.1. Módulo de autenticación.

En el módulo de autenticación el usuario controlará todo lo relacionado con el registro de nuevas cuentas y el inicio de sesión, así mismo será el punto de inicio del aplicativo web.

- **Inició de sesión.**

Para ingresar al aplicativo cualquier usuario que quiera interactuar con las demás opciones dentro del mismo tendrá que estar registrado, por lo tanto, el formulario de inicio de sesión es el primer punto de interacción del usuario en la página.

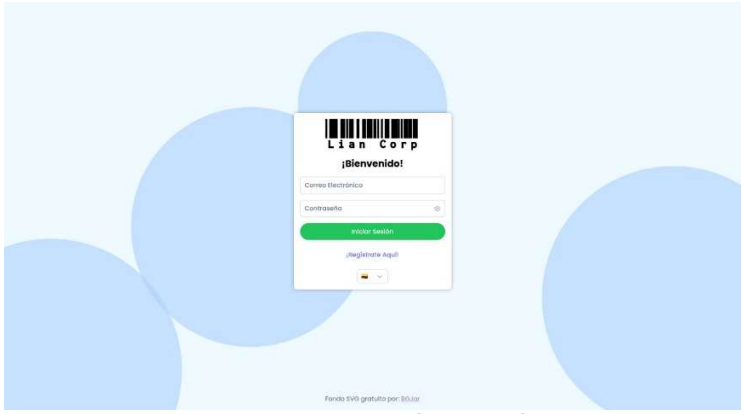


Imagen 35. *Inició de sesión*

- **Registro.**

El registro de cuentas se realizará a través del siguiente formulario, donde una vez registrada la información necesaria, el usuario tendrá que esperar la aprobación de un administrador para poder hacer su primer ingreso.

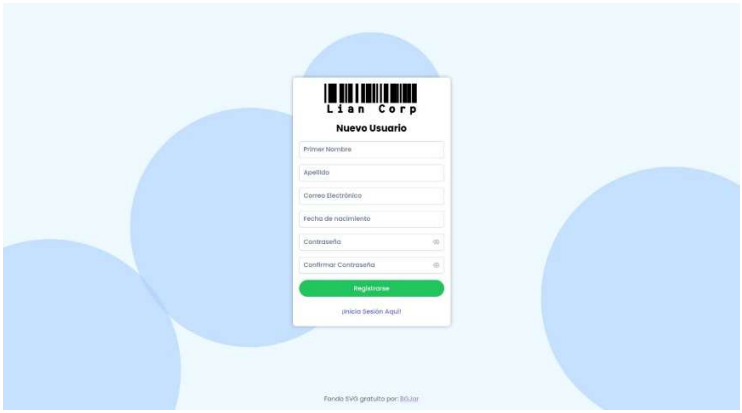


Imagen 36. *Registro.*

- **Aprobación de solicitudes.**

El usuario con rol administrador podrá aprobar y rechazar solicitudes de registro de cuentas a través de la siguiente interfaz.

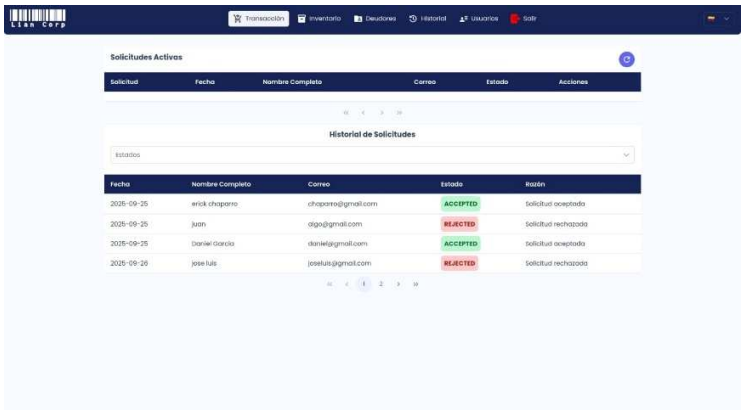


Imagen 37. *Aprobación de solicitudes*

12.3.2. Módulo de inventario.

Una vez el usuario se autentico exitosamente, será redirigido automático al módulo de inventario, en donde observará toda la información relacionada a los productos, así mismo podrá crearlos, editarlos y eliminarlos.

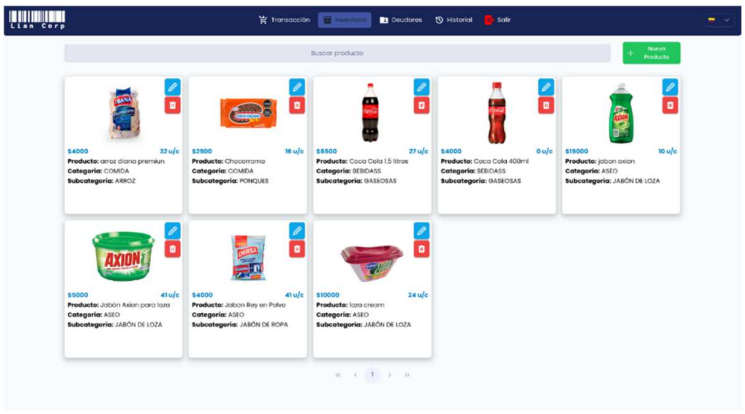


Imagen 38. *Modulo Inventario*

- **Edición de productos.**

En caso de que el usuario haya decidido editar un producto, será redirigido al siguiente formulario, donde podrá cambiar la información de este, así como los datos registrados de las categorías y subcategorías.

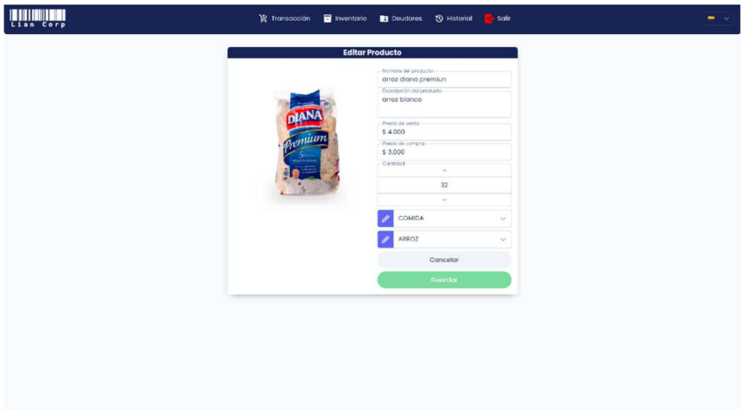


Imagen 39. *Editar productos*

- **Creación de productos.**

Si el usuario desea crear un producto, la aplicación lo redirigirá al siguiente formulario, donde también podrá crear, editar y eliminar categorías y subcategorías.

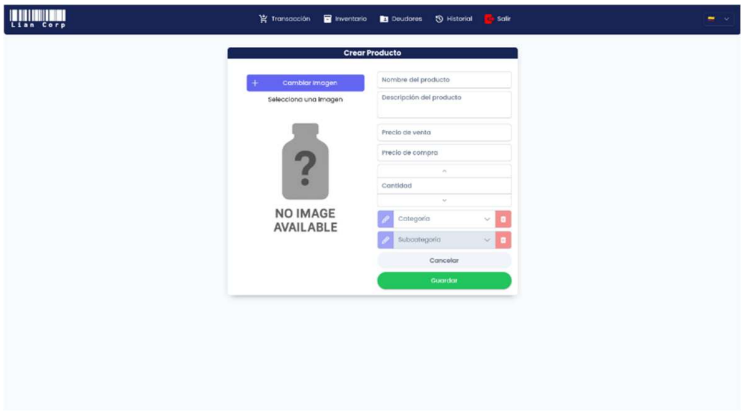


Imagen 40. *Crear Productos.*

- **Formularios de categorías y subcategorías.**

Para la edición y creación tanto de categorías como subcategorías el usuario deberá completar los formularios mostrados a continuación:

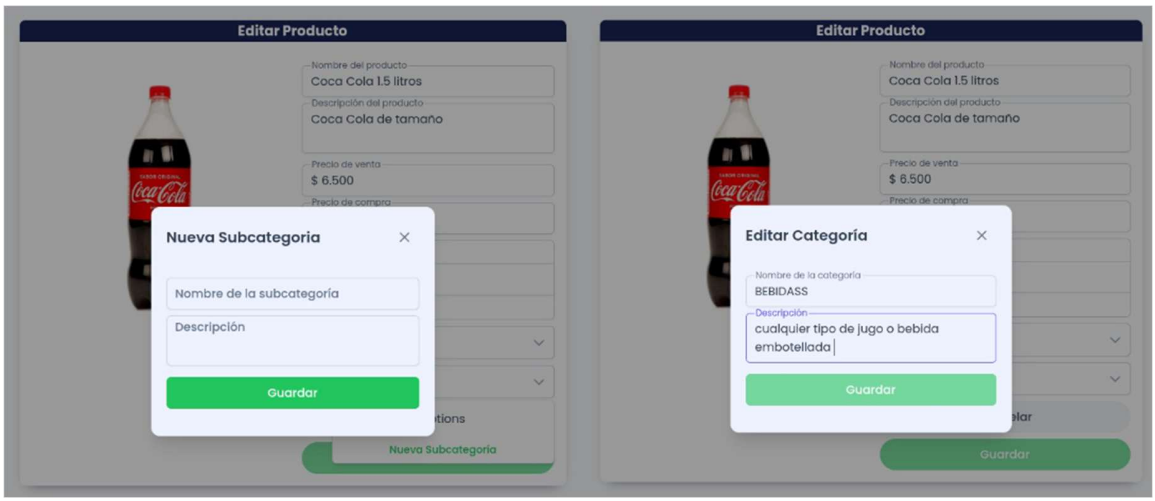


Imagen 41. *Edición y creación de categorías y subcategorías.*

12.3.3. Módulo de transacciones.

En el módulo de transacciones, el usuario tendrá el control sobre la creación, visualización y eliminación de las operaciones de compra y venta.

- **Creación de transacciones.**

En este apartado el usuario tendrá la posibilidad de crear transacciones de tipo, compra, venta (ya sean pagadas en su totalidad o a crédito).

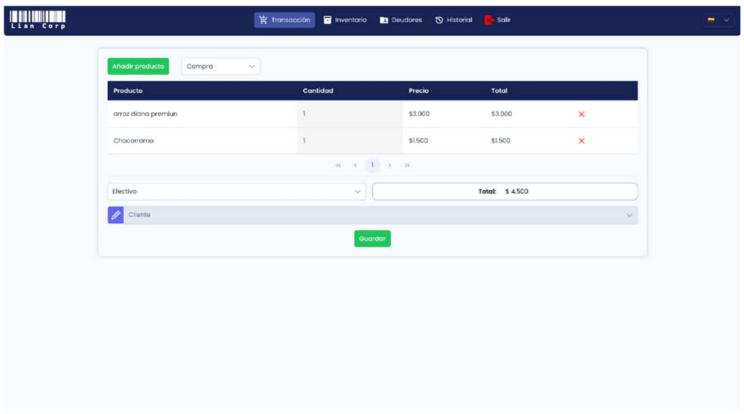


Imagen 42. *Creación de transacciones.*

- **Selección de productos.**

El usuario podrá seleccionar los productos del cliente a través de un modal que se despliega desde un botón dentro de la misma interfaz de creación de transacciones.

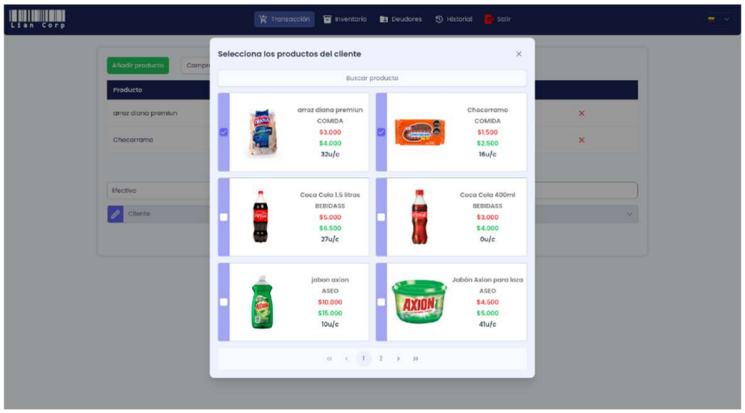


Imagen 43. *Selección de productos.*

- **Historial de Transacciones.**

El usuario podrá consultar las transacciones previamente registradas, en la interfaz de historial. Además, contara con opciones de filtrado por fecha, tipo y cliente. También permitirá visualizar los detalles de cada movimiento, así como su eliminación.

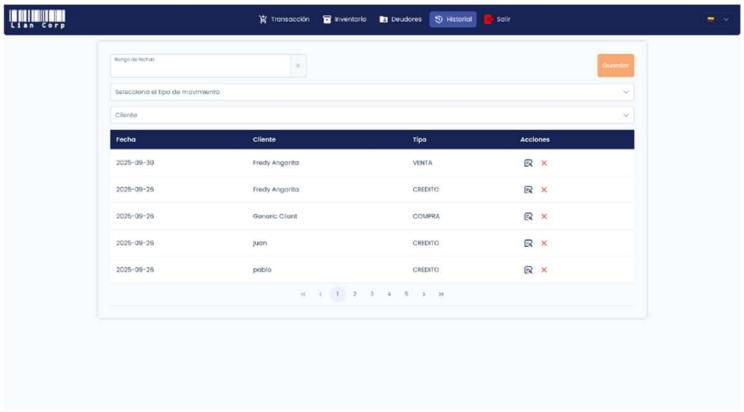


Imagen 44. *Historial de transacciones.*

- **Detalle de una transacción.**

En los detalles de una transacción, el usuario podrá ver información acerca del cliente al que se le vendió, los productos, la fecha, el tipo y el total de la transacción.

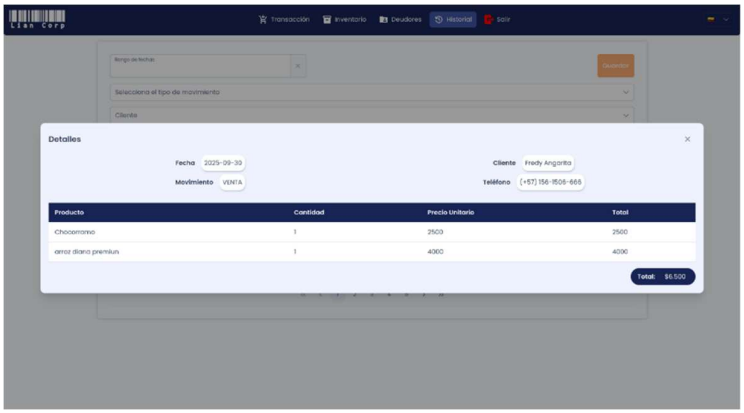


Imagen 45. *Detalle de una transacción.*

12.3.4. Módulo de deudores.

En el módulo de deudores el usuario encontrará todas las deudas activas por parte de los clientes, donde podrá filtrarlos por nombre.

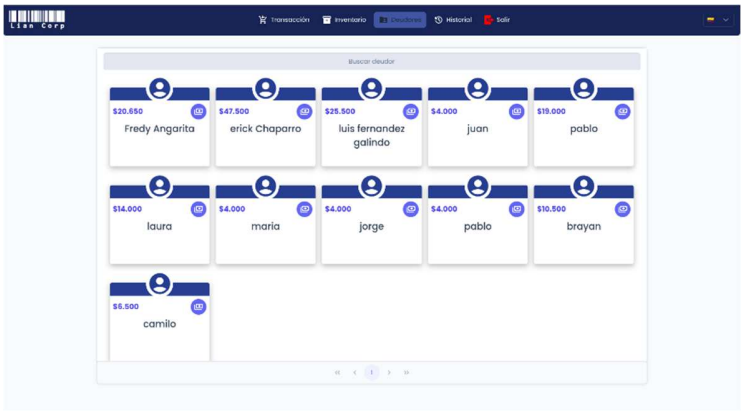


Imagen 46. *Módulo de deudores.*

- **Pagos efectuados.**

En la siguiente interfaz el usuario podrá observar todos los pagos efectuados por un cliente, así como el formulario de creación de un nuevo pago.

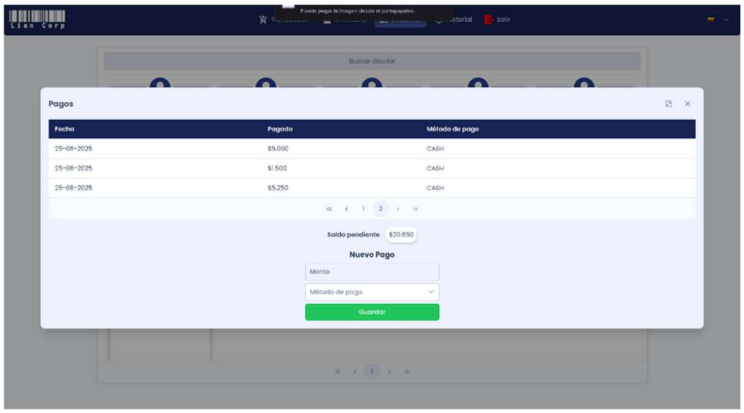


Imagen 47. *Pagos efectuados.*

13. PRUEBAS

Para garantizar la funcionalidad esperada, se hizo una revisión sobre cada módulo, en donde se buscó comportamientos no esperados, como por ejemplo él envió de formularios con datos erróneos, fallos en los filtros, etc.

13.1. Autenticación

En el módulo de autenticación se verifico todo lo relacionado a la protección de rutas, así como el funcionamiento optimo del registro, ingreso, la persistencia de la sesión y la finalización de esta.

Inicio de sesión.

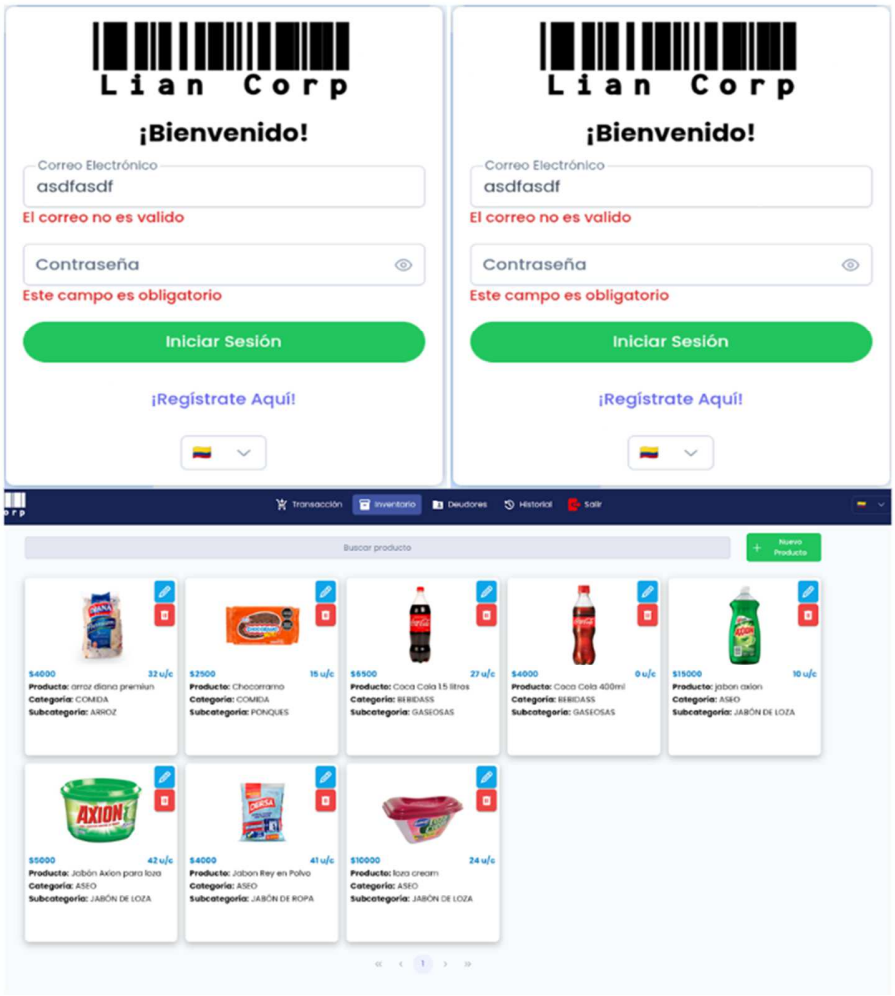


Imagen 48. *Pruebas inicio de sesión.*

La autenticación de sitio web muestra un comportamiento satisfactorio ya que:

- Verifica que los datos ingresados en el inicio de sesión sean los esperados.
- Permite ver la contraseña escrita.
- Permite dirigirse al formulario de registro.
- Si hay algún error en los datos ingresados, se muestra la respectiva retroalimentación.
- La sesión se mantiene luego de recargar la página.
- Si se intenta ingresar a algunas de las rutas del aplicativo por medio de la URL, se redirige automáticamente al inicio de sesión.
- Una vez autenticado el usuario es redirigido a la página de inventario, allí a través de la barra de navegación se puede cerrar la sesión y finalmente lo redirigirá al inicio de sesión.

Registro.

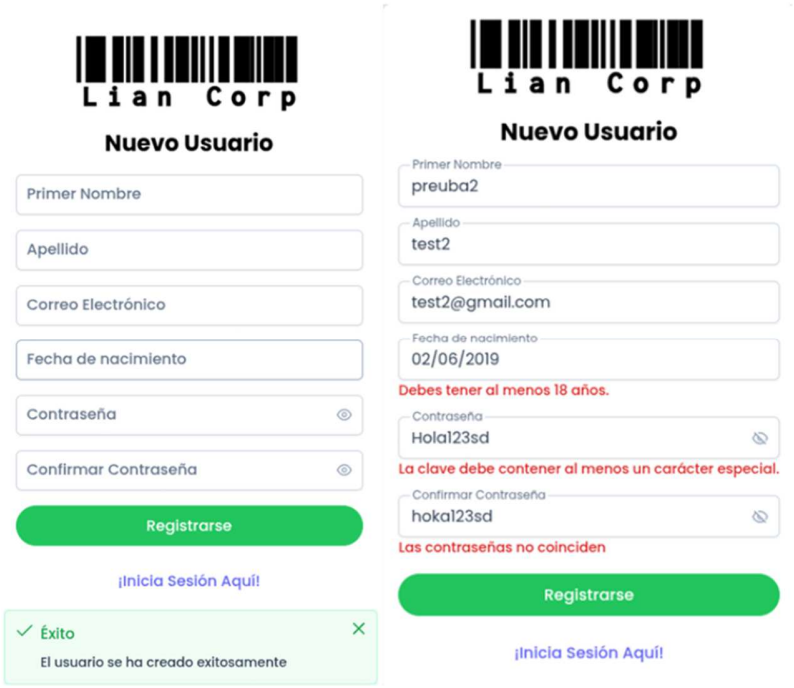


Imagen 49. *Pruebas registro de usuarios.*

El registro de nuevos usuarios se comporta de la manera esperada puesto que:

- Se verifican los campos del usuario, donde el envío de datos, solo se hace si son correctos.

- Para el primer inicio de sesión el usuario debe haber sido verificado por un administrador, caso contrario, el inicio de sesión no será posible.
- Una vez se hace el registro, el usuario es redirigido al inicio de sesión y muestra la retroalimentación del proceso.
- El registro permite cambiar al formulario de inicio de sesión.

Aprobación de nuevos usuarios.

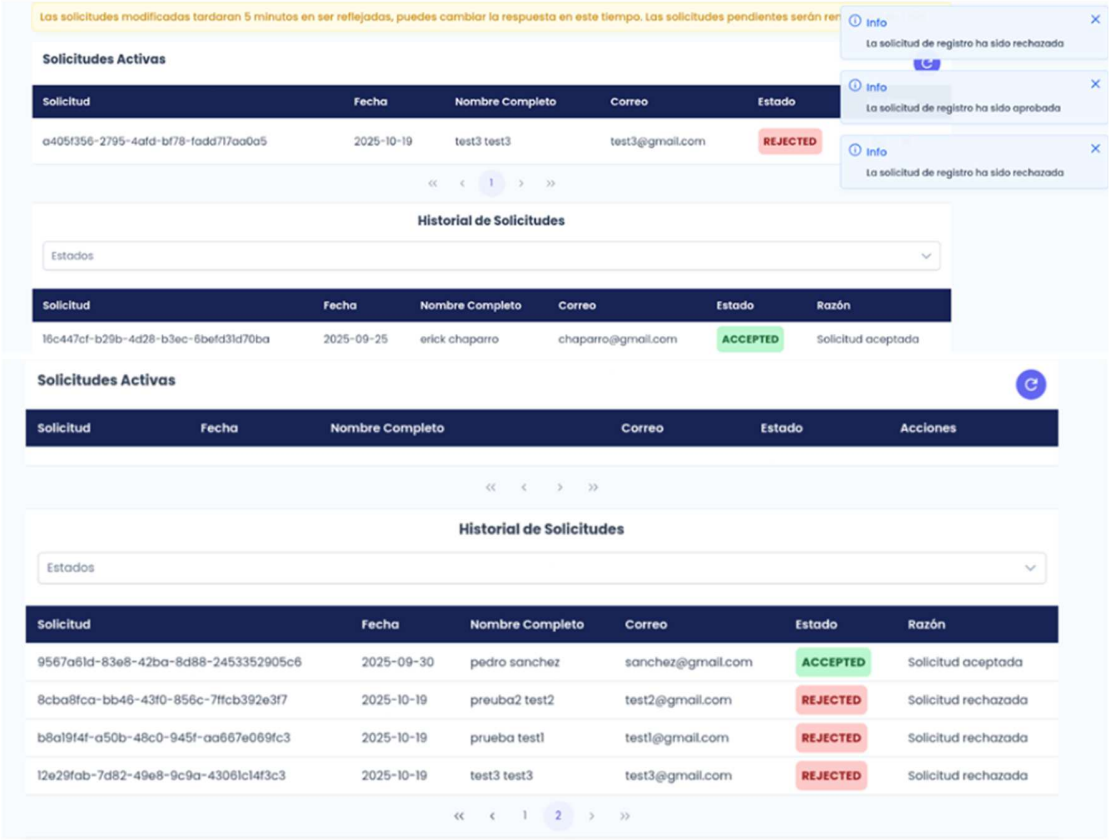


Imagen 50. *Pruebas Aprobación de nuevos usuarios.*

La página de usuarios se comporta de la manera esperada puesto que:

- Solo los usuarios de tipo “ADMIN” tienen permitido el acceso a la ruta.
- La información sobre las solicitudes ya activas y resueltas cargan al entrar en la página, así mismo la paginación en cada una de estas tablas funciona bien.

- Al responder una solicitud, el cambio del estado es realizado, y posteriormente muestra el resultado de la operación.
- La respuesta a las solicitudes activas puede ser cambiada hasta 5 minutos después de hacer el primer cambio en el estado, una vez acabado el tiempo la solicitud será enviada al historial y no se podrán hacer cambios.
- La tabla de historial permite filtrar según el estado de la solicitud.
- El botón de recarga permite refrescar las tablas para revisar si los cambios en la base de datos fueron realizados.

13.2. Inventario.

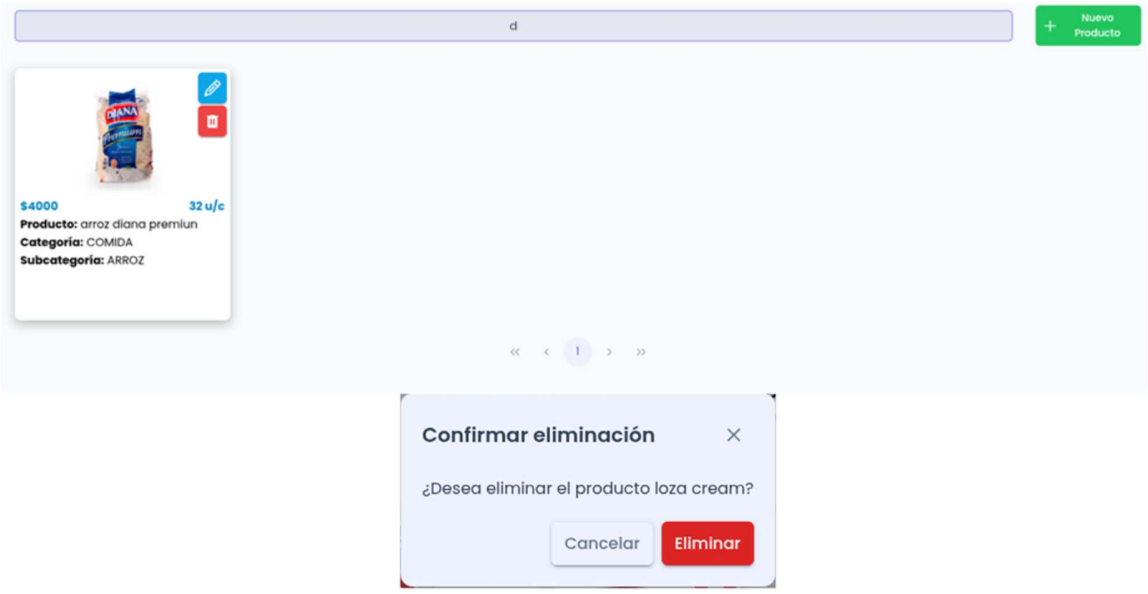


Imagen 51. *Pruebas inventario.*

La página de inventario se comporta correctamente puesto que:

- Los productos previamente registrados, se muestran adecuadamente.
- Se le permite al usuario filtrar productos por nombre.
- La eliminación de un producto pide una confirmación y una vez aceptada, se procede a hacer la operación, en consecuencia, se refresca la información y se muestra el mensaje de éxito de la acción.
- El usuario cuenta con acceso constante a los formularios de crear y editar un producto, y se redirecciona correctamente.

Formulario de creación de productos.

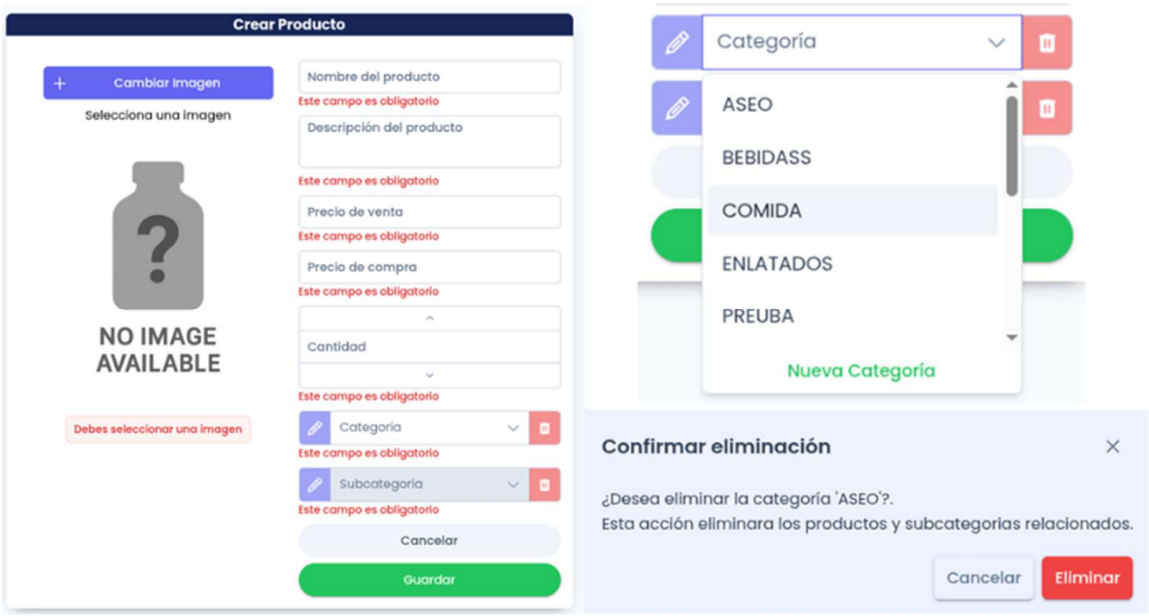


Imagen 52. *Pruebas nuevo producto.*

El funcionamiento del formulario de crear productos es correcto ya que:

- El formulario valida que cada campo cumpla con las condiciones necesarias para poder realizar el envío de la información.
- Al cargar una imagen, se muestra una previsualización de esta, y cuando la imagen tiene un tamaño mayor a 10MB, se muestra un mensaje de error.
- La eliminación y edición tanto de categorías como subcategorías, solo se habilita si hay una elección en el campo correspondiente.
- Se valida que tanto para la selección, creación y edición de subcategorías, se haya elegido previamente una categoría.
- La eliminación de categorías y subcategorías exigen una confirmación, que una vez aceptada, muestra el resultado del proceso.
- Al abrir el selector de categorías y subcategorías, la opción de creación siempre es visible.
- Cuando se crea una categoría o subcategoría, la lista del selector es actualizada automáticamente.
- Al hacer el envío de la información del producto, se muestra el resultado de la acción y se limpian los campos.

Formulario de edición de productos.

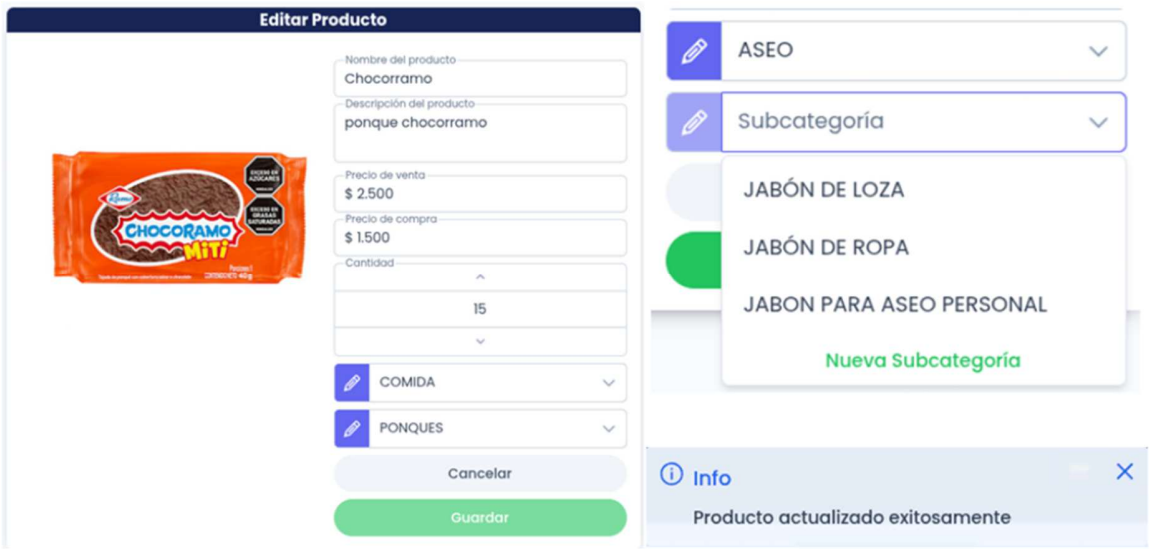


Imagen 53. *Pruebas edición de productos.*

El formulario de edición cuenta con un comportamiento adecuado debido que:

- Los campos son cargados automáticamente con la información registrada en el backend.
- El envío de información es habilitado siempre y cuando haya alguna modificación y sea válido.
- El botón de edición de categorías y subcategorías, solo se habilita si hay una selección hecha.
- Se valida que tanto para la creación y edición de subcategorías, exista una elección previa de una categoría.
- Al abrir el selector de categorías y subcategorías, la opción de creación siempre es visible.
- Cuando se crea una categoría o subcategoría, la lista del selector es actualizada automáticamente.
- Una vez presionado el botón para confirmar los cambios, se procede a hacer el envío y se redirecciona a la página del inventario.

Edición y creación de categorías y subcategorías.



Imagen 54. *Prueba formularios de subcategorías y categorías.*

El funcionamiento de los formularios para la creación y edición tanto de categorías y subcategorías se considera correcto por:

- Los formularios de edición son cargados automáticamente con la información registrada cuando es abierto.
- En todos los formularios, se valida que los campos rellenados, cumplan con las condiciones de envío.
- El envío de información en formularios de edición solo se realiza si hubo una modificación y al igual que con la creación de categorías y subcategorías, los datos ingresados deben ser válidos.

13.3. Transacción.

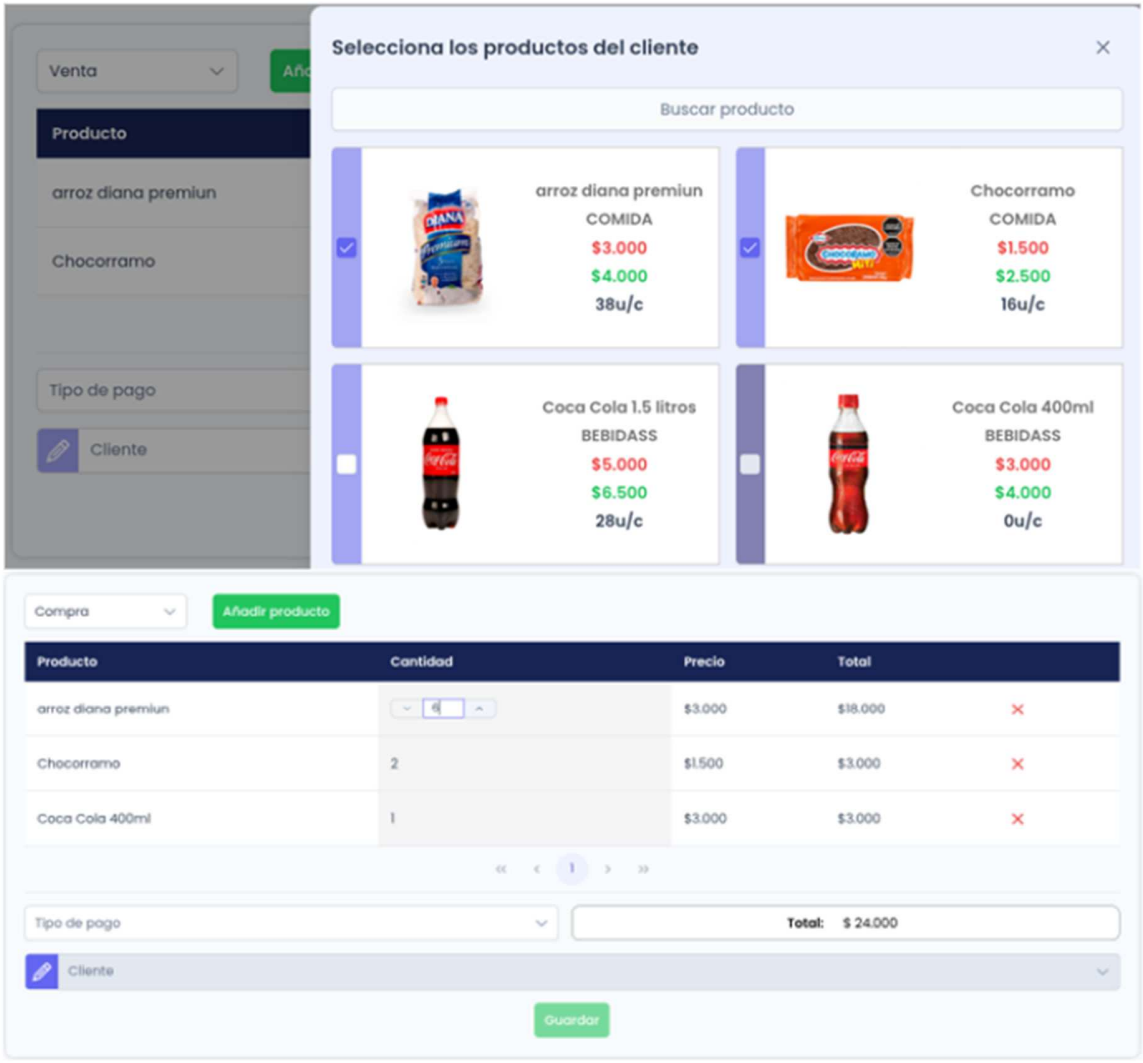


Imagen 55. *Pruebas creación de transacciones.*

Se encuentra el funcionamiento de la página de creación de transacciones optimo puesto que:

- Una vez seleccionado algún producto dentro del modal que se despliega con el botón de “añadir producto”, la tabla se actualiza automáticamente con la información de este último.
- Cuando el tipo de transacción sea diferente de “compra” y el producto tenga 0 existencias, se deshabilita la selección de este último.

- Al presionar sobre la columna de cantidad de cualquier producto, se habilita la edición, una vez se confirma la nueva cantidad el valor total de la tupla es recalculado.
- El cálculo del precio total se realiza a partir de precios diferenciales entre transacciones de compra y venta.
- La selección de clientes se habilita o deshabilita según el tipo de transacción.
- La edición de un cliente solo se habilita si existe una selección.
- Una vez se confirma que todos los campos contengan la información necesaria y activar el botón de “guardar” se mostrará el resultado de la acción y limpiará los campos.

Creación y edición de usuarios.

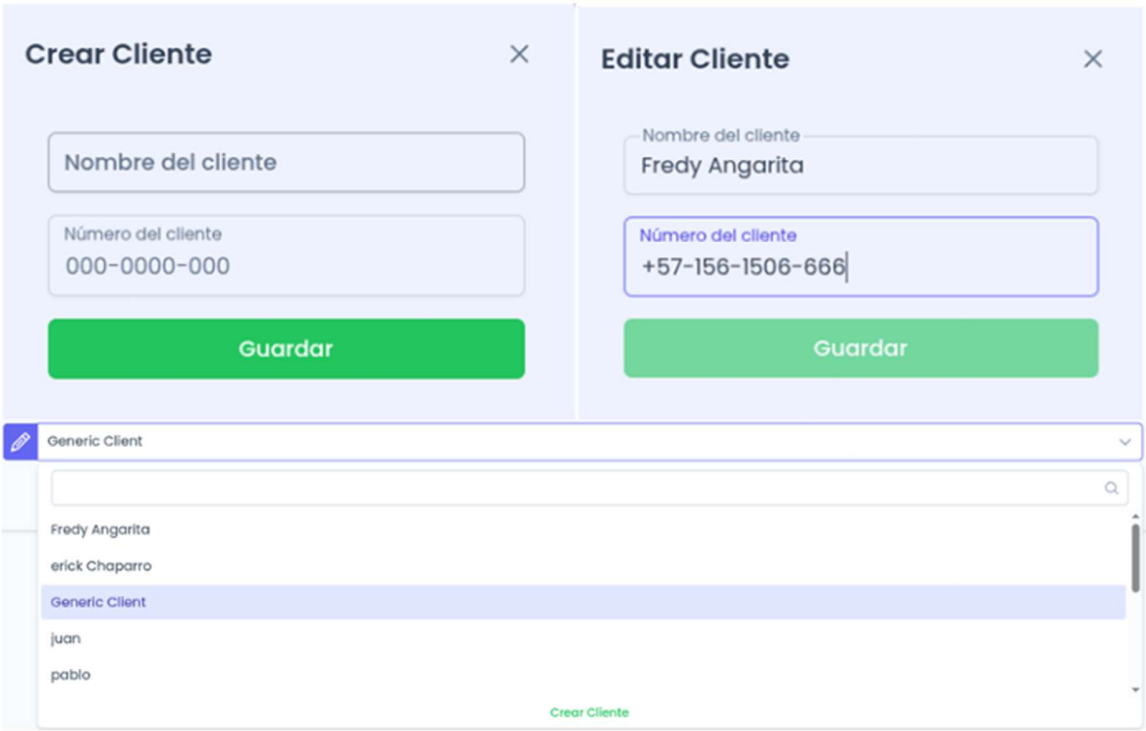


Imagen 56. *Pruebas creación y edición de usuarios.*

Los formularios de creación y edición funcionan correctamente debido que:

- El usuario siempre tiene acceso a la creación de clientes por medio del botón de “crear clientes”.

- El selector de clientes permite el filtrado por nombre.
- El envío de datos, solo se realiza cuando los datos sean correctos.

Historial de transacciones.

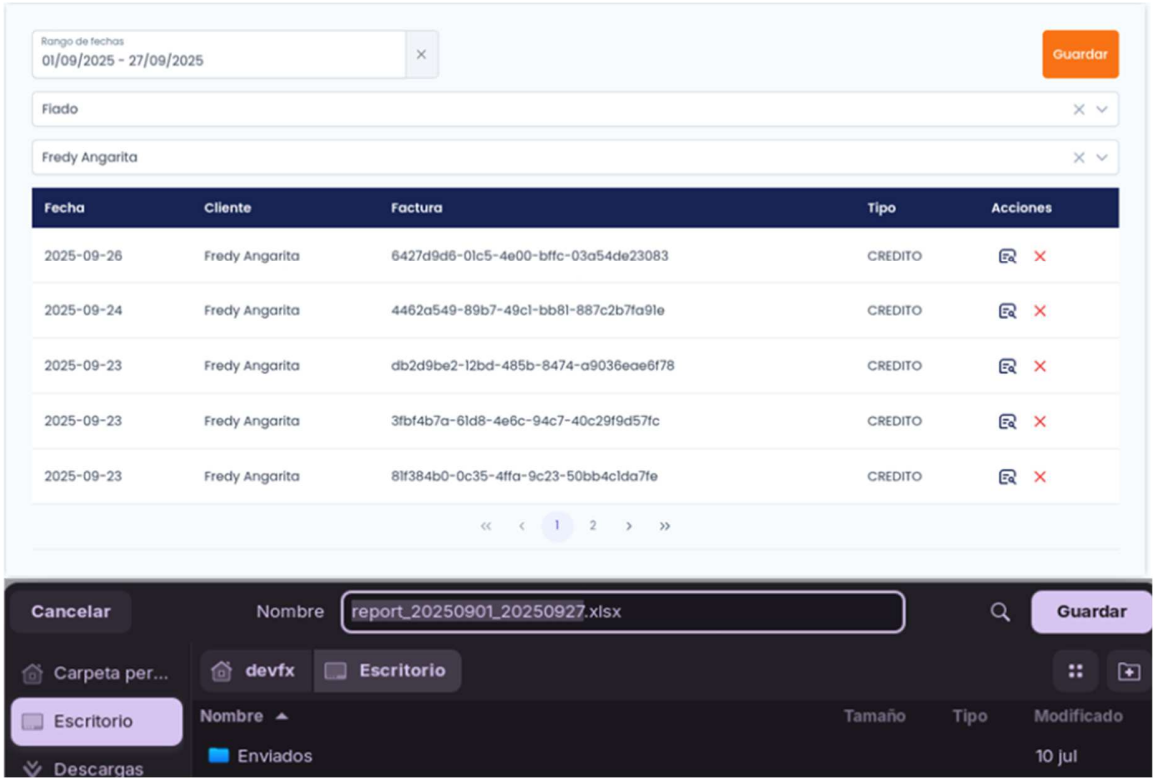


Imagen 57. *Prueba historial de transacciones.*

La página de transacciones cuenta con un funcionamiento correcto puesto que:

- Las transacciones son cargadas automáticamente al ingresar a la página y se organizan de la fecha más reciente a la más antigua.
- Permite el filtrado por rango de fechas, tipo de transacción y cliente.
- El usuario puede observar el detalle de la transacción realizada, donde se muestran datos relevantes como la fecha, el cliente, y los productos vendidos.
- Permite eliminar transacciones que hayan sido creadas con errores.

- La generación de archivos se habilita una vez se seleccione un rango de fechas y cuando es accionado el botón correspondiente, se abre el administrador de archivos del sistema para seleccionar la ruta de guardado.

13.4. Deudores.

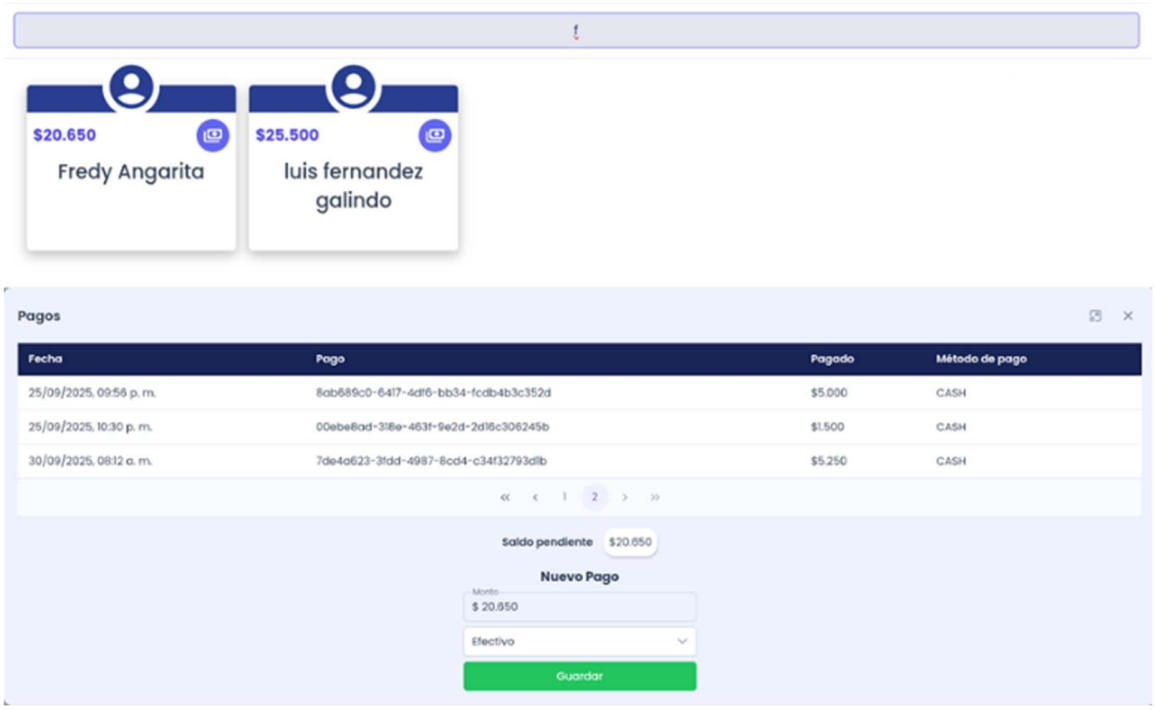


Imagen 58. *Pruebas página de deudores.*

La página de deudores se comporta correctamente ya que:

- Al ingresar a la página, se cargan correctamente los datos sobre las deudas activas.
- Al presionar sobre el botón de una deuda activa, se despliega el modal que contiene el historial y el formulario de pagos.
- El monto máximo que se puede registrar siempre es el total de la deuda seleccionada.
- Una vez los datos ingresados por el usuario cumplan con las validaciones del formulario y el botón de envío sea accionado, el modal se cierra y muestra el resultado de la operación.

14. RESULTADOS.

- Se desarrollaron los módulos de creación de transacciones, seguimiento de deuda por parte de los clientes y gestión de inventario, diseñados para satisfacer y facilitar los procesos del emprendimiento, pasando de una gestión y control de manera manual, a una forma mucho más rápida y sistematizada.

- Se implementó un sistema de seguridad de autentificación y autorización compuesto por la estrategia de JWT el cual el microservicio de usuario es encargado de proveer el token, junto con el Apigateway para restringir los servicios a los usuarios permitidos, adicional, se integró un módulo para administradores de aprobación de solicitudes de registro, esto con el fin, de que no todas las personas que se registren puedan ingresar al aplicativo desarrollado para el emprendimiento Distribuidora Lian.

- Se diseñó una arquitectura basada en microservicios para el sistema, promoviendo una mayor interpolación entre los componentes y su comunicación, separando lógica de negocio en su microservicio respectivo y modelo de datos, esto con el fin de brindar una mayor sostenibilidad, flexibilidad, mantenibilidad y escalamiento a futuro.

- Se realizaron pruebas funcionales de todo lo desarrollado, donde se evidenció el correcto funcionamiento de los diferentes módulos del sistema y el procesamiento de la lógica de negocio empleada. Se observa un frontend amigable al usuario y seguridad por parte del backend.

15. CONCLUSIONES.

15.1. Aspecto organizacional

- El seguimiento de la deuda es un procedimiento que se debe realizar con gran rigurosidad, puesto que puede conllevar a malentendidos y discusiones. Por lo tanto, un registro detallado sobre los productos, precios, cantidades, así como los pagos efectuados para saldar este tipo de transacción es de vital de importancia para garantizar la transparencia del proceso para las partes involucradas.

15.2. Aspecto tecnológico

- El registro de información en formatos como papel, cuadernos, libros es una práctica ineficiente por la pobre trazabilidad de los datos, así como su difícil replicación y su poca perdurabilidad en el tiempo, cualidades donde los sistemas digitales tienen una gran ventaja sobre los medios tradicionales de registro.

- El uso de motores de bases de datos para el almacenaje de información permite una mayor persistencia, filtrado y seguimiento de los datos, así como el control sobre ellos.

- El uso de Nuxt ha permitido desarrollar componentes reusables, así como módulos escalables, permitiendo ofrecer al usuario un diseño responsivo y amigable para su uso. Además de contribuir a la reducción errores de digitación por medio de formularios que comprueban los datos ingresados.

- La implementación de tokens JWT ha permitido generar roles y separar funciones específicas según el tipo de usuario. Esta capa de protección aplicada sobre cada endpoint y ruta del aplicativo garantiza un acceso seguro y controlado a los recursos.

15.3. Aspecto funcional

- La exportación de reportes de Deudas y Transacciones en un formato Excel, es esencial si se desea una interoperabilidad y autonomía para los usuarios, además de que fortalece el propósito del sistema de control de deudas, consolidando los datos en un solo archivo y añadiendo al sistema una capa de ofimática.

16. RECOMENDACIONES

- Como primera recomendación para mejorar el producto entregado, en la parte del frontend evitar el uso de “Magic Strings” para el manejo de traducciones ya que usar las claves de los archivos JSON directamente en el template de los diferentes componentes, entorpece la actualización de la interfaz de usuario, por ejemplo, títulos, descripciones, etiquetas y botones.
- Se recomienda validar la información de los usuarios que se registran en el aplicativo, puesto que una vez aprobado por el usuario administrador y pasen los 5 minutos de verificación, no se podra hacer una eliminación del usuario registrado desde el aplicativo.
- Para mantener una organización en los productos creados, se recomienda no crear categorías y subcategorías redundantes puesto que entorpeceria la clasificación de la mercancia.
- Implementar una integración con WhatsApp para el recordatorio periódico de la venta fiadas realizadas por el cliente y aún no ha cancelado.
- Creación de un módulo para los clientes, en donde puedan observar sus compras realizadas, historial de deudas y los montos que ha abonado, esto con el fin de dar mayor transparencia al proceso de seguimiento de deuda.

17. BIBLIOGRAFÍA

[1] N. Kraenau, M. Silva and P. Castañeda, "Hybrid Model Based on Machine Learning for the Prediction of Consumer Credit Delinquency in the Banking Sector of Peru," 2024 International Symposium on Intelligent Robotics and Systems (ISoIRS), Changsha, China, 2024, pp. 305-308, doi: 10.1109/ISoIRS63136.2024.00066.

[2] ¿CÓMO APRUEBAN o deniegan las entidades financieras las solicitudes de prÃ©stamo? [Anónimo]. BBVA NOTICIAS [página web]. [Consultado el 12, octubre, 2024]. Disponible en Internet: <https://www.bbva.com/es/salud-financiera/como-aprueban-o-deniegan-un-prestamo-criterio-entidades-financieras/>.

[3] ESTRATEGIAS EFECTIVAS para la gestión de deudas en una empresa [Anónimo]. heisohn [página web]. [Consultado el 12, octubre, 2024]. Disponible en Internet: <https://www.heinsohn.co/blog/gestion-de-deudas-en-una-empresa/>.

[4] RUIZ, Ramon. Gestión de cobros - qué es, definición Y concepto. S&F [página web]. (septiembre, 2023). [Consultado el 23, noviembre, 2024]. Disponible en Internet: <https://www.segurosyfinanzashoy.com/gestion-de-cobros-que-es-definicion-y-concepto/>.

[5] 5 PUNTOS problemáticos de la gestión de registros manuales | Access [Anónimo]. Access [página web]. (1, agosto, 2022). [Consultado el 12, octubre, 2024]. Disponible en Internet: <https://www.accesscorp.com/es-cr/blog/5-puntos-problematicos-de-la-gestion-de-registros-manuales/>.

[6] ¿POR QUE su empresa necesita un Software de gestión empresarial? - OnERP [Anónimo]. OnERP [página web]. [Consultado el 29, abril, 2023]. Disponible en Internet: <https://onerp.es/software-de-gestion-empresarial/>. (estos son de lo que esta en el planteamiento)

[7] GUTIÉRREZ, Valentina y JARAMILLO, Diana Patricia. Reseña del Software Disponible en Colombia Para la Gestión de Inventarios en Cadenas de AbastecimientoReview of the software available for inventory management in supply chains in Colombia. ScienceDirect [página web]. (31, marzo, 2009). [Consultado el 18, noviembre, 2024]. Disponible en Internet: <https://www.sciencedirect.com/science/article/pii/S0123592309700658>.

[8] BBVA. Qué es la morosidad y cuáles son sus consecuencias. BBVA NOTICIAS [página web]. (22, julio, 2024). [Consultado el 10, diciembre, 2024]. Disponible en Internet: <https://www.bbva.com/es/salud-financiera/que-es-la-morosidad-y-que-consecuencias-tiene/>

[9] MOROSIDAD - ¿Qué es la morosidad? | SumUp Facturas [Anónimo]. SumUp - Una forma más inteligente de cobrar [página web]. [Consultado el 10, diciembre, 2024]. Disponible en Internet: <https://www.sumup.com/es-es/facturas/glosario/morosidad/>.

[10] G. MARTÍNEZ, Eduardo. ¿Qué es la morosidad, cuáles son los tipos y sus consecuencias? idealista/news [página web]. (5, octubre, 2024). [Consultado el 10, diciembre, 2024]. Disponible en Internet: <https://www.idealista.com/news/inmobiliario/vivienda/2024/10/05/819541-que-es-la-morosidad-cuales-son-los-tipos-y-sus-consecuencias#:~:text=La%20principal%20consecuencia%20de%20la,establecer%20relaciones%20comerciales%20o%20contractuales>.

[11] EUROINNOVA. Cobro de morosos: tipos de deudores y métodos de cobranza. Euroinnova International Online Education [página web]. [Consultado el 10, diciembre, 2024]. Disponible en Internet: <https://www.euroinnova.com/articulos/escuelaiberoamericana/cobro-de-morosos>.

[12] ZENDESK. ¿Cómo cobrar una cartera de clientes morosos sin ofenderlos? Zendesk [página web]. (23, abril, 2023). [Consultado el 10, diciembre, 2024]. Disponible en Internet: <https://www.zendesk.com.mx/blog/como-se-le-cobra-a-un-cliente-moroso/>.

[13] MOONFLOW. Gestión de cobranza: definición e importancia. Software de Cobranzas en piloto automático [página web]. (7, mayo, 2023). [Consultado el 16, diciembre, 2024]. Disponible en Internet: <https://www.moonflow.ai/blog/definicion-gestion-de-cobranza>.

[14] KOBRA. ¿Qué es la cobranza preventiva? - Kobra. Kobra [página web]. [Consultado el 16, diciembre, 2024]. Disponible en Internet: <https://kobra.red/blog/que-es-la-cobranza-preventiva>.

[15] GARCÍA VILLACORTA, Arturo. Gestión de cobranzas: ¿Cómo motivar a los deudores a pagar? | Conexión ESAN. ESAN Graduate School of Business - ESAN [página web]. (31, mayo, 2022). [Consultado el 16, diciembre, 2024]. Disponible en Internet: <https://www.esan.edu.pe/conexion-esan/gestion-de-cobranzas-como-motivar-a-los-deudores-a-pagar>.

[16] EL MEJOR software de gestión de inventario de código abierto | Odoo [Anónimo]. Odoo [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://www.odoo.com/es_ES/app/inventory?utm_source=google&utm_medium=cpc&utm_campaign=CO-ES-Inventory_App-Specific-Values&gad_source=1&gclid=Cj0KCQiA4rK8BhD7ARIsAFe5LXJh

glCS8hom5R_Ya-1onU-OZgkV5v_UZrmt3vWMxUuoRfkMEUsXG5caAnc8EALw_wcB>.

[17] SHOPIFY [Anónimo]. Shopify [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://www.shopify.com/ph>.

[18] ONLINE INVENTORY management software | zoho inventory [Anónimo]. Zoho | Cloud-Software-Suite fÃ¼r Unternehmen [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://www.zoho.com/inventory/>.

[19] MOONFLOW. Software de Cobranzas en piloto automático. Software de Cobranzas en piloto automático [página web]. [Consultado el 22, enero, 2025]. Disponible en Internet: <https://www.moonflow.ai/es>.

[20] INTIZA. Intiza - gestión de cobranzas. Intiza [página web]. [Consultado el 22, enero, 2025]. Disponible en Internet: <https://www.intiza.com/cobranzas>.

[21] DUEMINT. Software de cobranza, conciliación y pago a proveedores. Duemint [página web]. [Consultado el 22, enero, 2025]. Disponible en Internet: <https://www.duemint.com/>.

[22] CUADRADO, Gustavo Cimas. Qué es un lenguaje de programación: Guía para principiantes | OpenWebinars. OpenWebinars.net [página web]. (16, julio, 2020). [Consultado el 19, enero, 2025]. Disponible en Internet: <https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/#qué-es-un-lenguaje-de-programación>.

[23] JAVASCRIPT WITH syntax for types. [Anónimo]. TypeScript: JavaScript With Syntax For Types. [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://www.typescriptlang.org/>.

[24] ¿QUÉ ES java? - explicación del lenguaje de programación java - AWS [Anónimo]. Amazon Web Services, Inc. [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://aws.amazon.com/es/what-is/java/>.

[25] FRAMEWORK: QUÉ es, para qué sirve y algunos ejemplos [Anónimo]. Unir Formacion Profesional [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <http://www.unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/>.

[26] NUXT. Introduction · get started with nuxt v3. Nuxt [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://nuxt.com/docs/3.x/getting-started/introduction>.MURADAS, Yanina. Qué es Spring Framework y por qué usarlo | OpenWebinars.

OpenWebinars.net [página web]. (5, junio, 2018). [Consultado el 19, enero, 2025]. Disponible en Internet: <https://openwebinars.net/blog/conoce-que-es-spring-framework-y-por-que-usarlo/>.

[27] MURADAS, Yanina. Qué es Spring Framework y por qué usarlo | OpenWebinars. OpenWebinars.net [página web]. (5, junio, 2018). [Consultado el 19, enero, 2025]. Disponible en Internet: <https://openwebinars.net/blog/conoce-que-es-spring-framework-y-por-que-usarlo/>.

[28] ¿QUÉ SON y para qué sirven los microservicios? Beneficios de los microservicios [Anónimo]. Red Hat - We make open source technologies for the enterprise [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://www.redhat.com/es/topics/microservices?sc_cid=7015Y0000048RsyQAE&gad_source=1&gclid=Cj0KCQiA4rK8BhD7ARIsAFe5LXKK_mVOJVb7os50n3gqMo6q0kaZOUvaNzek7uOh0sPpT_kIwxPJIE0aAgOvEALw_wcB>.

[29] ¿QUÉ ES una base de datos? - Explicación de las bases de datos en la nube - AWS [Anónimo]. Amazon Web Services, Inc. [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://aws.amazon.com/es/what-is/database/>.

[30] ¿QUÉ ES SQL? - Explicación de lenguaje de consulta estructurado (SQL) - AWS [Anónimo]. Amazon Web Services, Inc. [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://aws.amazon.com/es/what-is/sql/>.

[31] ¿QUÉ ES PostgreSQL? [Anónimo]. Kinsta® [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://kinsta.com/es/base-de-conocimiento/que-es-postgresql/#qu-es-postgresql>.

[32] QUÉ ES la arquitectura hexagonal, ventajas y desventajas [Anónimo]. Blog de HubSpot | Marketing, Ventas, Servicio al Cliente y Sitio Web [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://blog.hubspot.es/website/que-es-arquitectura-hexagonal>.

[33] ¿QUÉ ES una API de REST? [Anónimo]. Red Hat - We make open source technologies for the enterprise [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://www.redhat.com/es/topics/api/what-is-a-rest-api?sc_cid=7015Y0000048RsyQAE&gad_source=1&gclid=Cj0KCQiA4rK8BhD7ARIsAFe5LXLmP3U6yuh4f9PayjIsg4O9x7em088XgBOwqKllHZKsPfmLp3Zz3z4aAhzfEALw_wcB>.

[34] JSON PARA principiantes: qué es, para qué sirve y ejemplos [Anónimo]. Blog de HubSpot | Marketing, Ventas, Servicio al Cliente y Sitio Web [página web]. [Consultado el 19, enero, 2025]. Disponible en Internet: <https://blog.hubspot.es/website/que-es-json>.

[35] MAGAÑA, Luis Miguel López. Qué es Json Web Token y cómo funciona | OpenWebinars. OpenWebinars.net [página web]. (17, enero, 2020). [Consultado el 19, enero, 2025]. Disponible en Internet: <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>.

[36] PAEZ CARDENAS, Patricia; ARIAS PERALTA, Christian David y WANUMEN SILVA, Luis Felipe. Metodología para elaboración de requerimientos en aplicaciones con servicios web | Revista Vínculos. Home Page [página web]. (6, noviembre, 2018). [Consultado el 22, enero, 2025]. Disponible en Internet: <https://doi.org/10.14483/2322939X.13505>.

[37] ATLASSIAN. Scrumban: domina dos metodologías ágiles | Atlassian. Atlassian [página web]. [Consultado el 22, enero, 2025]. Disponible en Internet: <https://www.atlassian.com/es/agile/project-management/scrumban#:~:text=Scrumban%20es%20una%20metodología%20ágil,la%20complejidad%20de%20un%20proyecto.>.

[38] LAOYAN, Sarah. Scrumban: lo mejor de dos metodologías ágiles [2024] • Asana. Asana [página web]. (14, febrero, 2024). [Consultado el 22, enero, 2025]. Disponible en Internet: <https://asana.com/es/resources/scrumban>.

[39] KANBANTOOL. ¿Qué es scrumban? | kanban tool. Kanban Tool - Kanban Boards for Business | Kanban Software [página web]. [Consultado el 22, enero, 2025]. Disponible en Internet: <https://kanbantool.com/es/guia-kanban/que-es-scrumban>.

[40] ATLASSIAN. Scrumban: domina dos metodologías ágiles | Atlassian. Atlassian [página web]. [Consultado el 9, mayo, 2024]. Disponible en Internet: <https://www.atlassian.com/es/agile/project-management/scrumban#:~:text=Scrumban%20es%20una%20metodología%20ágil,la%20complejidad%20de%20un%20proyecto.>.

[41] LAOYAN, Sarah. Scrumban: lo mejor de dos metodologías ágiles [2024] • Asana. Asana [página web]. (14, febrero, 2024). [Consultado el 9, mayo, 2024]. Disponible en Internet: <https://asana.com/es/resources/scrumban>.

[42] KANBAN TOOL. ¿Qué es Scrumban? | Kanban Tool. Kanban Tool - Kanban Boards for Business | Kanban Software [página web]. [Consultado el 9, mayo, 2024]. Disponible en Internet: <https://kanbantool.com/es/guia-kanban/que-es-scrumban>.

[43] HANS ZAMORA CARRILLO y NORBERTO NOVOA TORRES. Nociones, consideraciones y ventajas de la inteligencia de negocios BI. En: Revista vínculos [en línea]. 19, diciembre, 2019. vol 16, no 2 [consultado el 19, enero, 2025], disponible en Internet: < https://orcid.org/0000-0002-9688-4271>.

[44] PAEZ CARDENAS, Patricia; ARIAS PERALTA, Christian David y WANUMEN SILVA, Luis Felipe. Metodología para elaboración de requerimientos en aplicaciones con servicios web. En: Revista vínculos [en línea]. 6, noviembre, 2018. vol. 15, no. 2 [consultado el 24, enero, 2024], p. 160-174. Disponible en Internet: <https://doi.org/10.14483/2322939X.13505> ISSN 2322-939X

[45] PEÑA-ROBAYO, Martha; MORENO, Ana del Pilar. Amenazas emergentes en la computación en la nube: desafíos de seguridad y respuesta. En: Revista vínculos [en línea]. 20, abril, 2024. vol 21, no 1 [consultado el 24, enero, 2025], disponible en Internet: <https://revistas.udistrital.edu.co/index.php/vinculos/article/view/22566>

[46] GALINDO RAMÍREZ, Ximena; GÓMEZ DUARTE, Miguel Andrés y HERNÁNDEZ GUTIÉRREZ, Jairo. Seguridad en la nube, evolución indispensable en el siglo XXI. En: Revista vínculos [en línea]. 11, junio, 2019. vol. 16, no. 1 [consultado el 24, enero, 2025], p. 110-127. Disponible en Internet: <https://doi.org/10.14483/2322939x.15535>. ISSN 2322-939X.

[47] MURILLO GONZÁLEZ, Gilberto; GUZMAN CARAVEO, Zully Kristel y MAYO, Herman Aguilar. Identificación de los elementos necesarios para la elaboración de Cubos OLAP utilizando Microsoft Analysis Services con SQL SERVER y Excel. En: Revista vínculos [en línea]. 25, junio, 2020. vol. 17, no. 1 [consultado el 25, enero, 2025], p. 32-41. Disponible en Internet: <https://doi.org/10.14483/2322939x.15573>. ISSN 2322-939X.

[48] ROMERO, Gustavo Andrès, et al. Revisión y perspectivas para la construcción de bases de datos robustas con datos faltantes: caso aplicado a información financiera. En: Tecnura [en línea]. 1, enero, 2023. vol. 27, no. 75 [consultado el 25, enero, 2025], p. 12-37. Disponible en Internet: <https://doi.org/10.14483/22487638.18268>. ISSN 2248-7638.

[49] GIRAL RAMÍREZ, Diego Armando, et al. Evaluación de modelos de programación lineal y no lineal para la planeación de sistemas de transmisión en el software GAMS. En: Tecnura [en línea]. 1, julio, 2021. vol. 25, no. 69 [consultado el 25, enero, 2025], p. 16-50. Disponible en Internet: <https://doi.org/10.14483/22487638.17957>. ISSN 2248-7638.

[50] ADAN GALLO, Johanna Catalina, et al. Nuevos desafíos de las pequeñas y medianas empresas en tiempos de pandemia. En: Tecnura [en línea]. 1, abril,

2022. vol. 26, no. 72 [consultado el 25, enero, 2025], p. 185-208. Disponible en Internet: <https://doi.org/10.14483/22487638.17879>. ISSN 2248-7638.

[51] ARIAS BATTLE, Yeraldin y ROA RODRÍGUEZ, Guillermo. Modelo en gerencia de proyectos para pymes de consultoría informática en Bogotá, basado en marcos ágiles de trabajo. En: Tecnura [en línea]. 25, septiembre, 2022. vol. 26, no. 74 [consultado el 24, enero, 2025], p. 49-69. Disponible en Internet: <https://doi.org/10.14483/22487638.17094>. ISSN 2248-7638.

[52] SÁNCHEZ CÉSPEDES, Juan Manuel; RODRÍGUEZ MIRANDA, Juan Pablo y RAMOS SANDOVAL, Olga Lucia. Decision Support Systems (DSS) Applied to the Formulation of Agricultural Public Policies. En: Tecnura [en línea]. 1, octubre, 2020. vol. 24, no. 66 [consultado el 24, enero, 2025], p. 95-108. Disponible en Internet: . ISSN 2248-7638.