



UNIVERSIDAD
NACIONAL
AUTÓNOMA DE
NICARAGUA,
MANAGUA
UNAN-MANAGUA

FACULTAD REGIONAL MULTIDISCIPLINARIA DE CHONTALES

Departamento de ciencia, tecnología y salud

Proyecto de graduación para optar al título de:

Ingeniero en sistema de la información

Título del proyecto:

Desarrollo de una aplicación web para el control de ventas y facturación en línea, para la farmacia veterinaria COOPA, en la ciudad de Acoyapa-Chontales, durante el segundo semestre del año 2020

Carrera:

Ingeniería en sistema de información

Autores:

Br. Jacqueline Margarita Dávila Oporta

Br. Keylin Danieska López Obando

Tutores:

M.Sc. Miriam patricia Téllez

M.Sc Yesenia Sobeyda Téllez

Juigalpa, Chontales, febrero 2021

¡A la libertad por la universidad!

VALORACIÓN DOCENTE

En los últimos años, ha revolucionado la forma de llegar a los clientes gracias a la tecnología informática y sus bondades en diferentes áreas del comercio electrónico. Las PYMES han crecido utilizando aplicaciones web y móviles que faciliten las compras personales sin fronteras, ni tiempos limitados y la internet se ha transformado como un canal de ventas efectivo.

Acoyapa, es una ciudad ganadera en crecimiento, donde la idea y “**Desarrollo de una aplicación web para el control de ventas y facturación en línea, para la farmacia veterinaria COOPA, en la ciudad de Acoyapa – Chontales, durante el segundo semestre del año 2020**”, es interesante, innovador y creativo dando solución a la fácil obtención de productos veterinarios con entrega rápida y segura.

El proyecto es oportuno y permite a los habitantes la migración, adaptación y satisfacción del uso de aplicaciones tecnológicas, que ayuden a la obtención de productos veterinarios en la puerta de su casa.

Es importante destacar, que en el documento se evidencia la metodología y estándares requeridos en el desarrollo de software para obtener un producto de calidad aceptable, y este proyecto cumple con los requerimientos establecidos para presentarse en la defensa de grado de la carrera. Considerando evidente que las estudiantes han logrado un aprendizaje integral por el estudio constante durante los cinco años de perseverar en la carrera de Ingeniería en Sistemas de Información.

Por lo antes expuesto, consideramos que las bachilleres **Jackqueline Margarita Dávila Oporta** y **Keylin Danieska López Obando**, reúnen los requisitos para la evaluación del jurado examinador en la defensa de trabajo de grado para optar al título de Ingeniero en Sistemas de Información.

M. Sc. Miriam Patricia Téllez Marín

M. Sc. Yesenia Sobeyda Téllez Gómez

Dedicatoria

El presente proyecto de graduación, está dedicado primeramente a Dios, por darme la sabiduría y fortaleza necesaria para poder culminar con éxito mi carrera, la palabra de Dios dice que toda sabiduría viene de parte de él, y fuera de él no podemos hacer nada.

Con todo el amor a mi Mamá, Reyna del Carmen Oporta, a mi Papá. Javier Salvador Dàvila Barrera, ellos han sido el pilar fundamental en toda mi vida y formación, sin duda alguna son los mejores padres.

A mis hermanos: Bryan Dàvila, Tatiana Dàvila, Jimmy Dàvila, que me han brindado su apoyo siempre, a mi novio que ha sido parte de este proceso y me ha motivado cada momento.

Br. Jackqueline Margarita Dàvila Oporta.

El señor es mi fortaleza y mi escudo; mi corazón en él confía; de él recibí ayuda. Mi corazón salta de alegría, y con cánticos le daré gracias. (Salmos 28:7)

A mi madre Bernarda Antonia Obando Galis, por inculcarme buenos valores y costumbres darme apoyo, a mi abuela Eduarda Galis Taleno, mis bisabuelos Graciela Taleno Leiva, Juan Andrés Galis, mis hermanos Erik Kanesky López Espinoza, Arlen Alexander López Espinoza, mi tío Mario Enrique López López quienes me ha motivado y apoyado en todo momento de vida quienes han puesto toda su confianza para lograr mi objetivo y aportando en mi formación profesional.

Br. Keylin Danieska López Obando

Agradecimiento

“Nadie puede llegar a la cima armado sólo de su talento. Dios da el talento; el trabajo transforma el talento en genio”. Anna Pavlova.

Agradecemos a Dios por darnos la vida y salud necesaria hasta este momento ya que sin él no podemos hacer nada, a nuestros padres por el apoyo brindado, por formar parte de esta bonita experiencia de vida, a todos nuestros maestros que acrisolaron nuestra formación profesional en la Universidad, en especial a nuestros tutores de proyecto de graduación, M.Sc. Yesenia Sobeyda Téllez y M.Sc. Miriam patricia Téllez, a nuestros hermanos y familiares que siempre nos han animado.

De igual manera a la universidad UNAN-MANAGUA–FAREM-CHONTALES, por habernos brindado la oportunidad de formarnos como profesionales y enriquecer nuestros conocimientos.

Resumen ejecutivo

El bajo rendimiento en el control de productos veterinarios debido a las malas prácticas, no maximiza las ventas, provoca pérdidas de medicamentos por vencimiento y no asegura la facturación a los clientes, generando reducción en las ganancias de los productos.

Se desarrolla la aplicación web “**Farma-Veterinaria**” para el control de ventas y facturación en línea, para la farmacia veterinaria COOPA, en la ciudad de Acoyapa-Chontales, durante el segundo semestre del año 2020.

Donde se propone una aplicación innovadora, que permita el aumento de las ventas en la farmacia, el detalles de las ventas realizada, control del Stock de los productos registrados, control de reportes de ventas y comunicación constante con el cliente, mediante un chat individual, para la farmacia veterinaria COOPA en Acoyapa Chontales, de igual manera sirve como herramienta para mejorar una posición competitiva en el mercado, trayendo beneficios como ahorro de tiempo, cuentas claras y conscientes, disminuyendo costes de personal.

Se ha implementado la metodología ágil SCRUM. El proceso de desarrollo de la aplicación web, se ha dividido en 4 Sprint con una duración de dos semanas cada uno, 25 historias de usuarios y 67 tareas.

La aplicación web está desarrollada y diseñada con el Framework de Angular 10, Nodejs, JavaScript como lenguaje de programación y la base de datos MariaDB. Esta tecnología permite ofrecerle al cliente un producto de calidad.

“Farma-Veterinaria” es una aplicación con diseño web adaptativo, que busca la correcta visualización en distintos dispositivos, además puede iniciar en navegadores como: Google Chrome, Firefox, Microsoft Edge.

Índice general

1. Generalidades del proyecto:	13
2. Solución:	13
3. Diseño o modelo:	14
3.1.1 Diseño de funcionamiento, usuario administrador	14
3.1.2. Usuario cliente	15
4. Infraestructura	16
4.1. Usuario cliente:	16
4.1.2. Usuario administrador:	19
5. Diseño de navegación.	21
5.1. Descripción del mapa de navegación del administrador	21
5.1.2 Descripción del mapa de navegación del cliente.	22
6. Diseño arquitectónico.	23
6.1. Diagrama de componente Nivel 0	23
6.1.2. Diagrama de componente nivel 1	24
6.1.2.3. Diagrama de componente nivel 2	25
6.1.2.3.4 Diagrama de componente nivel 3	27
7. Oportunidad en el mercado.	28
8. Propuesta valor.	29
9. Ciclo del proyecto	29
9.1. Justificación de la tecnología	29
10. Justificación de la metodología.	35
11. Metodología seleccionada	38
12. Selección de la base de dato	39
13. Historias de usuarios.	40
13.1. Usuario no registrado:	40
13.1.2. Usuario cliente	40
13.1.2.3. Usuario administrador:	40
Historia de usuario	40

14. Product Backlog:	50
14.1 Planificación inicial de las iteraciones	51
15. Sprint Backlog	52
16. Planificación del primer Sprint	52
16.1 Burn Down Chart	53
17. Planificación del segundo sprint	55
17.1. Burn Down Chart	56
17.1.2. Resultado del segundo sprint.	57
18. Planificación del tercer Sprint	59
18.1. Burn Down Chart	60
18.1.2. Resultado del tercer sprint	60
19. Planificación del cuarto sprint	62
19.1. Burn Down Chart	63
20. Presupuesto.	65
21. Visibilidad del producto	67
22. Marco lógico.	67
23. Matriz de marco lógico	68
24. Árbol del problema	73
25. Objetivo del proyecto	74
26. Árbol de análisis de la solución	75
27. Cronograma de actividades.	76
28. Bibliografía	78
Anexos	80
Modelo canvas	1
Base de datos	2
Manual técnico	3
Manual de usuario	4
Plan de pruebas	5

Índice de ilustraciones

Ilustración 1: Diseño o modelo.	14
Ilustración 2: Diseño o modelo	15
Ilustración 3: Infraestructura	16
Ilustración 4: Infraestructura	17
Ilustración 5: Infraestructura	18
Ilustración 6: Infraestructura	19
Ilustración 7: Infraestructura	20
Ilustración 8: Diseño de navegación	21
Ilustración 9: Mapa de navegación cliente	22
Ilustración 10: Diagrama de componente nivel 0	23
Ilustración 11: Diagrama de componente nivel 1	24
Ilustración 12: Diagrama de componente nivel 1	25
Ilustración 13: Diagrama de componente nivel 2	25
Ilustración 14: Diagrama de componente nivel 2	26
Ilustración 15: Diagrama de componente nivel 3	27
Ilustración 17: Ver Tienda de producto	54
Ilustración 18: Agregar al carrito de compra	54
Ilustración 19: Ver carrito de compra	54
Ilustración 20: Eliminar producto del carrito	54
Ilustración 21: Realizar pedido	55
Ilustración 22: Detalle de pedido	55
Ilustración 23: Gráfico del segundo Sprint	57
Ilustración 24: Pago en línea	57

Ilustración 25: PayPal	57
Ilustración 26: Cuenta en PayPal.....	57
Ilustración 27: Factura.....	57
Ilustración 28: Ver información de la farmacia	58
Ilustración 29: Registrar productos.....	58
Ilustración 30: Eliminar producto	58
Ilustración 31: Ver detalle de venta	58
Ilustración 32: Contáctenos	59
Ilustración 33: Gráfico del tercer Sprint.....	60
Ilustración 34: Editar producto.....	61
Ilustración 35: Existencia de producto	61
Ilustración 36: Restar existencia de producto.....	61
Ilustración 37: Reportes de ventas	61
Ilustración 38: Ver lista de producto en inventario	61
Ilustración 39: Buscar productos	62
Ilustración 40: Administrar estado	62
Ilustración 41: Gráfico del cuarto Sprint.....	63
Ilustración 42: Ver cliente.....	64
Ilustración 43: Inicio de sesión	64
Ilustración 44: Registrar cliente.....	64
Ilustración 45: Chat individual	64
Ilustración 46: Facturación	65
Ilustración 47: Árbol del problema.....	73
Ilustración 48: Árbol de objetivos.....	74
Ilustración 49: Árbol de análisis de soluciones	75

Índice de tablas

Tabla 1: Tecnología seleccionada.....	34
Tabla 2: selección de la metodología.....	38
Tabla 3: Selección de la base de datos	39
Tabla 4: Ver información de la farmacia	41
Tabla 5: Ver catálogo	42
Tabla 6: Registro de clientes.....	42
Tabla 7: Inicio de sesión.....	42
Tabla 8: Agregar al carrito de compra.....	43
Tabla 9: Ver carrito de compra	43
Tabla 10: Actualizar compra.....	43
Tabla 11: Eliminar producto del carrito	44
Tabla 12: Realizar pedido.....	44
Tabla 13: Pago en línea	44
Tabla 14: Guardar detalle pedido.....	45
Tabla 15: Chat-individual.....	45
Tabla 16: Contáctenos.....	45
Tabla 17: Registrar producto	46
Tabla 18: Editar producto	46
Tabla 19: Eliminar producto.....	46
Tabla 20: Buscar productos en inventarios.....	47
Tabla 21: Ver lista de productos en inventarios	47
Tabla 22: Reportes de ventas diarias.....	47
Tabla 23: Ver pedidos	48
Tabla 24: Administrar estado de pedido	48

Tabla 25: Ver clientes.....	48
Tabla 26: Existencia de producto.....	49
Tabla 27: Restar existencia de producto	49
Tabla 28: Restar existencia de producto	50
Tabla 29: Pila de producto.....	51
Tabla 30: Planificación inicial de las iteraciones	52
Tabla 31: Planificación del primer Sprint.....	53
Tabla 32: Planificación de segundo Sprint.....	56
Tabla 33: Planificación del tercer Sprint	60
Tabla 34: planificación del cuarto Sprint.....	63
Tabla 35: Presupuesto	66
Tabla 36: Presupuesto de pruebas de software	66
Tabla 37: Matriz de marco lógico	68
Tabla 38: Matriz de marco lógico	69
Tabla 39: Matriz de marco lógico	70
Tabla 40: Matriz de marco lógico	71
Tabla 41: Matriz de marco lógico	72

1. Generalidades del proyecto:

El proyecto es una aplicación web, para el control de venta y facturación de la farmacia veterinaria COOPA, que tiene como fin, automatizar los procesos manuales.

La app está dirigida para la farmacia veterinaria COOPA, sin embargo, tiene la flexibilidad de adaptarse a otros modelos de negocio con las mismas características, ya que es un proyecto desarrollado con tecnologías genéricas que se adecuan a la necesidad del cliente.

2. Solución:

“El incremento de animales productores de alimentos, junto con el gasto sanitario que estos suponen, hacen que el sector del medicamento veterinario este adquiriendo cada vez más peso en Nicaragua, ante esta situación, se nos plantea la duda de si los propietarios de las farmacias veterinarias están correctamente preparados e informados” [1, p. 20] .

La poca gestión de ventas de productos veterinarios y la inadecuada facturación de los productos vendidos, no solo afecta al propietario de la farmacia veterinaria, sino también al cliente que realiza compras en dicho establecimiento. La problemática, es que realizan procesos manuales como: ventas, reportes, y pequeñas facturas comerciales y en hojas de cálculo, el registro de los productos, esto conlleva a no tener un control detallado de los productos registrados y pérdida de tiempo, además la comunicación con el cliente solo se efectúa cuando el cliente llega a comprar a la farmacia.

De acuerdo a la problemática expuesta se considera inevitable dar solución a las necesidades que presenta la farmacia veterinaria, utilizando herramientas tecnológicas innovadoras que facilita la automatización de los procesos, como: gestión de venta, facturación, y control de los productos que tienen en la tienda, chat individual para la constante comunicación con el cliente, esto permite el ahorro de tiempo, y brinda la oportunidad de dar publicidad a los productos nuevos que existe en la farmacia veterinaria, la aplicación web generara reportes de ventas con los datos de los clientes, así como facturas que gestionen el control exacto de las ventas realizadas, proveyendo al propietario de la farmacia información valiosa para mejorar las toma de decisiones

La tecnología a utilizar para el desarrollo de la aplicación web es el framework de angular 10, Node.js, que permitirá el desarrollo de interfaz de usuarios y la base de datos MariaDB para el almacenamiento de los datos, como editor de código se utiliza Visual Studio Code,

leguaje de programación JavaScript. La metodología ágil a usar en el proyecto, es SCRUM ya que su objetivo es planificar y controlar el trabajo realizado.

3. Diseño o modelo: se representa de forma gráfica, el comportamiento que tiene el usuario con la aplicación web.

3.1.1 Diseño de funcionamiento, usuario administrador

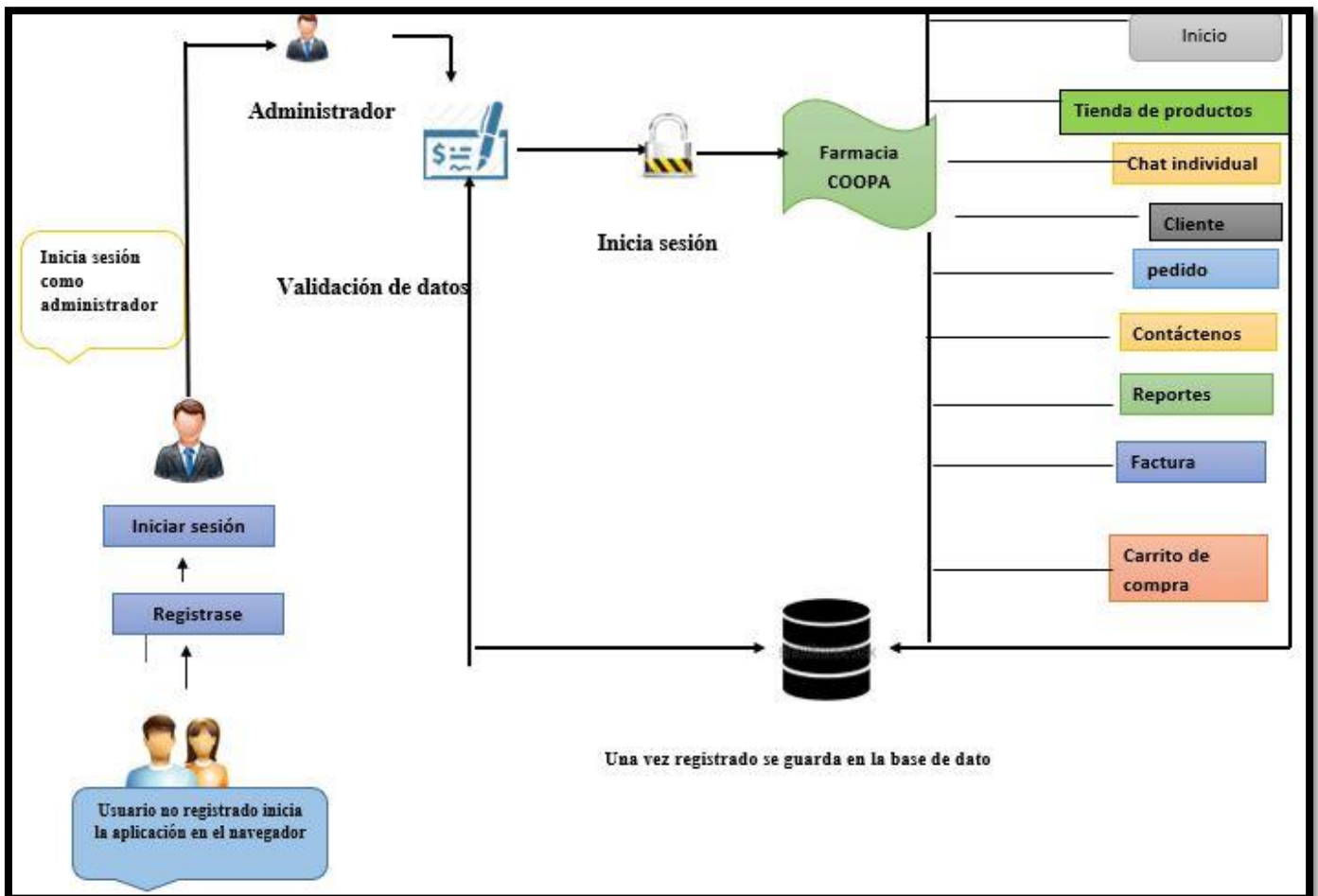


Ilustración 1: Diseño o modelo.

Elaboración propia

3.1.2. Usuario cliente

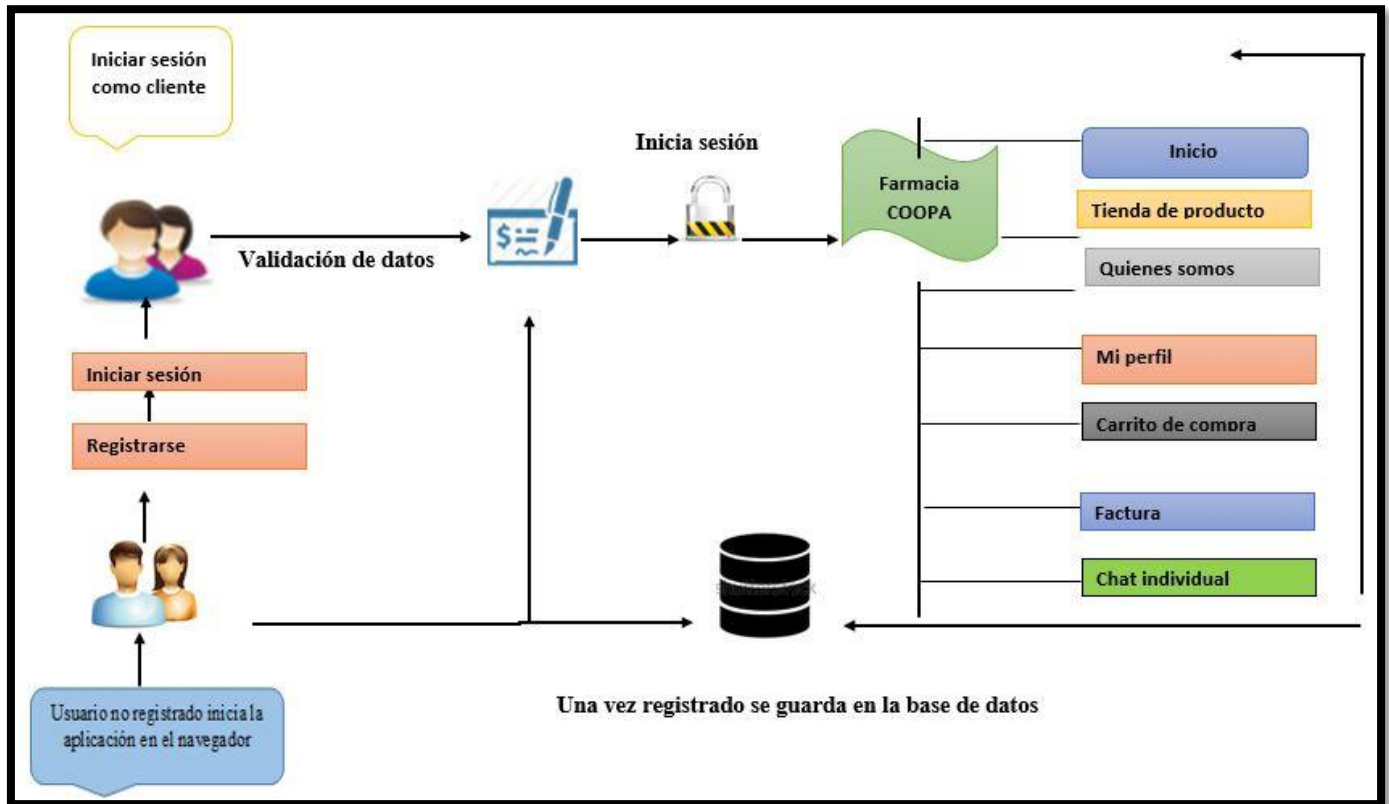


Ilustración 2: Diseño o modelo

Elaboración propia

4. Infraestructura

4.1. Usuario cliente: Representación de forma gráfica las pantallas que tiene la aplicación web.



Ilustración 3: Infraestructura

Elaboración propia



Ilustración 4: Infraestructura

Elaboración propia



Quienes somos

Contáctenos

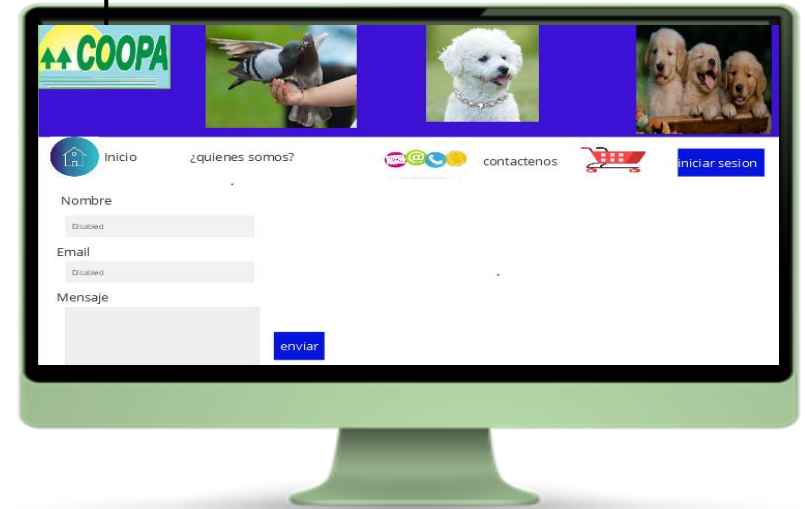


Ilustración 5: Infraestructura

Elaboración propia

4.1.2. Usuario administrador:

Muestra las vistas a las cuales tendrá acceso el usuario administrador dentro de la aplicación. Como las siguientes: inicio, quienes somos, contáctenos, productos, ventas, reporte de ventas, registro de usuarios y el chat

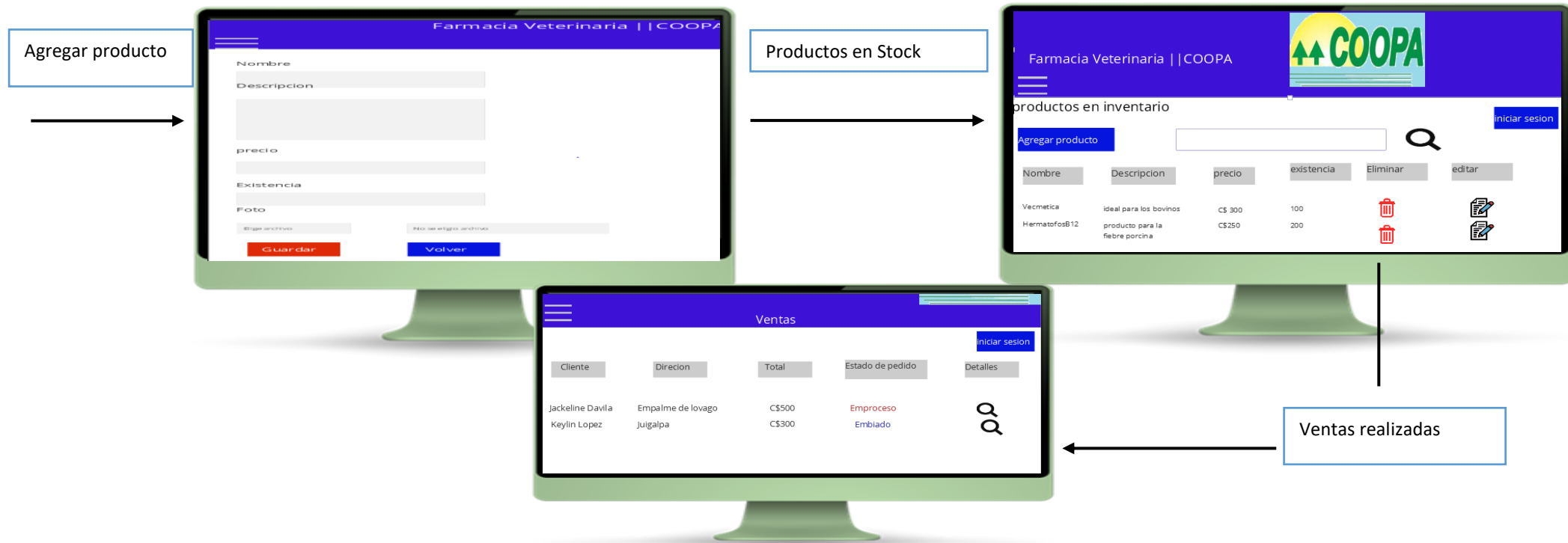


Ilustración 6: Infraestructura

Elaboración propia

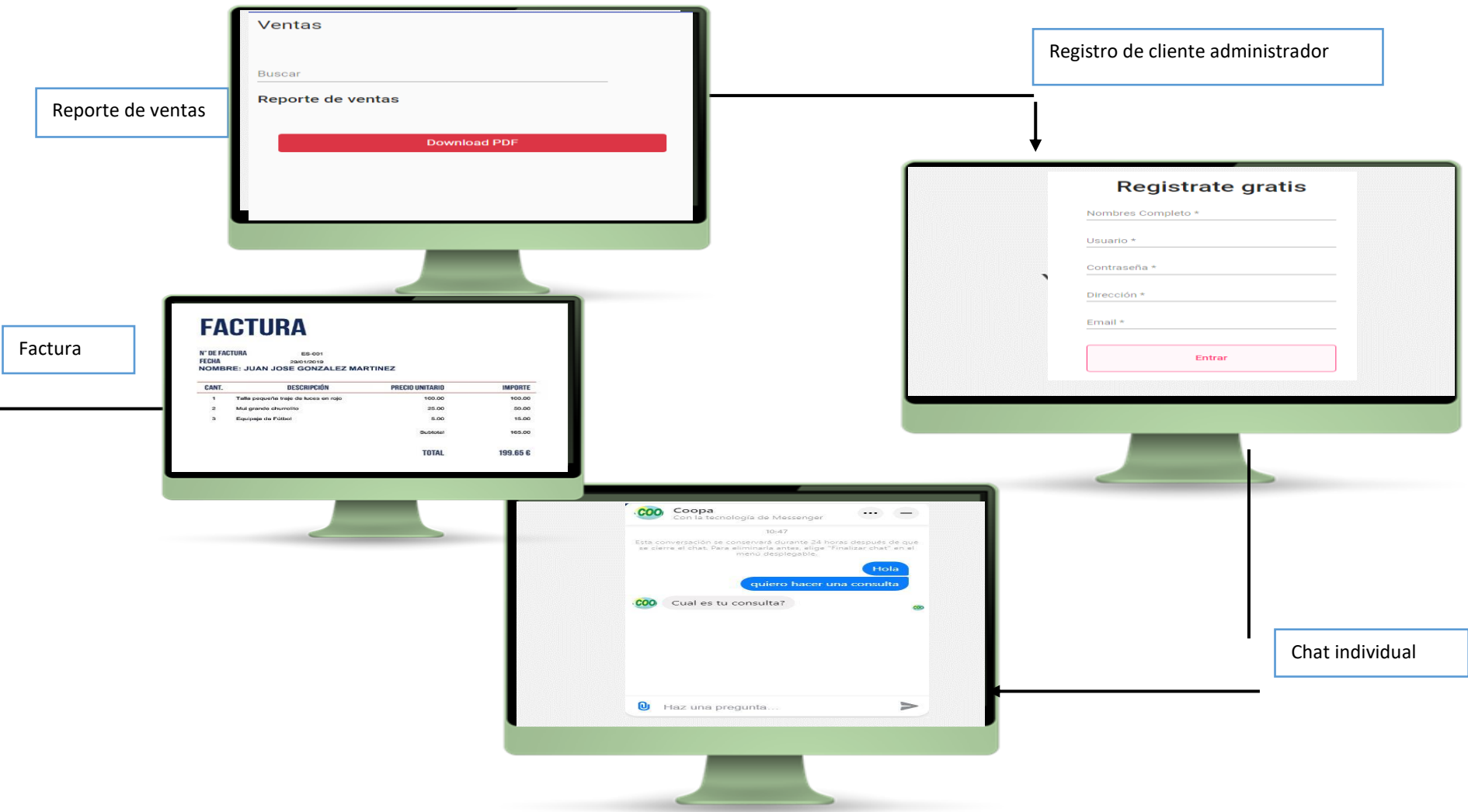


Ilustración 7: Infraestructura

Elaboración propia

5. Diseño de navegación.

5.1.Descripción del mapa de navegación del administrador

En este mapa se describe las funcionalidades que tiene el usuario administrador, como principal tendrá acceso a toda la aplicación y controlar las prioridades del sistema.

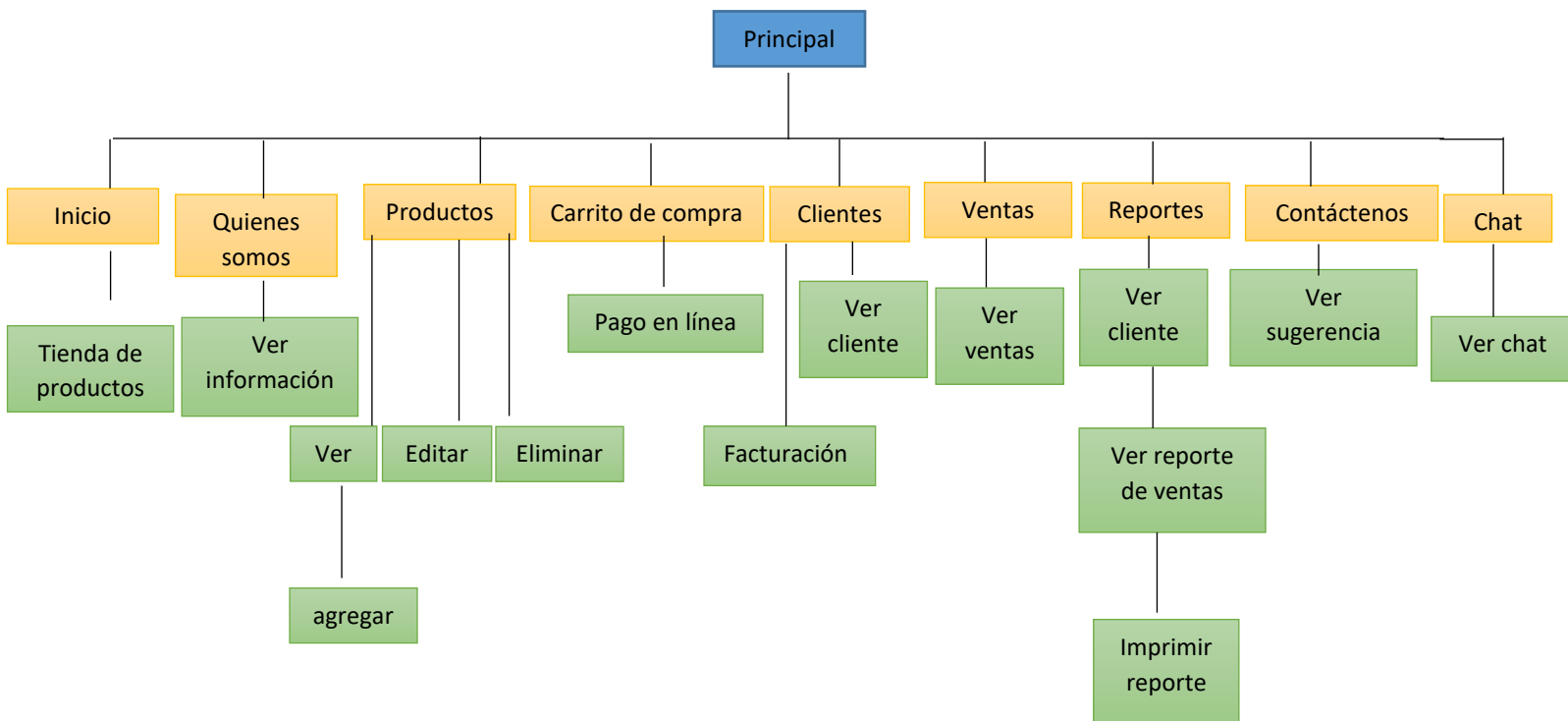


Ilustración 8: Diseño de navegación

Elaboración propia

5.1.2 Descripción del mapa de navegación del cliente.

El mapa describe las funcionalidades que tiene el usuario cliente.

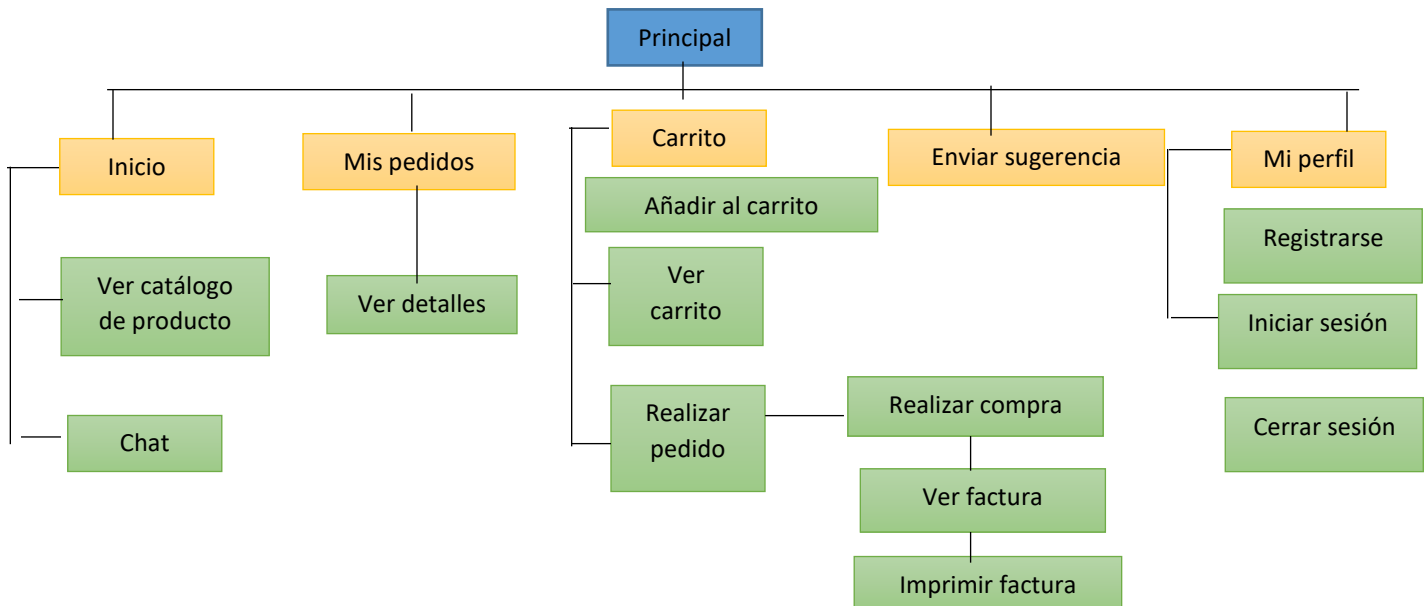


Ilustración 9: Mapa de navegación cliente

Elaboración propia

6. Diseño arquitectónico

Las siguientes imágenes representan el diseño arquitectónico de la infraestructura de la aplicación web Fam-veterinaria COOPA, a través de un diagrama de componente el cual describe el funcionamiento lógico, está estructurado en componente de interfaces y funcionalidades

6.1. Diagrama de componente Nivel 0

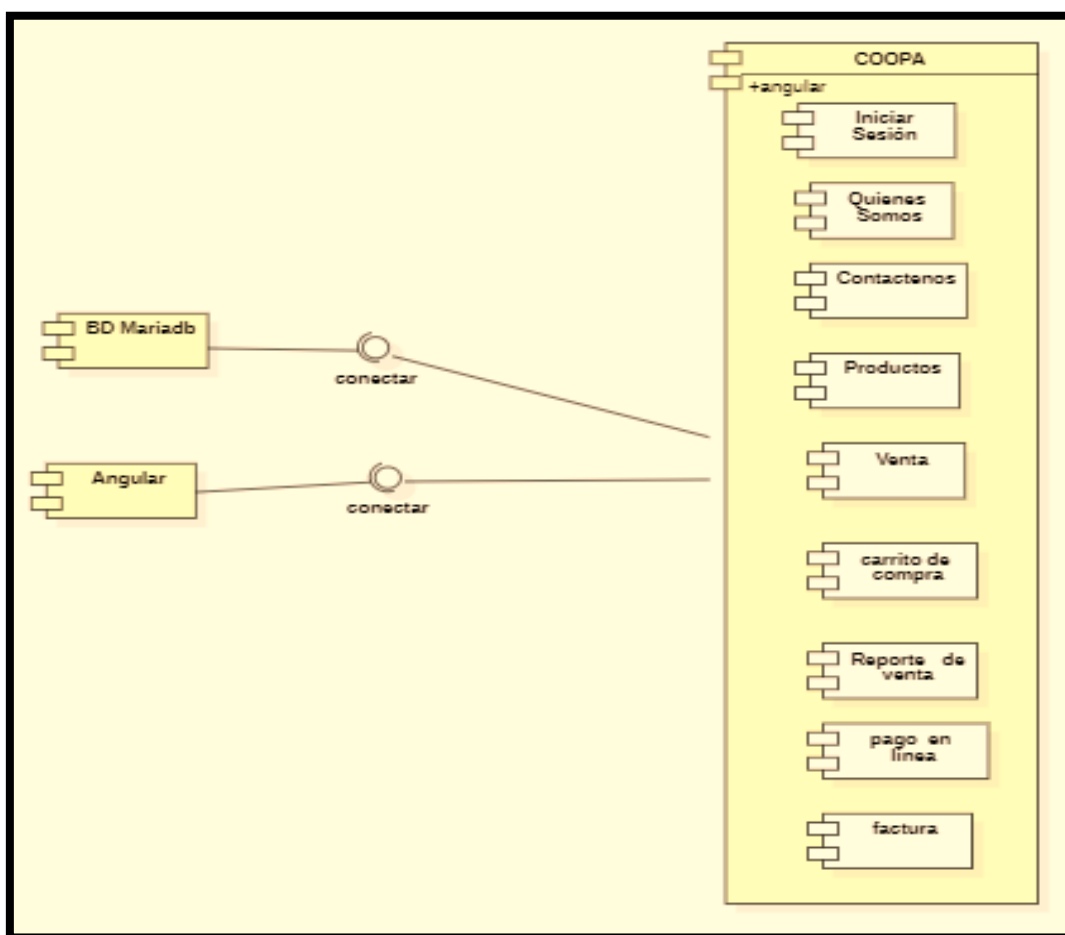


Ilustración 10: Diagrama de componente nivel 0

Elaboración propia

6.1.2. Diagrama de componente nivel 1

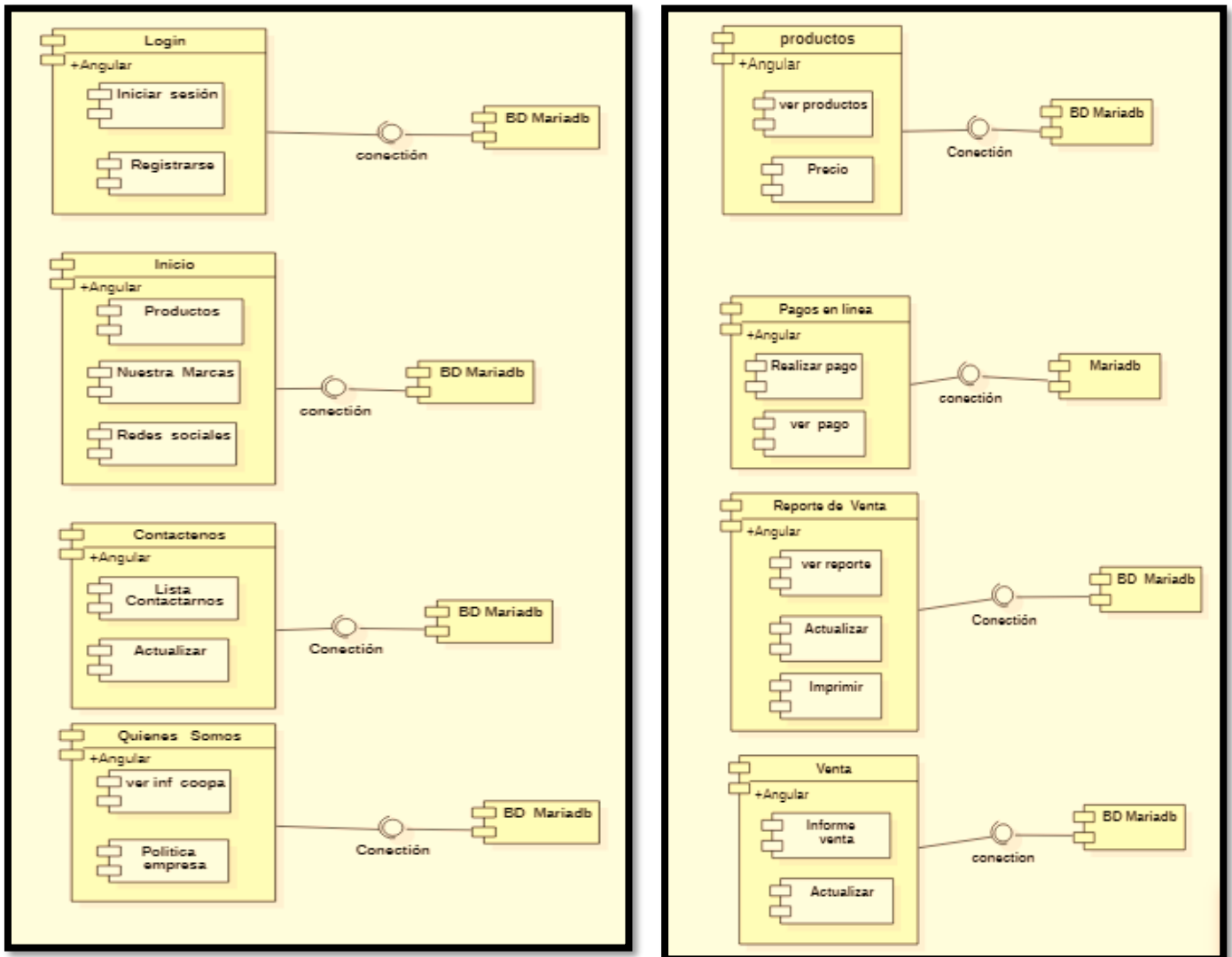


Ilustración 11: Diagrama de componente nivel 1

Elaboración propia

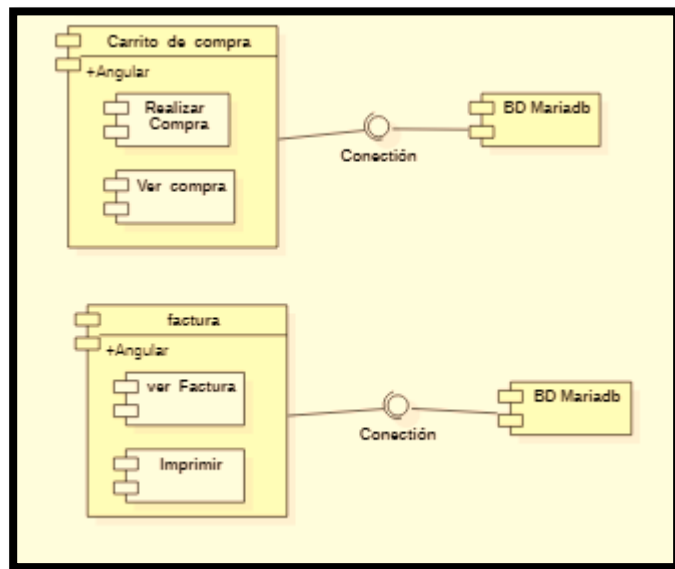


Ilustración 12: Diagrama de componente nivel 1

Elaboración propia

6.1.2.3. Diagrama de componente nivel 2

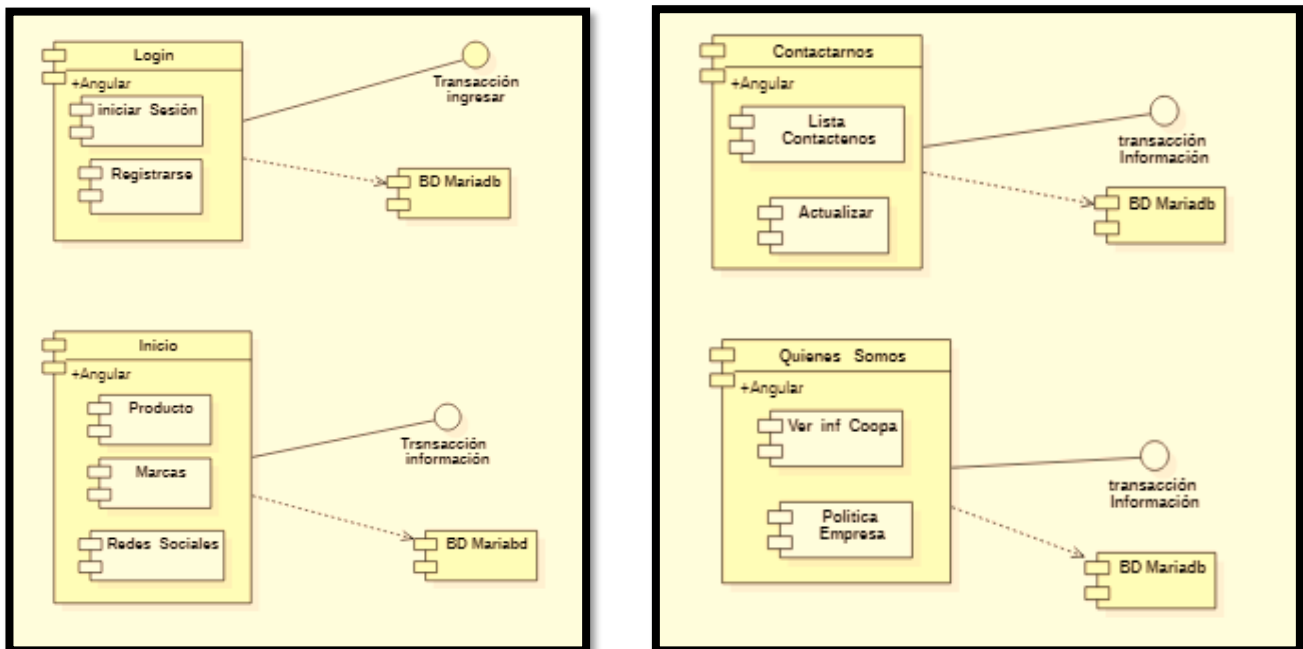


Ilustración 13: Diagrama de componente nivel 2

Elaboración propia

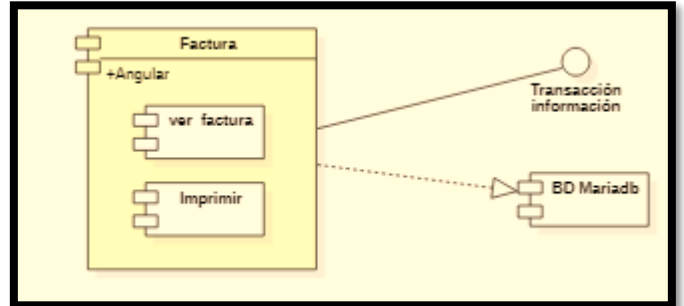
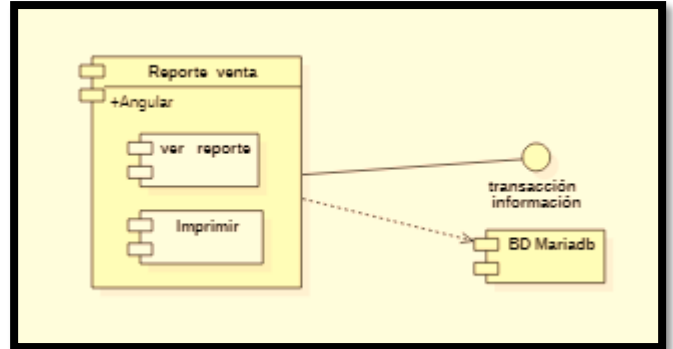
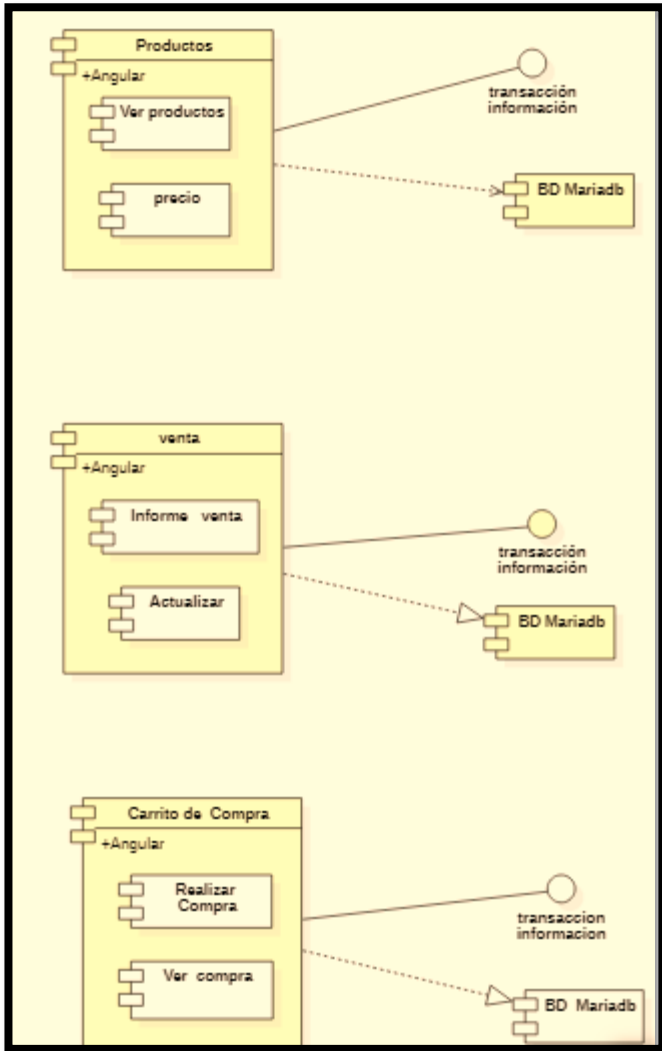


Ilustración 14: Diagrama de componente nivel 2

Elaboración propia

6.1.2.3.4 Diagrama de componente nivel 3

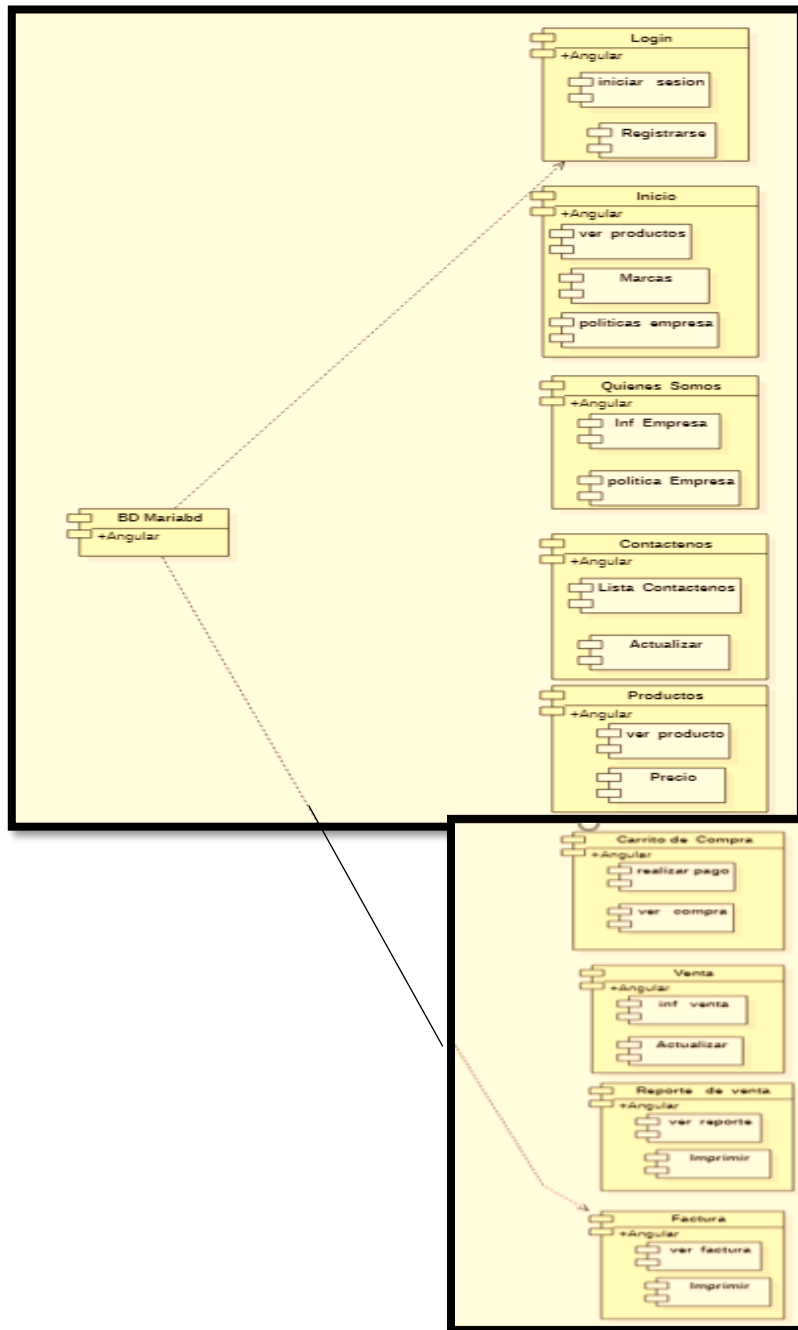


Ilustración 15: Diagrama de componente nivel 3

Elaboración propia

7. Oportunidad en el mercado

“En primer lugar se debe definir un marco que oriente, en la búsqueda de oportunidades, es necesario entender el rumbo de la empresa y conocer sus recursos, fortalezas y capacidades” [2, p. 17].

En el mundo de los negocios de hoy, las empresas no tienen garantizada la continuidad de su crecimiento y rentabilidad. Los avances tecnológicos acortan el ciclo de vida de los productos y servicios, los modelos de negocios cambian y nuevos competidores aparecen desde afuera de la industria. Esta constante inestabilidad exige buscar nuevas oportunidades de negocio. Es por esta razón que se ha desarrollado una aplicación web, para el control de ventas, y facturación de productos veterinarios.

Este proyecto está desarrollado específicamente para la farmacia veterinaria COOPA, siendo ellos el segmento consumidor, sin embargo, este software se ha diseñado de forma genérica y adaptable a otras empresas o negocios que presenten las mismas características de la aplicación web.

Los beneficios de facilidad y flexibilidad que brinda la tecnología son muchos y novedosos, por tanto, este sector puede ser beneficiado en gran manera, cambiando sus procesos tradicionales y migrando a la implementación de nuevas tecnologías en esta área.

Una aplicación web que mejora la gestión de ventas, y facturación, mantiene actualizado los registros de los clientes, los que pueden ser explotados con inteligencia de negocios para aumentar la venta de productos.

Por otra parte, se ha identificado un mercado, de acuerdo a los clientes potenciales que existen y la rentabilidad que tiene el software para que el cliente sea capaz de adquirir el producto. Considerando esto, se puede decir que la oportunidad de mercado en esta área es amplia.

8. Propuesta valor.

Con el modelo de negocio planteado en este documento, la farmacia COOPA podrá adquirir una aplicación web, que mejore el control de venta y facturación de productos veterinarios

La aplicación está diseñada con características innovadoras que resuelve los problemas de dicha farmacia, mejorando así la gestión de ventas realizadas, generando reportes de ventas de las facturas entregada al cliente, esto permite a los propietarios de la farmacia, tomar decisiones que ayuden a la economía de la empresa, además el cliente realizará sus pagos en línea.

Uno de los beneficios importante de esta aplicación es que cuenta con un área de chat individual, mejorando la comunicación con el cliente, de manera que se utiliza estrategias de negocio para poder ofrecerles a los clientes productos que brinda la farmacia.

9. Ciclo del proyecto.

9.1. Justificación de la tecnología

La tecnología está evolucionando a un ritmo acelerado y los cambios de ella con mucha rapidez, siendo esto un reto para todos los que utilizan las tic adecuada para desarrollar un proyecto.

“la sociedad actual se mueve como nunca antes había ocurrido alrededor de la tecnología, la sociedad por tanto se ve altamente influenciada por la nueva tecnología emergente, existiendo una estrecha relación entre la sociedad y la tecnología” [3, p. 46]

Para el desarrollo de este proyecto se necesita una metodología que agilice cada uno de los procesos de las tareas a ejecutar, de esta manera se podrá llevar a cabo los objetivos y escoger de manera correcta la tecnología que posee la menor curva de aprendizaje para poder realizar un software cien por ciento funcional.

La aplicación está diseñada en plataformas web, es por ello que se seleccionó la plataforma y framework Angular, manejada para escribir aplicaciones web en HTML (Hiper Text Markup Lenguaje), TypeScript, como editor de texto Visual Studio Code, y para el proceso de gestión de datos MariaDB.

Para la selección de la tecnología se tomaron en cuenta las siguientes características

- Experiencia con el Framework Angular
- Curva de aprendizaje (el grado de aprendizaje obtenido en el transcurso de tiempo)

Otra tecnología seleccionada para realizar las pruebas de la aplicación con el fin de comprobar el correcto funcionamiento del código, son Jasmine y Karma, Jasmine es un marco de prueba de JavaScript y Karma es una herramienta que permite ejecutar prueba de Jasmine, desde la línea de comando.

La utilización de diversas tecnologías nos permite desarrollar una aplicación totalmente funcional.

En la siguiente tabla comparativa se muestran las diferentes tecnologías, detallado los conocimientos técnicos, su ponderación, ventaja y desventajas, de manera que es por ello que se ha seleccionado las mejores herramientas tecnológicas para el desarrollo de la aplicación.

Al observar la siguiente tabla se notan las diferentes tecnologías con las que se pueden implementar la aplicación web Farm-veterinaria, para la farmacia COOPA, y cual de ella es la más adecuada para desarrollo, se realizaron comparaciones y se define que la tecnología del Framework de Angular es la mejor opción para poner en marcha el proyecto, ya que brinda una serie de herramientas que facilitan al desarrollador realizar trabajos un poco más rápido.

Tecnologías para el desarrollo de la aplicaciones web

Nombre		Licencia			Conocimientos técnicos requeridos			Ponderación		Descripción	ventajas	Desventajas	Cumplimientos de características
		lenguaje	experiencia	Curva aprendizaje	Experiencia	Curva aprendizaje							
PhoneGap	Open Source	JavaScript	40%	60%	24%	76%	Es un framework que permite desarrollar aplicaciones hibrida mediante tecnología web, HTML, CSS3 y JavaScript [4].	Facilidad de desarrollo	Mala práctica.	Aplicación nativa	Inconsistencia	30%	
		ECMAScript	0%	100%									
		HTML	40%	60%									
		CSS	40%	60%									
		Apache Cordova	0%	100%									
Ionic	Open Source	JavaScript	30%	70%	24%	76%	Es un framework que facilita la el desarrollo de aplicaciones móviles con tecnología web [6]	Entorno compatible para todas las plataformas	El rendimiento puede ser ligeramente menor que las desarrolladas de formas nativas	Framework libre	Es una herramienta nueva y puede ser difícil encontrar módulos compartidos por los usuarios [7]	50%	
		CMAScript	0%	100%									
		HTML	30%	70%									
		CSS	30%	70%									
		Angular	30%	70%									
React native	Open Source	JavaScript	40%	100%	20%	80%	Es un framework de JavaScript para crear aplicaciones	Genera modelo de objeto del documento	Inmadurez de la plataforma	Otro problema es los crashes, lo que	20%		
		ECMAScript	20%	80%									

		HTML	20%	80%			reales nativas IOS y Android, basada en la librería de JavaScript [8]	Aplica comodidad Compuesto por componentes [9]	significa que la aplicación se cierra de manera inesperada [9]	
		CSS	30%	70%						
		JavaScript	40%	70%						

Nombre	Licencia	Conocimientos técnicos requeridos			Ponderación		Descripción	ventajas	Desventajas	Cumplimientos de características
		lenguaje	experiencia	Curva aprendizaje	Experiencia	Curva aprendizaje				
jQuery Mobile	MIT	JavaScript	40%	60%	22%	78%	Es un Framework javaScript para el desarrollo rápido y fácil de sitios webs optimizados para teléfonos móviles. Con este framework, aceleramos la velocidad de desarrollo de aplicaciones [10]	Temas personalizados Tamaños reducidos Facilidad de uso Multi plataformas Soporte HTML5 [11]		20%
		HTML	40%	60%						
		CSS	30%	70%						
		ECMAScript	0%	100%						
		JavaScript	0%	100%						
Framework 7	MIT	JavaScript	40%	60%	22%	78%	Es un framework de HTML, para desarrollar aplicaciones IOS y Android es una de las librería y soluciones de código abierto [12]	Facilita la funcionalidad en diferentes navegadores Permite trabajar con layout simétrica Refuerza buenos hábitos de diseño [13]	Limita tu libertad Añade código extra Te fuerza a usar la semántica propia de framework Podría ser una pérdida de tiempo. [14]	30%
		HTML	40%	60%						
		CSS	30%	70%						
		ECMAScript	0%	100%						
		JavaScript	40%	100%						
Angular	Open Source	JavaScript	40%	60%	50%	50%	Angular es un framework Javascript	Ahorra tiempo	Difícil de manejar la nomenclatura	50%

		CMAScript	0%	100%			potente, muy adecuado para el desarrollo de aplicaciones frontend modernas, de complejidad media o elevada. [15]	Usa lenguaje TypeScript Las aplicaciones son fáciles de mantener [16]	Falta de documentación [17]	
		HTML	80%	20%						
		CSS	80%	20%						
		JavaScript	40%	60%						

Tabla 1: Tecnología seleccionada

Elaboración propia.

10. Justificación de la metodología

Con el paso de los años el desarrollo de software ha ido evolucionando, cada vez es más común que las empresas, a la hora de abordar el desarrollo de nuevos productos, obtén por aplicar la llamada metodología ágiles [18]

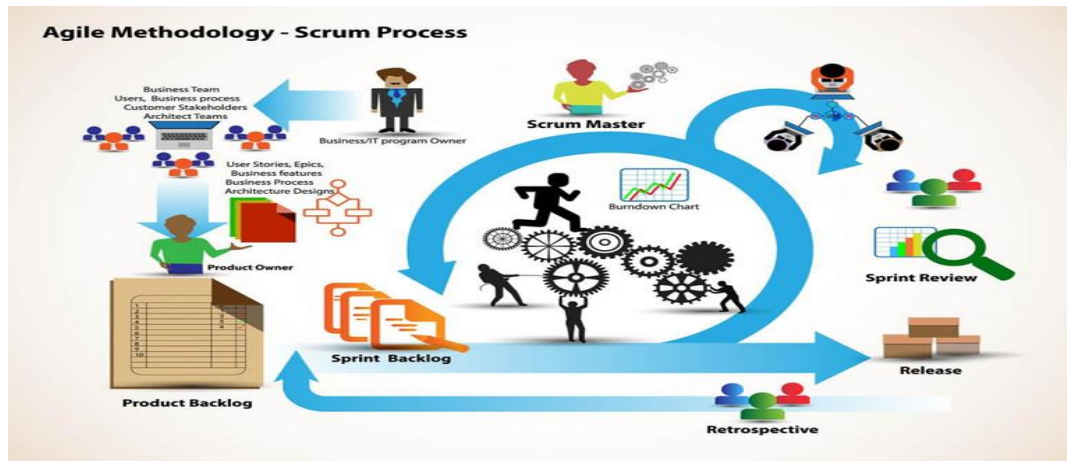
Hoy en día la gestión de procesos y equipos es un valor fundamental para entender una organización con gran influencia en los resultados del negocio. Muchas empresas manejan departamentos como si fueran partes independientes, sin tener en cuenta que una organización es un todo y no solo la suma de sus partes, lo que conlleva a errores de planificación, conflictos, retraso en los proyectos, sobre costes que influyen en la desmotivación de los equipos como consecuencia de una mala gestión

Como contrapartida a las metodologías de gestión de proyectos tradicionales, en donde los métodos de trabajo son muy formales, y conllevan a realizar una gran carga de trabajo de gestión generando cantidades de documentación, es por ello que surge las metodologías ágiles, es un nuevo sistema de gestión que se basa en dar respuesta a los problemas con los que se encuentran las metodologías tradicionales.

Son metodología centradas en la interacción, comunicación y en la reducción de elementos intermedios, fomentando la comunicación entre los miembros del equipo.

Para el desarrollo de la aplicación web se necesita una metodología que agilice cada uno de los procesos de las tareas a ejecutar.

SCRUM es actualmente la metodología ágil más popular. Se basa principalmente en la premisa de ejecutar un proyecto en entregas parciales y regulares del producto. El desarrollo del producto se realiza de forma incremental y evolutiva, lo que resulta ideal en el entorno dinámico y cambiante. [19]



Como se observa en la imagen, SCRUM nos proporciona un marco de trabajo para soportar la innovación y permitir que equipo auto-organizados entreguen resultados de alta calidad en tiempo corto.

En un equipo SCRUM intervienen varios perfiles

- **El product Owner:** es la persona encargada del éxito del producto desde el punto de vista de los clientes. Define los objetivos del proyecto y se focaliza en maximizar la rentabilidad del producto. [20]
- **El Scrum Master:** es el Coach del resto del equipo y quien le ayuda a alcanzar su máximo nivel de productividad para garantizar el cumplimiento del objetivo [21].
- **El Scrum Team:** está formado por todos los individuos necesarios para la construcción del producto en cuestión. [22]
- **Stakeholders:** son las personas a las que el proyecto les producirá algún beneficio (comerciante, dueños, directores)

Utilizar SCRUM puede ser una gran ventaja competitiva. Ya que el equipo sabe lo que tiene que hacer cada día, si algo va mal, se reunirán, y el equipo va mejorando la manera de trabajo

Dentro de los beneficios que podemos mencionar al utilizar la metodología SCRUM son las siguientes.

- Fomenta la motivación y el compromiso del equipo.
- La organización horizontal promueve la autonomía y la auto-organización.
- El método de trabajo y la revisión continua produce un mayor valor de negocio, lo que facilita el retorno de la inversión.

En la siguiente tabla se puede observar las comparaciones entre las metodologías ágiles y las metodologías tradicionales lo que permite identificar que las metodologías ágiles son las más adecuadas para el desarrollo de aplicaciones en corto tiempo además que son fáciles de utilizar. De manera que al compararlas se puede definir que la metodología SCRUM es la más adecuada para el desarrollo de la aplicación web, tomando en cuenta que el equipo de trabajo tiene experiencia en el uso de la metodología.

11. Metodología seleccionada

metodología	características	N.º de personas	Presupuesto	tiempo	Conocimiento extra necesario	Curva de aprendizaje	Documentación generada
SCRUM	<p>La colaboración ya que es una de las mejores cosas que tiene SCRUM</p> <p>La auto organización porque SCRUM está muy enfocado en los equipos.</p> <p>La priorización, debido a que existen prioridades y se establecen criterios [23]</p>	5-9	8 desarrolladores	7 a 30 días	No aplica	El equipo de trabajo ya ha utilizado esta metodología	Product backlog, Sprint backlog, Burndown chart
Kanban	<p>Crea un flujo continuo e interrumpido.</p> <p>Circuito de retroalimentación [24]</p>	5 a más	Se necesita analizar variables de tiempo, costo para poder abordar el presupuesto	El periodo es entre la primera tarea y el flujo de trabajo y la salida final del sistema	No aplica	Entender muy bien el funcionamiento del producto para luego poder mejorarlo	Tarjeta de fabricación, Tarjetas de transporte de proveedores, Kanban de emergencia, Kanban orden de trabajo, Knanban de etiqueta

Tabla 2: selección de la metodología

Elaboración propia

12. Selección de la base de dato

Para la selección de la base de datos fue necesario crear un cuadro comparativo donde se establecieran los parámetros a evaluar, tomando en cuenta las necesidades del proyecto establecida por el equipo de desarrollo.

Base de dato	Tipo de dato	soporte	experiencia	almacenamiento
Couchbas e lite	documentos	Android, IOS	no	Local
LevelDB	Pares clave-valor	Android, IOS	no	Local
firebase	No relacional	Multi plataforma web, móvil.	no	En la nube
MariaDB	relacional	multiplataforma	si	local

Tabla 3: Selección de la base de datos

Elaboración propia

En el desarrollo del proyecto son necesarias tecnología con licencias de código abierto, ya que permite reducir el costo de desarrollo de la aplicación, se toma en cuenta la experiencia del equipo y las ventajas que tiene utilizar la base de datos seleccionada.

La base de dato seleccionada fue MariaDB ya que es un software de open source cuyas características muchas veces se desconocen. MariaDB mejora el rendimiento que otras alternativas más conocidas y utilizadas tradicionalmente.

A demás existen novedades que han sido implementados en MariaDB dentro de las cuales podemos destacar:

- Nuevos motores de almacenamientos como Área
- Nuevas características disponibles, relacionada directamente con las características disponible en base de datos NoSQL
- Nueva gestión de conexión con la base de datos que permite multiplicar el número de acceso de manera concurrente.

13. Historias de usuarios.

En las siguientes tablas se muestran a detalle las historias de usuarios definida para el cumplimiento de desarrollo de la aplicación web para la farmacia veterinaria COOPA. En este sentido se puede hablar de los diferentes usuarios de la aplicación web.

13.1. Usuario no registrado: el usuario podrá tener acceso dentro de la aplicación web a ver el catálogo de los productos que ofrece la empresa, y puede conocer más de la farmacia veterinaria con la información que genera la aplicación.

13.1.2. Usuario cliente: como usuario cliente, podrán registrarse, iniciar sesión, agregar producto al carrito de compra, así como, eliminar y actualizar, ver las compras realizadas, ver catálogo de los productos, enviar mensajes a la farmacia para poder tener un mejor servicio, realizar sus pagos en línea.

13.1.2.3. Usuario administrador: el usuario administrador podrá registrar producto en inventario, eliminar, actualizar, ver los pedidos entregados, pendientes, y en proceso, podrá tener la opción de ver los reportes y factura de los productos, ver mensajes.

Historia de usuario

Son la descripción de las funcionalidades que va a tener el software, las cuales serán el resultado de la colaboración entre el cliente y el equipo e irán evolucionando durante toda la vida del proyecto.

Las historias de usuarios son pequeñas descripciones de los requerimientos de un cliente. Su utilización es común cuando se aplica marcos de entornos ágiles como SCRUM.

Permite describir de manera breve los requisitos funcionales del sistema. El siguiente esquema muestra el formato a aplicar.

Historia de usuario	
ID	
Nombre de historia:	
Como:	
Quiero:	
Para poder:	
Validación:	Valor:
	Prioridad:
	Estimación:

Id: es el número de secuencia de la historia de usuario

Nombre: texto que describe el nombre de la historia de usuario.

Como: describe el tipo de usuario

Quiero: es el detalle propio de la historia de usuario, la documentación del requisito.

Para poder: describe lo que va a alcanzar a realizar.

Validación: criterios o acciones que realiza el usuario

Valor: es la escala cuantitativa con respecto al valor que tiene la historia, 200, 100, 150, 50. Siendo 200 el de mayor valor y 50 el de menor.

Prioridad: es la escala cuantitativa que describe el nivel de importancia de la historia de usuario en este caso 1 es de menor prioridad y 5 es el de mayor prioridad.

Estimación: es el tiempo en que se realiza la historia de usuario

Historia de usuario	
ID.	1
Nombre de historia:	Ver información de la farmacia veterinaria
Como:	Usuario no registrado
Quiero:	Ver la información general de la empresa como su ubicación y cómo contactarme con ella
Para poder:	Tener un amplio conocimiento de la farmacia veterinaria
Validación: ✓ La información se presenta de manera ordenada y completa.	Valor: 50
	Prioridad: 1
	Estimación: 8 h

Tabla 4: Ver información de la farmacia

Elaboración propia

Historia de usuario	
ID.	2
Nombre de historia:	Ver tienda de producto
Como:	Usuario no registrado/cliente
Quiero:	Ver tienda de productos
Para poder:	Conocer los productos que ofrece la aplicación.
Validación: ✓ Validar que los datos de los productos son correctos	Valor: 150
	Prioridad: 4
	Estimación: 12 h

Tabla 5: Ver catálogo

Elaboración propia

Historia de usuario	
ID.	3
Nombre de historia:	Registrar cliente
Como:	Usuario no registrado
Quiero:	Registrarme en la aplicación
Para poder:	Realizar pedidos en la aplicación, agregar al carrito de compra, registrarme y realizar pagos en línea
Validación: ✓ Registrarse con los datos necesarios para la validación	Valor: 50
	Prioridad: 1
	Estimación: 18

Tabla 6: Registro de clientes

Elaboración propia

Historia de usuario	
ID.	4
Nombre de historia:	Inicio de sesión
Como:	cliente
Quiero:	Iniciar sesión
Para poder:	Tener una información amplia de los productos ofertados en la aplicación, realizar compras, pagos, detalle de mis pedidos, comunicación constante con fa farmacia veterinaria COOPA
Validación: ✓ Verificar que las credenciales del usuario sean las correctas	Valor: 50
	Prioridad: 1
	Estimación: 10h

Tabla 7: Inicio de sesión

Elaboración propia

Historia de usuario	
ID.	5
Nombre de historia:	Agregar al carrito de compra
Como:	Cliente
Quiero:	Agregar producto al carrito de compra
Para poder:	Realizar pedido.
Validación: ✓ Verificar que las compras se han guardado correctamente en la aplicación.	Valor: 200
	Prioridad: 5
	Estimación: 20h

Tabla 8: Agregar al carrito de compra

Elaboración propia.

Historia de usuario	
ID.	6
Nombre de historia:	Ver carrito de compra
Como:	cliente
Quiero:	Ver productos seleccionados para compra
Para poder:	Tener control de mis pedidos, detalles de mis pedidos
Validación: ✓ Visualizar productos y cantidades seleccionadas en la aplicación para validar la información	Valor: 200
	Prioridad: 5
	Estimación: 16h

Tabla 9: Ver carrito de compra

Elaboración propia

Historia de usuario	
ID.	7
Nombre de historia:	Actualizar compra
Como:	Cliente
Quiero:	Aumentar o disminuir compras seleccionadas
Para poder:	Obtener producto y cantidad deseada
Validación: Ver que el número mínimo sea 1 y máximo menor a las existencias	Valor: 150
	Prioridad: 4
	Estimación: 12h

Tabla 10: Actualizar compra

Elaboración propia

Historia de usuario	
ID.	8
Nombre de historia:	Eliminar producto del carrito
Como:	Cliente
Quiero:	Eliminar producto del carrito
Para poder:	Ordenar mis pedidos
Validación:	Valor: 150
✓ Eliminar producto del carrito	Prioridad: 4
✓ actualizar el total de la factura	Estimación: 14h

Tabla 11: Eliminar producto del carrito

Elaboración propia

Historia de usuario	
ID.	9
Nombre de historia:	Realizar pedido
Como:	Cliente
Quiero:	Realizar mis pedidos en la aplicación
Para poder:	Obtener mi producto a domicilio
Validación:	Valor: 200
✓ actualizar el carrito de compra	Prioridad: 5
	Estimación: 20h

Tabla 12: Realizar pedido

Elaboración propia

Historia de usuario	
ID.	10
Nombre de historia:	Pago en línea
Como:	Cliente
Quiero:	Realizar mis pedidos en la farmacia veterinaria.
Para poder:	Obtener mi producto a domicilio
Validación:	Valor: 200
Mostrar un botón que verifique el monto a pagar y que este corresponda al total a pagar	Prioridad: 5
	Estimación: 20h

Tabla 13: Pago en línea

Elaboración propia

Historia de usuario	
ID.	11
Nombre de historia:	ver detalle pedido
Como:	Cliente
Quiero:	Guardar pedido
Para poder:	Estar seguro de las compras realizadas
Validación:	Valor: 150
✓ Mostar una opción que diga guardar pedidos y ver los cambios	Prioridad: 4
✓ Mostrar factura	Estimación: 13h

Tabla 14: Guardar detalle pedido

Elaboración propia

Historia de usuario	
ID.	12
Nombre de historia:	Chat-individual
Como:	Cliente
Quiero:	Enviar mensaje a la farmacia veterinaria
Para poder:	Obtener una mayor información de los productos y servicios que ofrecen y comunicación con la empresa
Validación:	Valor: 100
Recibir mensaje	Prioridad: 5
Enviar mensaje	Estimación: 15h
Ver mensaje	

Tabla 15: Chat-individual

Elaboración propia

Historia de usuario	
ID.	13
Nombre de historia:	Contáctenos
Como:	Cliente
Quiero:	Comunicarme con la empresa
Para poder:	Tener mayor información de la farmacia veterinaria
Validación:	Valor: 50
✓ Verificar que el apartado contáctenos funcione correctamente	Prioridad: 1
	Estimación: 15h

Tabla 16: Contáctenos

Elaboración propia

Historia de usuario	
ID.	14
Nombre de historia:	registrar productos
Como:	administrador
Quiero:	Registrar productos de la farmacia
Para poder:	Ofrecer al cliente una amplia variedad de productos
Validación: <ul style="list-style-type: none"> ✓ Mostrar un botón que tenga la opción de registrar producto. ✓ Verificar que el producto veterinario sea registrado correctamente. 	Valor: 150
	Prioridad: 2
	Estimación: 20h

Tabla 17: Registrar producto

Elaboración propia

Historia de usuario	
ID.	15
Nombre de historia:	editar producto
Como:	Usuario administrador
Quiero:	Actualizar producto como: nombre, precio existencia, descripción, imagen, fecha de vencimiento
Para poder:	Ofrecer productos nuevos
Validación: <ul style="list-style-type: none"> ✓ Muestra un botón editar y edita los campos establecido en el formulario 	Valor: 150
	Prioridad: 2
	Estimación: 20h

Tabla 18: Editar producto

Elaboración propia

Historia de usuario	
ID.	16
Nombre de historia:	Eliminar producto
Como:	Usuario administrador
Quiero:	Eliminar producto del inventario de la farmacia
Para poder:	Ofrecer una mejor calidad de servicio al cliente
Validación: Validar que el campo eliminar producto realice los cambios correctos	Valor: 150
	Prioridad: 2
	Estimación: 20h

Tabla 19: Eliminar producto

Elaboración propia

Historia de usuario	
ID.	17
Nombre de historia:	Buscar productos en stock
Como:	administrador
Quiero:	Buscar productos
Para poder:	Analizar cómo va el inventario
Validación: ✓ Buscar en inventario el producto deseado por filtros ya sea por nombre o fecha de registro	Valor: 100
	Prioridad: 3
	Estimación: 20h

Tabla 20: Buscar productos en inventarios

Elaboración propia

Historia de usuario	
ID.	18
Nombre de historia:	Ver lista de producto de inventario
Como:	administrador
Quiero:	Listar los productos en inventario
Para poder:	Obtener una mayor claridad de todos los productos en inventarios
Validación: ✓ Listar los productos en inventario por orden de fecha	Valor: 100
	Prioridad: 2
	Estimación: 10h

Tabla 21: Ver lista de productos en inventarios

Elaboración propia

Historia de usuario	
ID.	19
Nombre de historia:	Reporte de venta diarias
Como:	administrador
Quiero:	Realizar el reporte de las ventas
Para poder:	Analizar cómo van las ventas en la farmacia
Validación: ✓ Realizar filtros de reporte de ventas por mes ✓ Verificar que los datos sean los correctos	Valor: 100
	Prioridad: 3
	Estimación: 20 h

Tabla 22: Reportes de ventas diarias

Elaboración propia

Historia de usuario	
ID.	20
Nombre de historia:	Detalle de venta
Como:	Administrador
Quiero:	Ver pedido que realizan los clientes en la aplicación
Para poder:	Entregar los pedidos en fecha establecida
Validación: <ul style="list-style-type: none"> ✓ Ver listas de pedidos de los clientes ✓ actualizar estado del pedido si fue entregado, pendiente o en proceso 	Valor: 100
	Prioridad: 3
	Estimación: 10 h

Tabla 23: Ver pedidos

Elaboración propia

Historia de usuario	
ID.	21
Nombre de historia:	Administrar estado de pedido
Como:	administrador
Quiero:	Gestionar el estado de los pedidos que realizan los clientes
Para poder:	Tener un mejor manejo de pedidos
Validación: <ul style="list-style-type: none"> ✓ Habilitar estado de pedido: pendiente, en proceso, entregado 	Valor: 100
	Prioridad: 2
	Estimación: 10 h

Tabla 24: Administrar estado de pedido

Elaboración propia

Historia de usuario	
ID.	22
Nombre de historia:	Ver cliente
Como:	Usuario administrador
Quiero:	Ver los clientes registrados en el sistema
Para poder:	conocer la información y notificar promociones futuras
Validación: <ul style="list-style-type: none"> ✓ Visualizar clientes registrado en el sistema ✓ Búsquedas a través de filtros pronombre o n° de identificación 	Valor: 50
	Prioridad: 1
	Estimación: 8 h

Tabla 25: Ver clientes

Elaboración propia

Historia de usuario	
ID.	23
Nombre de historia:	Existencia de producto
Como:	administrador
Quiero:	Que el sistema actualice la existencia de productos
Para poder:	Controlar los ítems al carrito
Validación: <ul style="list-style-type: none"> ✓ Verificar que los productos en existencia que muestre la aplicación también estén en la base de dato ✓ en caso de agotarse el producto no podrá agregarse al carrito 	Valor: 100
	Prioridad: 2
	Estimación: 8h

Tabla 26: Existencia de producto

Elaboración propia

Historia de usuario	
ID.	24
Nombre de historia:	Restar existencia de producto
Como:	Administrador
Quiero:	Que después de realizado el pedido, el sistema actualice los productos en la base de datos
Para poder:	Tener un mayor control de medicamentos vendidos, fechas de vencimientos, registro de productos.
Validación: <ul style="list-style-type: none"> ✓ Resta la cantidad vendida a la existencia de producto en inventario 	Valor: 100
	Prioridad: 2
	Estimación: 8h

Tabla 27: Restar existencia de producto

Elaboración propia

Historia de usuario	
ID:	25
Nombre de historia:	Facturación
Como:	Cliente
Quiero:	Que después que se realiza la venta, el sistema muestre la factura con los datos de la venta e imprimirla.
Para poder:	Tener un mayor soporte de las compras realizadas
Validación: ✓ Los datos de la factura sean la correcta con los datos del cliente	Valor: 100
	Prioridad: 2
	Estimación: 8h

Tabla 28: Restar existencia de producto

Elaboración propia

14. Product Backlog:

Es un listado priorizado de funcionalidades que son básicamente lo que el cliente tiene como expectativa del producto de software a desarrollarse.

El listado de las historias de usuarios será jerarquizado según la importancia para el negocio, dicha importancia se calcula multiplicando la urgencia por el tiempo.

La urgencia se medirá en una escala de 50 al 200, siendo 50 de menor valor.

El impacto se medirá en una escala del 1 al 5, siendo 5 el de mayor prioridad.

Pila de producto					
ID	Historia de usuario	valor	prioridad	importancia	Tiempo estimado(días)
1	Ver información de la farmacia	50	1	50	2
2	Ver tienda	200	5	1,000	2
3	Registrar cliente	50	1	50	2
4	Iniciar sesión	50	1	50	2
27	Facturación	50	1	50	2
5	Agregar carrito de compra	200	5	1,000	2
6	Ver carrito de compra	200	5	1,000	2
7	Actualizar compra	200	5	1,000	2
8	Eliminar producto del carrito	200	5	1,000	2
9	Realizar pedido	200	5	1,000	2
10	Pago en línea	200	5	1,000	2

11	Detalle de pedidos	200	5	1,000	2
12	Chat individual	100	5	500	2
13	Contáctenos	50	1	50	2
14	Registrar productos	150	2	300	2
15	Editar productos	150	2	300	2
16	Eliminar producto	150	2	300	2
17	Buscar producto en inventario	100	3	300	2
18	Ver lista de producto de inventario	100	2	200	2
19	Reporte de venta diarias	100	3	200	2
20	Detalle de venta	100	3	300	2
21	Administrar estado de pedido	100	2	200	2
22	Ver cliente	50	1	50	2
23	Existencia de producto	100	2	200	2
24	Restar existencia de producto	100	2	200	2
25	factura	100	5	500	2

Tabla 29: Pila de producto

Elaboración propia

14.1 Planificación inicial de las iteraciones

Planificación inicial de las iteraciones		
id	Historias de usuarios	Sprint programado
2	Ver tienda de producto	1
5	Agregar carrito de compra	1
7	Actualizar carrito de compra	1
6	Ver carrito de compra	1
8	Eliminar producto del carrito de compra	1
10	Pago en línea	1
11	Detalle de pedido	1
9	Realizar pedido	1
1	Ver información de la farmacia	2
13	Contáctenos	2
14	Registrar producto	2
15	Editar producto	2
16	Eliminar producto	2
20	Ver pedido	2
19	Reporte de venta diarias	3
21	Administrar estado de pedido	3

23	Existencia de producto	3
24	Restar existencia de producto	3
18	Ver lista de producto en inventario	3
17	Buscar producto en inventario	3
22	Ver cliente	4
12	Chat individual	4
3	Registrar cliente	4
4	Iniciar sesión	4
25	Facturación	4

Tabla 30: Planificación inicial de las iteraciones

Elaboración propia

15. Sprint Backlog

16. Planificación del primer Sprint

Planificación del primer Sprint			
Historia de usuario	tareas	Tiempo(horas)	Tarea tomada por
Ver tienda de producto	Crear diseño de la tienda	12	Jackeline Dàvila
	Crear interfaz de usuario	24	Jackeline Dàvila
	Agregar módulos	12	Keyling López
Agregar al carrito de compra	Crear registro de usuario	12	Keyling López
	Programar la funcionalidad carrito de compra	15	Jackeline Dàvila
	Conexión de la base de datos	12	Keyling López
Actualizar carrito de compra	Crear módulo de actualizar carrito	15	Jackeline Dàvila
	Programar el botón actualizar carrito	20	Jackeline Dàvila
	Realizar pruebas unitarias al módulo	10	Keyling López
Ver carrito de compra	Diseñar la interfaz carrito de compra	12	Keyling López
	Mostrar detalle de la compra	24	Jackeline Dàvila
	Pruebas del módulo	10	Jackeline Dàvila
Eliminar producto del carrito de compra	Diseño y vista de productos	10	Jackeline Dàvila
	Programar el botón eliminar	24	Jackeline Dàvila

Realizar pedido	Diseño de la interfaz de pedidos	13	Keyling López
	Programar la funcionalidad almacenar información del cliente en la base de datos	26	Jackeline Dávila
Detalle de pedido	Diseñar interfaz de usuario cliente para la compra	10	Keyling López
	Programar el botón guardar	18	Jackeline Dávila
	Pruebas de módulo	8	Jackeline Dávila
Pagos en línea	Realizar petición a PayPal para una cuenta de prueba	6	Keyling López
	Programar la conexión de PayPal con la aplicación	15	Keyling López

Tabla 31: Planificación del primer Sprint

Elaboración propia

16.1 Burn Down Chart

En este Sprint se ha asignado una duración de 2.1 semanas, se considera un equipo dedicado al desarrollo de 2 miembros, el primer Sprint se desarrolló desde el 19 de octubre al 2 de noviembre del 2020.

La tabla de Burn Down Chart, muestra la pendiente del trabajo ideal vs el real.

Como se observa en el gráfico, está compuesto por 8 historias de usuario y 21 tarea en el primer sprint, de la cual, pago en línea contiene 2 tareas que no se logró completar en la fecha establecida, de manera que el grafico muestra 19 completas y 2 tareas inconclusas, estas se anexarán en el siguiente Sprint, la franja azul muestra lo que se espera realizar en el lapso de tiempo determinado, y la franja verde, es la evolución que muestra el proyecto al realizar cada tarea.

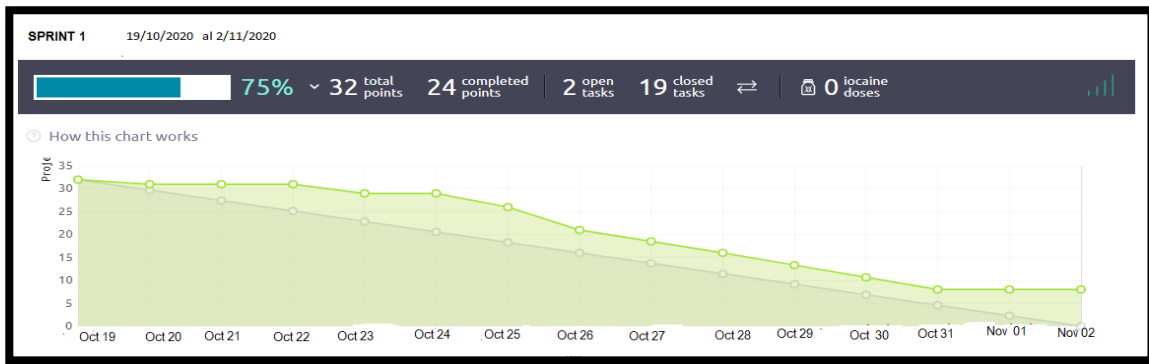


Ilustración 16: Gráfico del primer sprint 1

Elaboración propia

16.1.2. Resultado del Sprint. Aquí se muestran las pantallas realizadas por el sprint 1.

Ilustración 17: Ver Tienda de producto

Elaboración propia

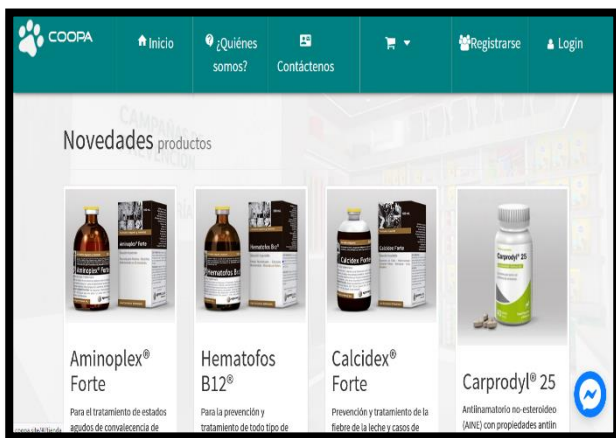


Ilustración 19: Ver carrito de compra

Elaboración propia

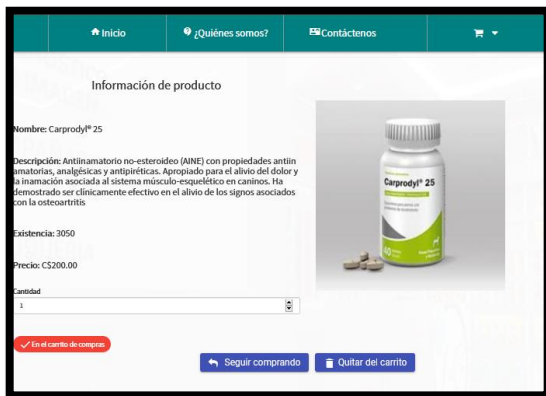
Ilustración 18: Agregar al carrito de compra

Elaboración propia



Ilustración 20: Eliminar producto del carrito

Elaboración propia



Producto	Descripción	Precio	Cantidad	Sub Total	Quitar
Calcidex® Forte	Prevención y tratamiento de la fiebre de la leche y casos de deficiencia de calcio, fósforo y magnesio; fiebre de leche, otras formas de hipocalcemia, hipomagnesemia, tónico reconstituyente, intoxicaciones y cetosis.	C\$300.00	1	C\$300.00	[X]
Carprodyll® 25	Antinflamatorio no-esteroides (AINE) con propiedades antiinflamatorias, analgésicas y antipiréticas. Apropriado para el alivio del dolor y la inflamación asociada al sistema músculo-esquelético en caninos. Ha demostrado ser clínicamente efectivo en el alivio de los signos asociados con la osteoartritis.	C\$200.00	1	C\$200.00	[X]
Total: C\$500.00					

Ilustración 21: Realizar pedido

Elaboración propia

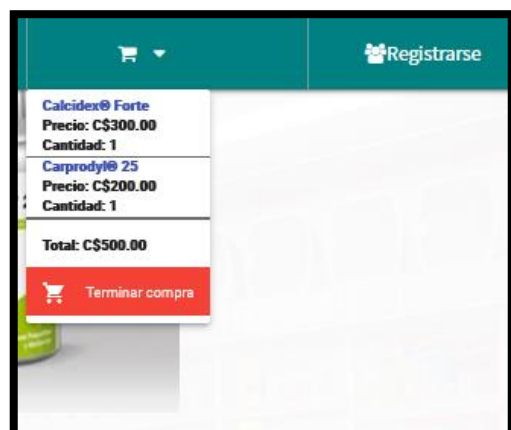


Ilustración 22: Detalle de pedido

Elaboración propia



17. Planificación del segundo sprint

Planificación del segundo sprint				
Historia de usuario	de	tareas	Tiempo(horas)	Tarea tomada por
Pago en línea		Realizar petición a PayPal para una cuenta de prueba	12	Jackeline Dàvila
		Programar la conexión de PayPal con la aplicación	24	Jackeline Dàvila
Ver información de la farmacia		Diseñar interfaz de información de la farmacia	12	Keyling López
		Programar la interfaz	15	Jackeline Dàvila
Registrar producto		Diseño de interfaz del administrador	15	Jackeline Dàvila
		Programar la interfaz producto	20	Jackeline Dàvila
Editar producto		Programar el botón editar	12	Keyling López
		Comprobar que el producto exista en inventario	24	Jackeline Dàvila
		Programar el botón guardar	10	Keylin López
		Realizar pruebas para el botón guardar	15	Jackeline Dàvila

Eliminar producto	Programar el botón eliminar producto	10	Jackeline Dàvila
Ver detalle de venta	Diseñar módulo detalle de venta	12	Jackeline Dàvila
	Programar módulo de detalle de venta	10	Keylin López
	Realizar pruebas del módulo	11	Jackeline Dàvila
Contáctenos	Diseño de interfaz, contáctenos	10	Keylin López
	Programar módulo contáctenos	8	Jackeline Dàvila

Tabla 32: Planificación de segundo Sprint

Elaboración propia

17.1. Burn Down Chart

En este Sprint se ha asignado una duración de 2.1 semanas, se considera un equipo dedicado al desarrollo de 2 miembros, en Sprint 2 se desarrolló desde el día 3 de noviembre al 17 del mismo mes del año 2020.

La tabla de Burn Down Chart, muestra la pendiente del trabajo ideal vs el real del segundo Sprint.

La tabla anterior muestra el progreso de las 7 historia de usuario y 16 tareas, elaborada en el tiempo establecido, la historia de usuario editar producto contiene 3 tarea que quedaron inconclusas el grafico muestra 13 tareas completas y 3 tareas incompleta, por lo tanto, tareas se anexarán al siguiente sprint.

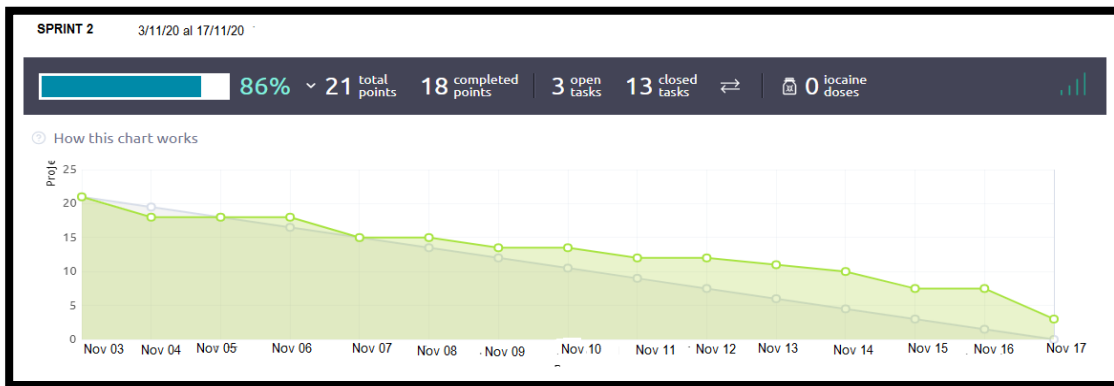


Ilustración 23: Gráfico del segundo Sprint

Elaboración propia

17.1.2. Resultado del segundo sprint. Muestra las pantallas realizadas en el segundo sprint

Ilustración 24: Pago en línea

Elaboración propia

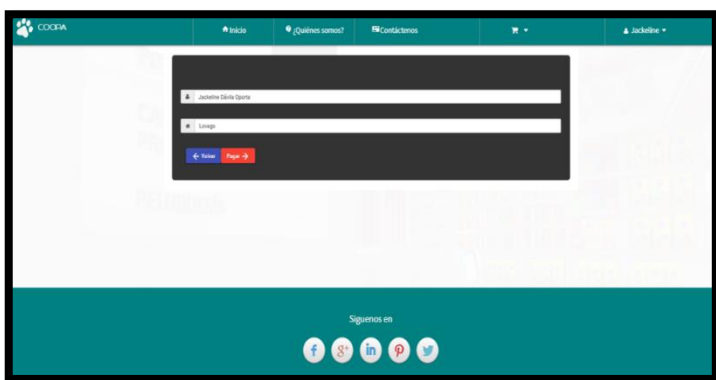


Ilustración 25: PayPal

Elaboración propia

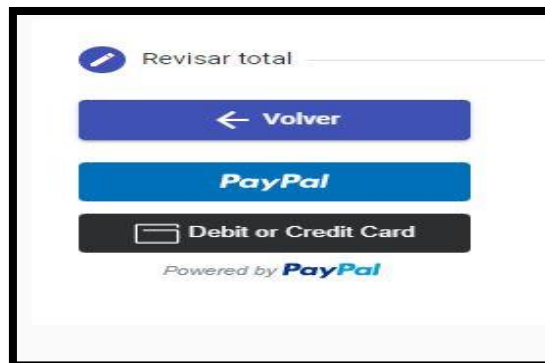


Ilustración 26: Cuenta en PayPal

Elaboración propia

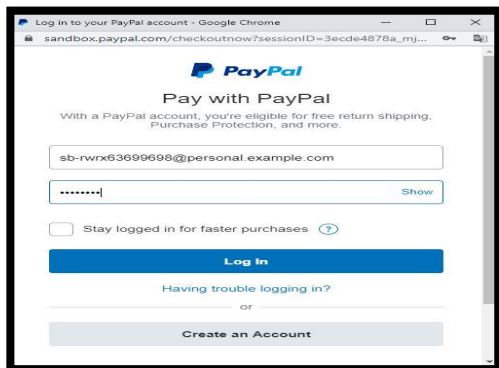


Ilustración 27: Factura

Elaboración propia

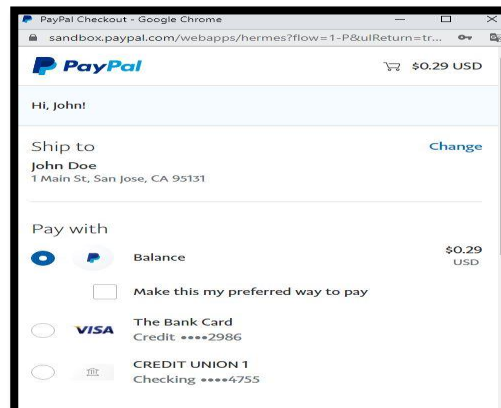


Ilustración 28: Ver información de la farmacia

Elaboración propia

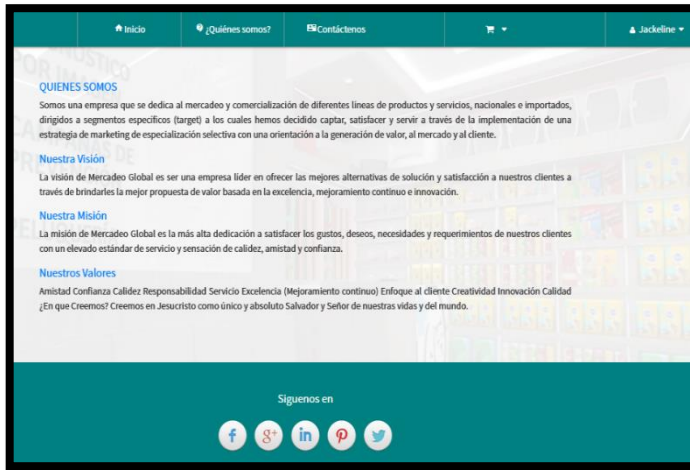


Ilustración 30: Eliminar producto

Elaboración propia

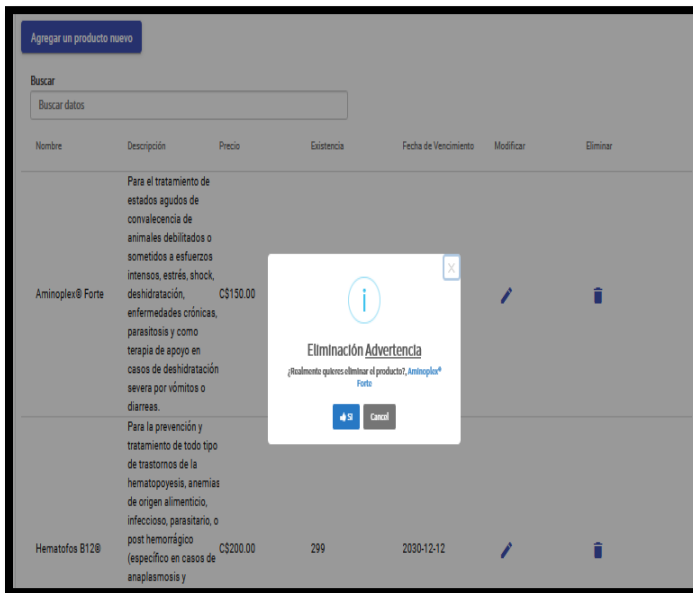


Ilustración 29: Registrar productos

Elaboración propia

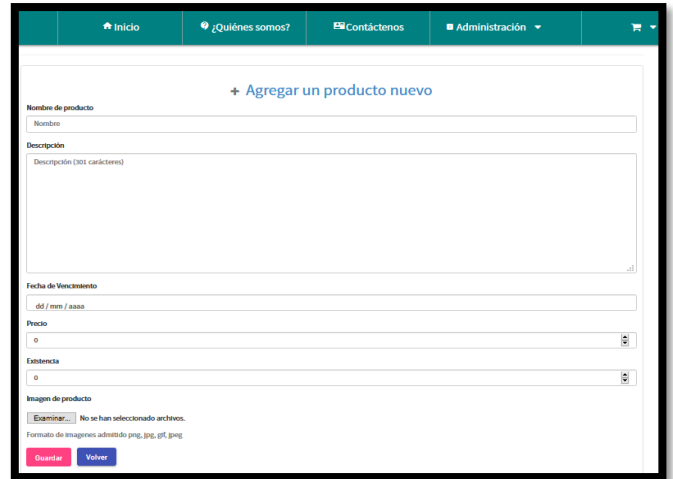


Ilustración 31: Ver detalle de venta

Elaboración propia

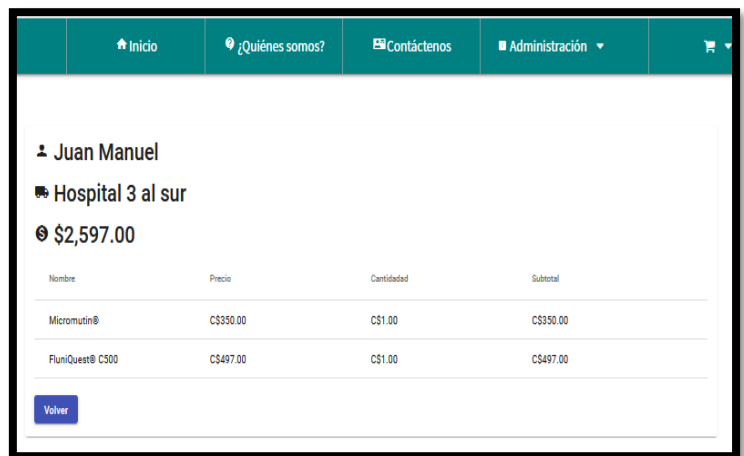
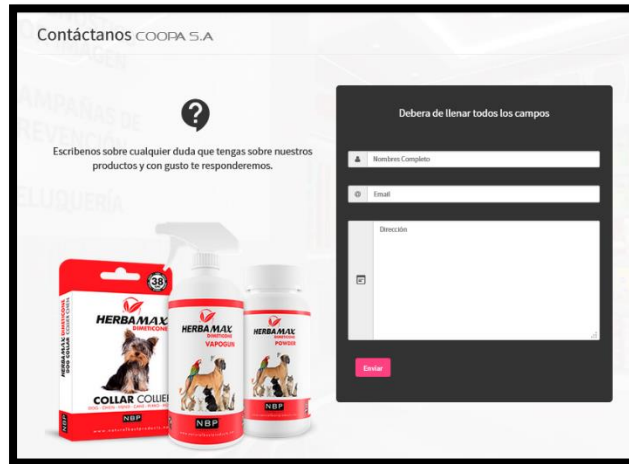


Ilustración 32: Contáctenos

Elaboración propia



18. Planificación del tercer Sprint.

Planificación del tercer sprint				
Historia de usuario	de	tareas	Tiempo(horas)	Tarea tomada por
Editar producto		Programar el botón editar	9	Jackeline Dàvila
		Programar el botón guardar y editar	8	Keyling López
		Realizar pruebas para el botón guardar	10	Jackeline Dàvila
Existencia de producto		Realizar peticiones a la base de datos para ver los productos existentes	12	Jackeline Dàvila
		Programación de la interfaz producto	24	Jackeline Dàvila
		Pruebas unitarias	12	Keyling López
Reporte de venta diarias		Diseño de la interfaz reporte de venta	12	Keyling López
		Programar los filtros para buscar reportes	15	Jackeline Dàvila
		Programar la descarga de reportes	12	Keyling López
Restar existencia de producto		Programar la resta de existencia de producto	15	Jackeline Dàvila
		Pruebas unitarias para restar existencia	10	Keyling López

Ver lista de productos en inventario	Diseñar la interfaz producto	12	Keyling López
Buscar productos en inventario	Programación búsqueda de producto en la base de dato, con filtros	10	Jackeline Dávila
Administrar estado de pedido	Diseño de la interfaz estado	8	Jackeline Davila
	Programar estados: pendiente, en proceso, entregado	15	Keylin López
	Realizar pruebas	8	Jackeline Dávila

Tabla 33: Planificación del tercer Sprint

Elaboración propia

18.1. Burn Down Chart: En este Sprint se ha asignado una duración de 2.1 semanas, se considera un equipo dedicado al desarrollo de 2 miembros, se inició el 18 de noviembre y se completó el 2 de diciembre.

El gráfico muestra las 16 tareas completas, además se observa las puntuaciones establecidas a medida que las tareas se van completando.

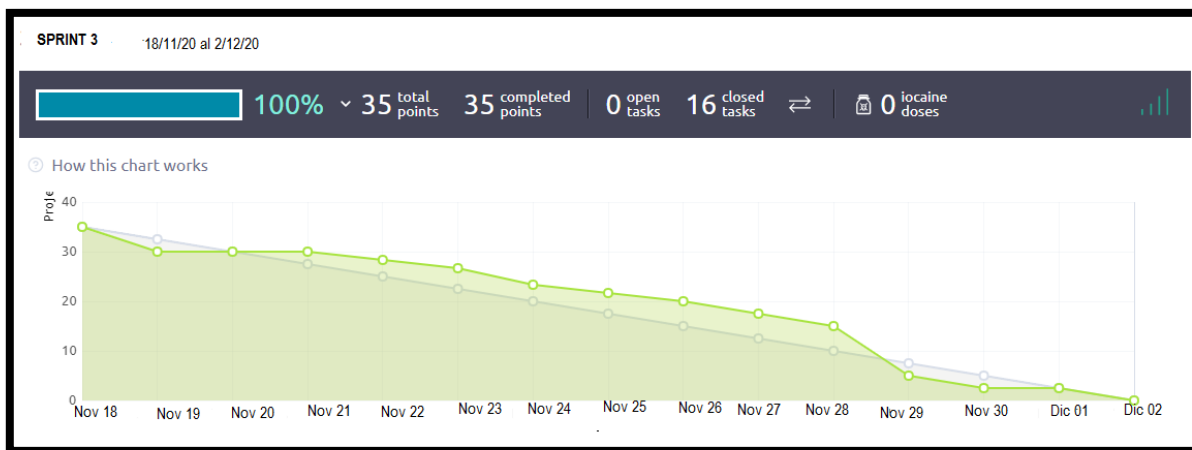


Ilustración 33: Gráfico del tercer Sprint

Elaboración propia

18.1.2. Resultado del tercer sprint. Muestra los los resultados obtenido del sprint

Ilustración 34: Editar producto

Elaboración propia

Panel de administración COOPA S.A.

+ Agregar un producto nuevo

Nombre de producto
Hematofos B12

Descripción
Para la prevención y tratamiento de todo tipo de trastornos de la hematopoyesis, anemias de origen alimenticio, infeccioso, parasitario, o post hemorrágico (específico en casos de anaplasmosis y piroplasmosis), debilidad, enflaquecimiento y convalecencia. Potente reconstituyente, tónico y estimulante.

Fecha de Vencimiento
12/12/2030

Precio
200

Existencia
299

Actualizar Volver

Ilustración 35: Existencia de producto

Elaboración propia

Nombre	Descripción	Precio	Existencia	Fecha de Vencimiento	Modificar	Eliminar
Hematofos B12	Para la prevención y tratamiento de todo tipo de trastornos de la hematopoyesis, anemias de origen alimenticio, infeccioso, parasitario, o post hemorrágico (específico en casos de anaplasmosis y piroplasmosis), debilidad, enflaquecimiento y convalecencia. Potente reconstituyente, tónico y estimulante.	C\$200.00	299	2030-12-12		

Ilustración 36: Restar existencia de producto

elaboración propia

Nombre	Descripción	Precio	Existencia	Fecha de Vencimiento	Modificar	Eliminar
Hematofos B12	Para la prevención y tratamiento de todo tipo de trastornos de la hematopoyesis, anemias de origen alimenticio, infeccioso, parasitario, o post hemorrágico (específico en casos de anaplasmosis y piroplasmosis), debilidad, enflaquecimiento y convalecencia. Potente reconstituyente, tónico y estimulante.	C\$200.00	296	2030-12-12		

Ilustración 37: Reportes de ventas

Elaboración propia

Panel de administración COOPA S.A.

REPORTE DE VENTAS

Buscar
Buscar datos

Cliente	Dirección de envío	Total	Fecha/vta
Juan Manuel	Hospital 3 al sur	C\$2,597.00	1/21/21
Teodoro Ruiz	Inta 5 al 2 oeste	C\$200.00	1/22/21
Jackeline Dávila Oporta	Lovago	C\$150.00	1/22/21
Bryan Davilaacoyapa		C\$150.00	1/22/21
Jackeline Dávila Oporta	Lovago	C\$350.00	1/22/21
Jackeline Dávila Oporta	Lovago	C\$150.00	1/28/21
Jackeline Dávila Oporta	Lovago	C\$500.00	2/3/21
Jackeline Dávila Oporta	Lovago	C\$800.00	2/3/21

Ilustración 38: Ver lista de producto en inventario

Elaboración propia

Buscar
hema

Nombre	Descripción	Precio	Existencia	Fecha de Vencimiento	Modificar	Eliminar
Hematofos B12	Para la prevención y tratamiento de todo tipo de trastornos de la hematopoyesis, anemias de origen alimenticio, infeccioso, parasitario, o post hemorrágico (específico en casos de anaplasmosis y piroplasmosis), debilidad, enflaquecimiento y convalecencia. Potente reconstituyente, tónico y estimulante.	C\$200.00	296	2030-12-12		

Ilustración 39: Buscar productos

Elaboración propia

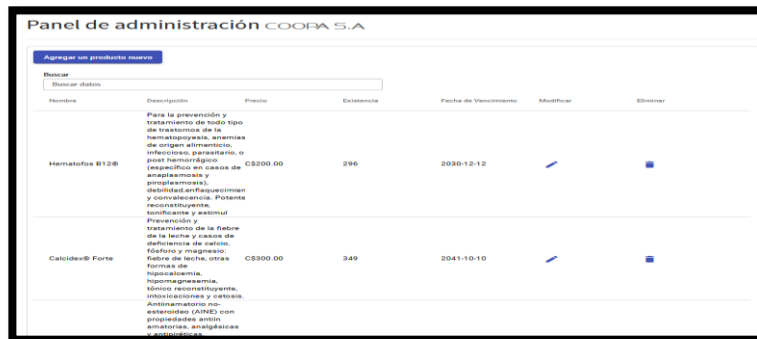
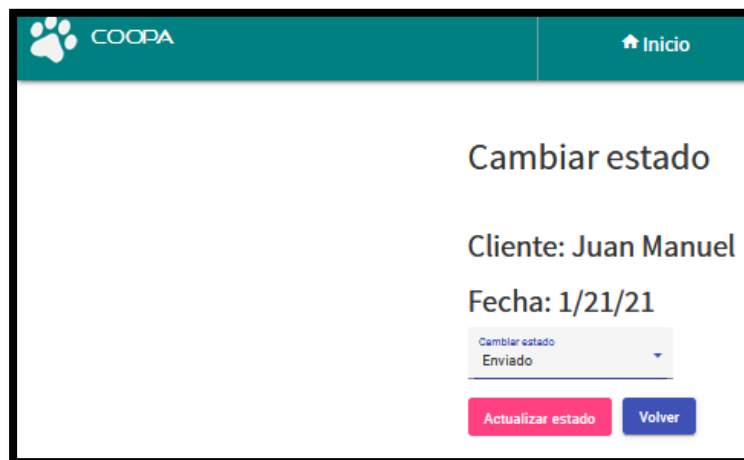


Ilustración 40: Administrar estado

Elaboración propia.



19. Planificación del cuarto sprint

Planificación del cuarto sprint				
Historia de usuario	de	tareas	Tiempo(horas)	Tarea tomada por
Ver cliente		Diseño de interfaz ver lista de los clientes de la farmacia	12	Jackeline Dàvila
		Programación del módulo cliente	24	Jackeline Dàvila
Chat individual		Diseño de la interfaz chat	12	Keyling López
		Solicitud a Facebook	15	Jackeline Dàvila
		Conexión del chat con la aplicación	12	Keyling López

	Alojamiento de la aplicación	10	
Registrar cliente	Diseño de registro de cliente	15	Jackeline Dàvila
	Programar el formulario registro	20	Jackeline Dàvila
	Pruebas unitarias del registro	10	Keyling López
Iniciar sesión	Diseño de la interfaz iniciar sesión	12	Keyling López
	Programar la funcionalidad iniciar sesión	24	Jackeline Dàvila
	Pruebas unitarias	14	Keyling López
Factura	Diseñar el módulo factura	20	Jackeline Dàvila
	Obtener registro de los productos vendidos	12	Jackeline Dàvila
	Mostrar las facturas	10	Jackeline Dàvila

Tabla 34: planificación del cuarto Sprint

Elaboración propia

19.1. Burn Down Chart

En este sprint se ha asignado una duración de 2.1 semanas, se considera un equipo dedicado al desarrollo de 2 miembros, el tiempo establecido fue desde el 3 de diciembre al 17 de ese mismo mes.

El siguiente grafico muestra 15 tareas finalizadas, y las puntuaciones establecidas.

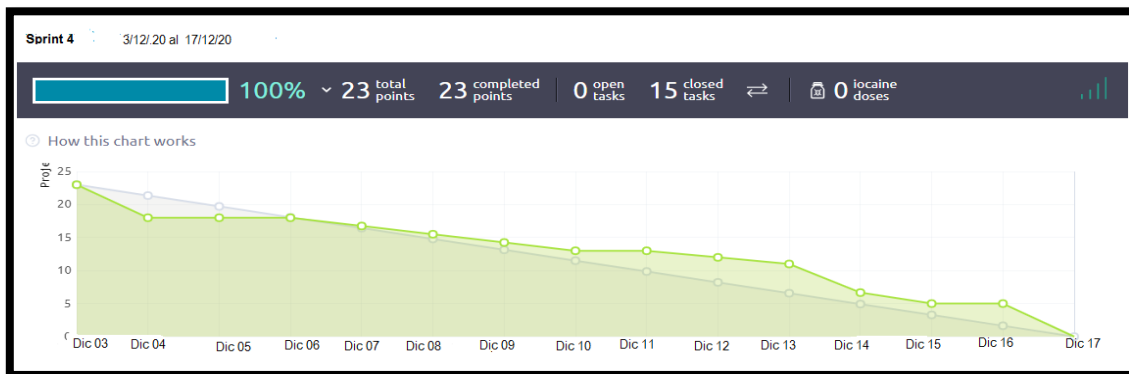


Ilustración 41: Gráfico del cuarto Sprint

Elaboración propia.

19.1.2. Resultados del cuarto sprint: se muestran las interfaces realizadas en las historias de usuario.

Ilustración 42: Ver cliente

Elaboración propia

Cliente	Dirección de envío	Total	Fecha	Estado	Detalles
Juan Manuel	Hospital 3 al sur	CS2.597.00	1/21/21	Entregado	
Teodoro Ruiz	inta 5 al 2 oeste	CS200.00	1/22/21	En espera	
Jackeline Dávila Oporta	Lovago	CS150.00	1/22/21	En espera	
Bryan davila	ecoyapa	CS150.00	1/22/21	En espera	
Jackeline Dávila Oporta	Lovago	CS350.00	1/22/21	En espera	
Jackeline Dávila Oporta	Lovago	CS150.00	1/28/21	En espera	
Jackeline Dávila Oporta	Lovago	CS500.00	2/3/21	En espera	
Jackeline Dávila Oporta	Lovago	CS800.00	2/3/21	En espera	

Ilustración 43: Inicio de sesión

Elaboración propia

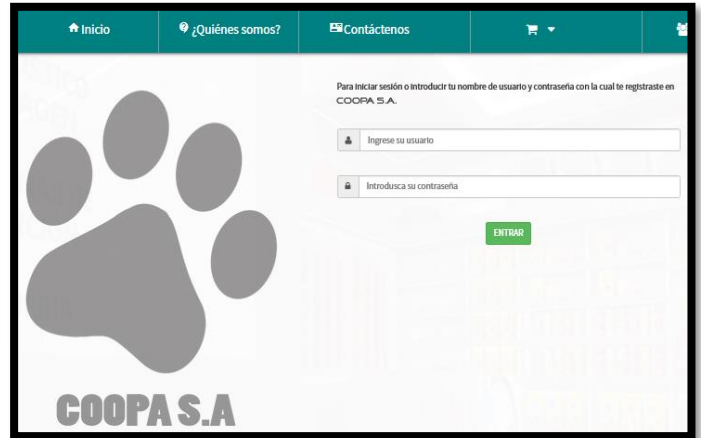


Ilustración 44: Registrar cliente

Elaboración propia

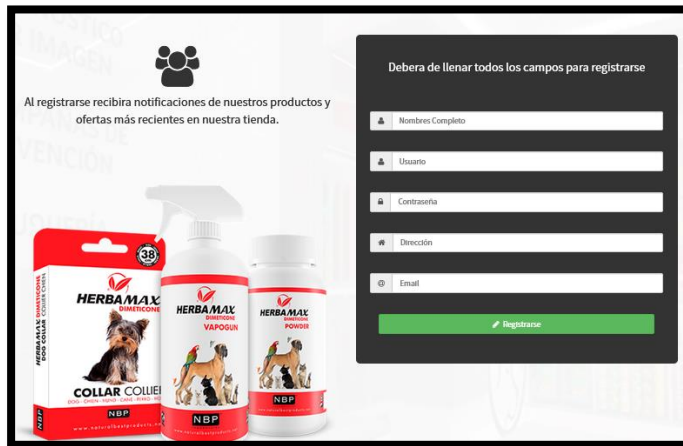


Ilustración 45: Chat individual

Elaboración propia



Ilustración 46: Facturación

Elaboración propia

Producto	Cantidad	Precio	Subtotal
jgjel	1	C\$100.00	C\$100.00
Total			C\$100.00

20. Presupuesto.

Para calcular el presupuesto de proyecto Farm-Veterinaria COOPA, es indispensable tomar en cuenta que.

- El costo de la aplicación no será determinado por línea de código, y esfuerzo por lo tanto se descarta el modelo básico de COCOMO
- Se ha tomado en cuenta los servicios necesarios para el desarrollo de la aplicación
- El pago de los miembros del equipo de desarrollo es una cantidad única hasta que el proyecto sea completado en la fecha establecida.
- Viático de transporte y alimentación, será un beneficio extra al equipo de desarrollo.

El presupuesto esta segmentado en dos puntos:

- 1) contratación de servicios.
- 2) Viático.

Contratación de servicio:

El desarrollo y publicación de la aplicación web para la farmacia veterinaria COOPA necesita contratar servicios esenciales para la ejecución las cuales son: Base de datos, internet, cuenta en Paypal, hosting y dominio para subir la aplicación.

Elemento	Cantidad	costo	Tiempo (2.5 meses)	Total, de elementos
Equipo de desarrollo				
Diseñador	1	USD 400.00	2 meses	USD 800.00
programador	1	USD 400.00	2.5 meses	USD 1,000.00
Tecnología				
MariaDB	1	USD 40.00	No aplica	USD 40.00
Necesario para publicar la app				
Conexión a internet	1	USD 30.00	3 meses	USD 90.00
Compra de hosting, dominio	1	USD 28.00	3 meses	USD 84.00
Necesarias para la visibilidad del proyecto				
Perfiles en redes sociales	2	USD -	-	-
Viatico				
trasporte	1	USD 80.00	2.5 meses	USD 200
Alimentación	2	USD 120.00	2.5 meses	USD 300
Monto total:				USD 2,514

Tabla 35: Presupuesto

Elaboración propia

Planeación de costo de pruebas de Software

ID recursos	Tipo de recursos	Unidad	Costo de recursos por unidad	Horas	Cantidad	Total
1	Desarrollador	2	\$40	8	16	USD 80
2	pc	2	\$20	15	30	USD 40
3	internet	1	\$10	15	15	USD 10
4	Materiales					
	Lápiz	2	\$2	6	4	USD 4
	cuadernos	2	\$2	6	4	USD 4
Total						USD 138

Tabla 36: Presupuesto de pruebas de software

Elaboración propia

Monto total del proyecto es de **USD 2,652**

21. Visibilidad del producto

La estrategia de la aplicación comienza antes del lanzamiento, se busca desarrollar una base de usuario interesados en la aplicación para poder conseguirlo se realizan las siguientes actividades.

- Hacer perfiles para la aplicación en distintas redes sociales, tales como Facebook, Instagram. Usar las redes como estrategia para llamar la atención a los clientes.
- Lanzar publicación por los medios locales. Esta estrategia en la actualidad continúa siendo una manera efectiva de atraer a los usuarios.

22. Marco lógico.

Se muestra el planteamiento del proyecto a través de la metodología de marco lógico.

Matriz de marco lógico.

La matriz de marco lógico es un resumen de las actividades y objetivos, el objetivo de un proyecto es la solución a una problemática a la que se han observado, y a la que sea definido una actividad para cumplir con la solución.

Se presenta la solución a la problemática planteada a través de la metodología marco lógico en la siguiente matriz.

23. Matriz de marco lógico

narrativo	indicador	Medio de verificador	Supuestos
<p>Fin</p> <p>Aumentar el control de las ventas y facturación de productos veterinarios, de la farmacia veterinaria COOPA.</p> <p>Propósito</p> <p>Desarrollar una aplicación web, para el control de ventas y facturación en línea de la farmacia veterinaria COOPA</p>	<ul style="list-style-type: none"> ❖ 100% de los productos están en inventario ❖ Las ventas de la farmacia veterinaria aumentan en un 90%. ❖ 80% de visitas a la aplicación web. ❖ Aplicación web 100% funcional ❖ El 80% de los clientes y productores que realizan compras en la farmacia veterinaria, hagan uso de la aplicación web como una alternativa más eficaz y rápida para la compra de sus productos. ❖ El 90% de los propietarios de la farmacia COOPA ven que el sistema de facturación en línea suple con las necesidades del negocio 	<ul style="list-style-type: none"> ❖ Reportes de ventas ❖ Verificar inventarios ❖ Facturación de venta realizadas ❖ Clasificaciones y reseñas por medio de un contador de visitas ❖ Reporte de la de personas que realizan compras a través de la aplicación web ❖ El administrador de la aplicación realiza facturación con facilidad, disminuyendo pérdidas económicas. 	<ul style="list-style-type: none"> ❖ Poca disponibilidad de la aplicación en los navegadores ❖ Personal de la farmacia COOPA no satisfecho ❖ Estructura de costo no cambie mediante el desarrollo del proyecto. ❖ Falta de interés en el uso de la aplicación por parte de los productores.

Tabla 37: Matriz de marco lógico

Elaboración propia

Componentes			
1. Diseñar una herramienta tecnológica que satisfaga los requerimientos de la gestión de venta y facturación de la farmacia veterinaria COOPA	❖ el usuario administrador está satisfecho con la herramienta tecnología desarrollada al 100%	Reportes de Stock, y ventas realizadas	Poca adaptación a la aplicación por parte de los usuarios
2. Crear un chat individual para la comunicación constante con el cliente	❖ El 90% de los clientes envían sugerencias	Ver los clientes que realizan consulta en el chat	Al cliente no le interese contactarnos
3. Crear un apartado de stock de inventario	❖ El stock de es inventario 100% funcional	Verificación de productos en inventarios	❖ Que la aplicación genere errores en inventarios
4. Generar reportes y facturas	❖ Los reportes de ventas y compras son 100% funcionales	Analizar reportes de la venta en aplicación web Gestión de factura	❖ Que muestre reportes erróneos ❖ La factura no es acorde con las fechas

Tabla 38: Matriz de marco lógico

Elaboración propia

Actividades del componente 1			
<p>Diseñar interfaz amigable y llamativa para el fácil manejo de la aplicación</p> <p>Desarrollar una herramienta tecnológica para la gestión de ventas y facturación de productos veterinarios en la farmacia COOPA</p> <p>Desarrollar una interfaz para usuario administrador, usuario cliente y usuario no registrado</p>	<ul style="list-style-type: none"> ❖ 95% de los clientes les gusta la interfaz de la aplicación ❖ 100% de cada interfaz de usuario administrador, cliente y usuario no registrado son funcionales 	<ul style="list-style-type: none"> ❖ Comentarios y reseñas por medio de las visitas realizadas Casos de pruebas 	<ul style="list-style-type: none"> ❖ Aceptación de la aplicación ❖ Errores de pruebas
	<ul style="list-style-type: none"> ❖ Desarrollo de las funcionalidades de la interfaz producto, ventas, clientes, son 100% funcionales. 	<ul style="list-style-type: none"> ❖ Clasificaciones y reseñas por medio de las visitas realizadas ❖ Reporte de la cantidad de productos que actualizan, eliminan, editan y agregan en la aplicación web. ❖ Reportes de ventas ❖ Casos de pruebas ❖ Facturación de venta 	<ul style="list-style-type: none"> ❖ Interfaz producto, venta y cliente poco entendible para el usuario administrador
	<ul style="list-style-type: none"> ❖ Desarrollo de interfaz de usuario administrador, cliente y usuario no registrado 100% funcional 	<ul style="list-style-type: none"> ❖ Reporte de la cantidad de usuarios registrados 	<ul style="list-style-type: none"> ❖ El usuario no se puede registrar en la aplicación.

Tabla 39: Matriz de marco lógico

Elaboración propia

<p>Desarrollar una interfaz para cliente</p>	<ul style="list-style-type: none"> ❖ Crear interfaz de inicio de sesión 100% funcional ❖ Crear interfaz de mi pedido 100% funcional ❖ Crear interfaz de carrito de compras 100% funcional 	<ul style="list-style-type: none"> ❖ Comentarios por parte de los usuarios ❖ Ver las sugerencias que envían los usuarios en la aplicación 	<ul style="list-style-type: none"> ❖ Problemas técnicos a la hora de mostrar la interfaz cliente
<p>Desarrollar una interfaz para usuario no registrado</p>	<ul style="list-style-type: none"> ❖ Desarrollar interfaz de inicio 100% funcional ❖ Desarrollar interfaz de inicio de sesión 100% funcional ❖ Desarrollar una interfaz de contáctenos 100% funcional ❖ Desarrollo de una interfaz tienda de producto 100% funcional 	<ul style="list-style-type: none"> ❖ Clasificaciones y reseñas por medio de las visitas realizadas ❖ Ver sugerencias 	<ul style="list-style-type: none"> ❖ El usuario no le parece la aplicación por lo tanto no se registra
<p>Actividades del componente 2</p> <p>Realizar una interfaz para el chat con las funcionalidades de enviar y recibir mensajes</p>	<ul style="list-style-type: none"> ❖ Interfaz funcional 100% 	<ul style="list-style-type: none"> ❖ Reporte de la cantidad de clientes en la sala de chat 	<ul style="list-style-type: none"> ❖ Interfaz poco entendible

Tabla 40: Matriz de marco lógico

Elaboración propia

<p>Actividades del componente 3</p> <p>Diseñar interfaz para la gestión de Stock de productos</p> <p>Diseño de interfaz de agregar eliminar y editar productos</p>	<p>❖ Interfaz de Stock de producto es 100% funcional</p>	<p>Ver informe de los productos registrados</p>	<p>❖ La aplicación tiene errores en el ingreso de productos.</p>
<p>Actividades del componente 4</p> <p>Diseñar interfaz de reporte con las funcionalidades de ver reporte de ventas y filtrar reportes</p>	<p>❖ Interfaz de reportes 100% funcional</p>	<p>❖ Verificación de reportes</p>	<p>❖ Información errónea</p>
<p>diseño de interfaz de facturación, con la funcionalidad eliminar, editar y agregar facturas</p>	<p>❖ Interfaz de factura 100% funcional</p>	<p>Verificación de factura</p>	<p>❖ Datos erróneos</p>

Tabla 41: Matriz de marco lógico

Elaboración propia

24. Árbol del problema.

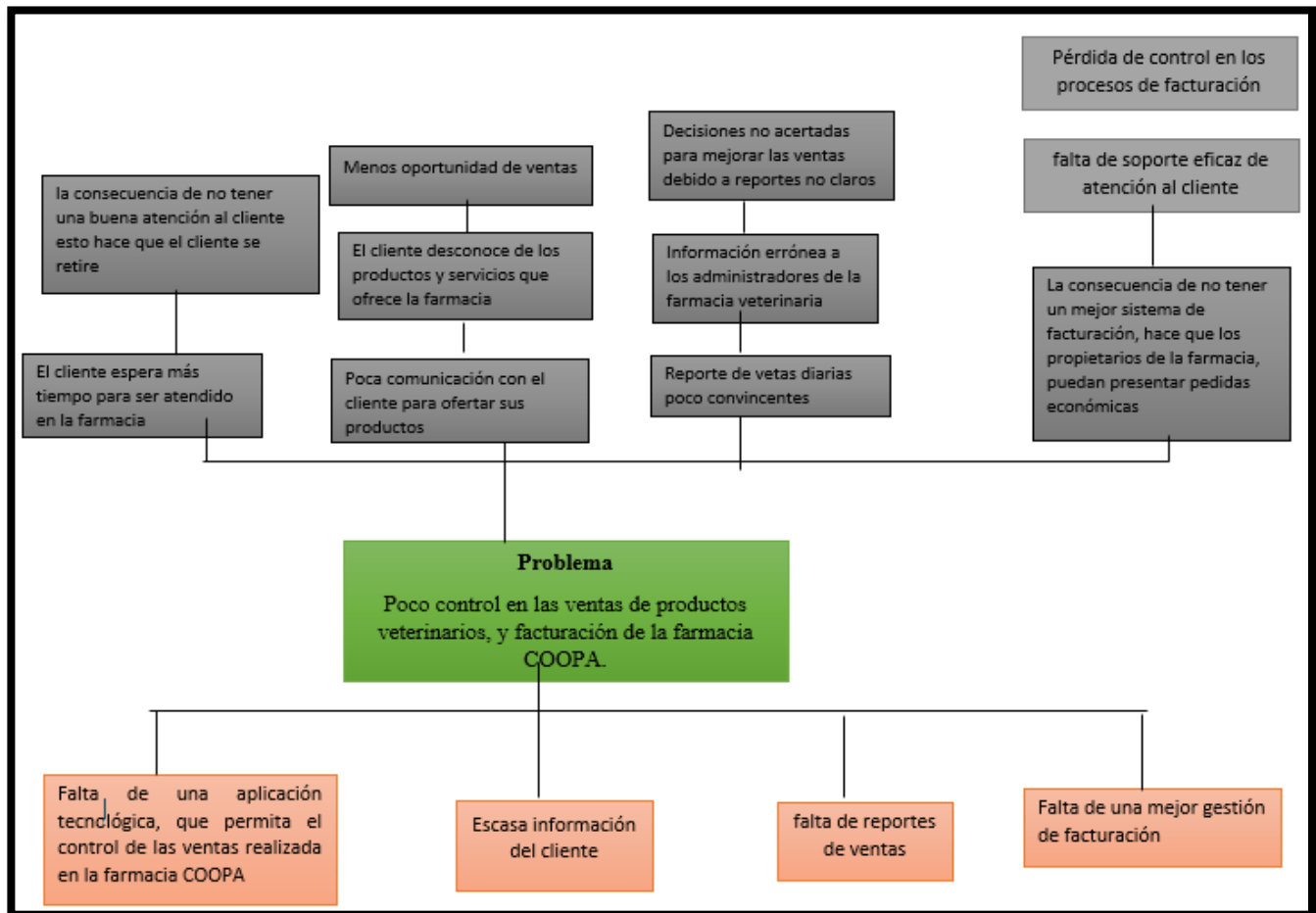


Ilustración 47: Árbol del problema

Elaboración propia

25. Objetivo del proyecto

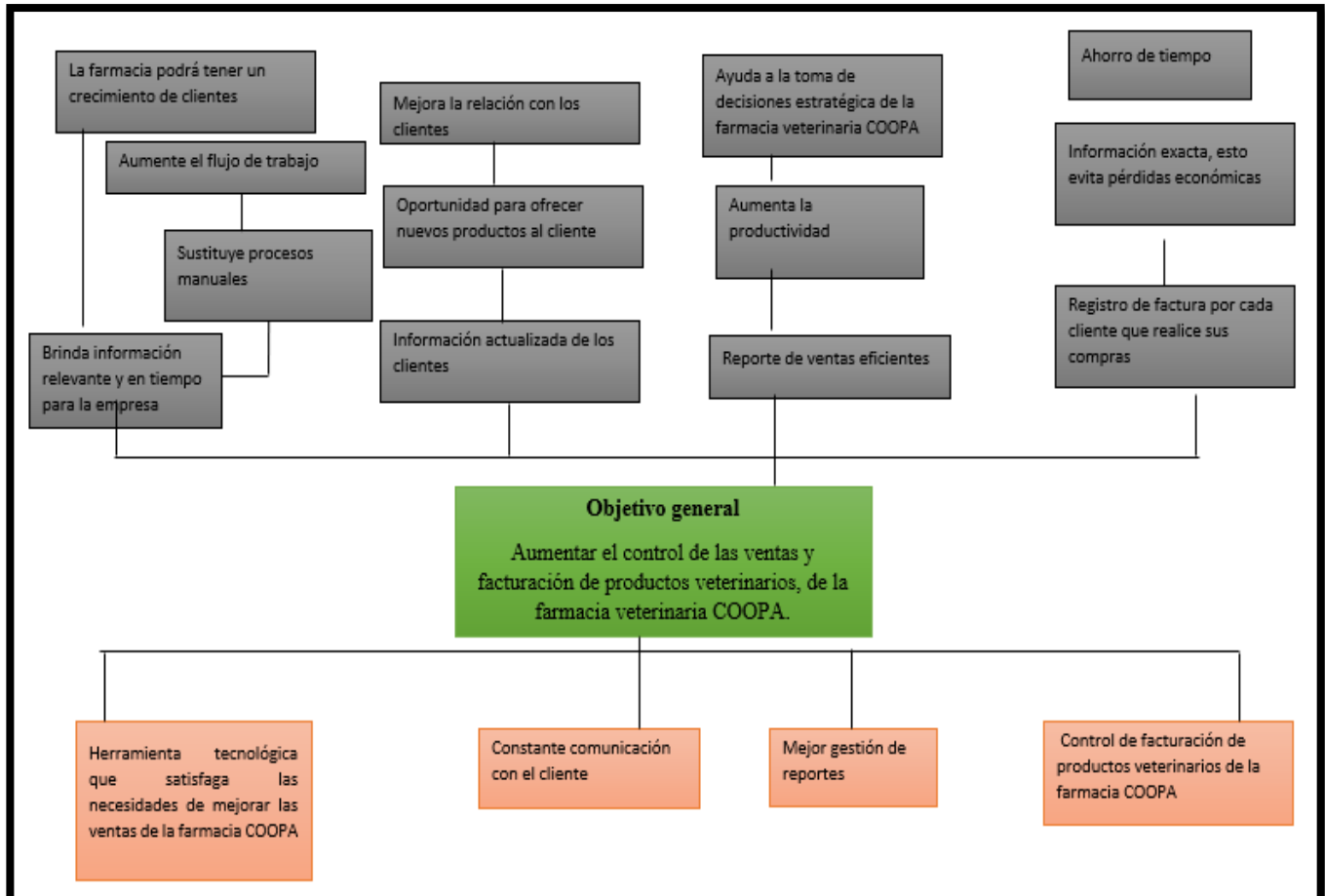


Ilustración 48: Árbol de objetivos

Elaboración propia

26. Árbol de análisis de la solución

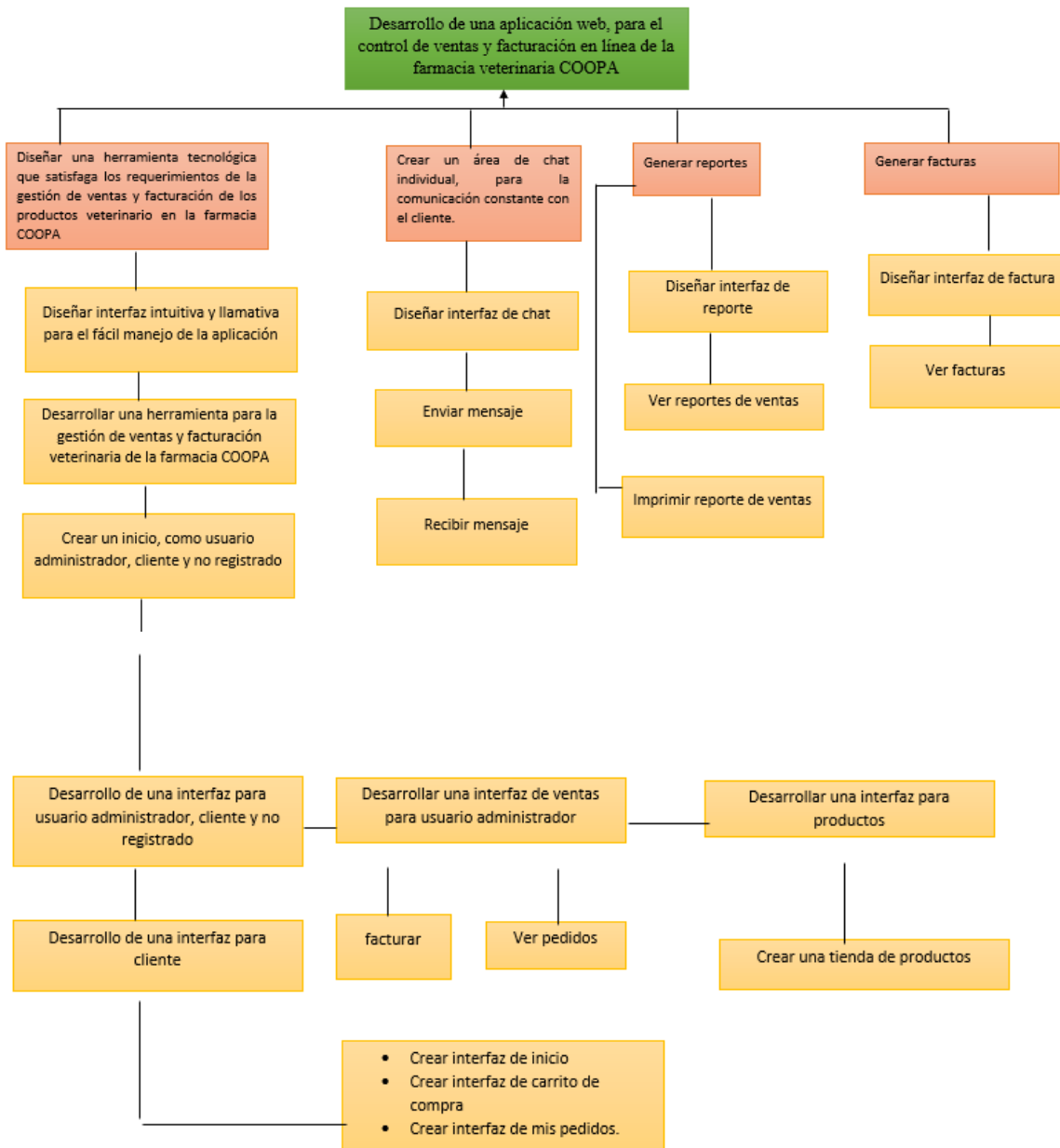
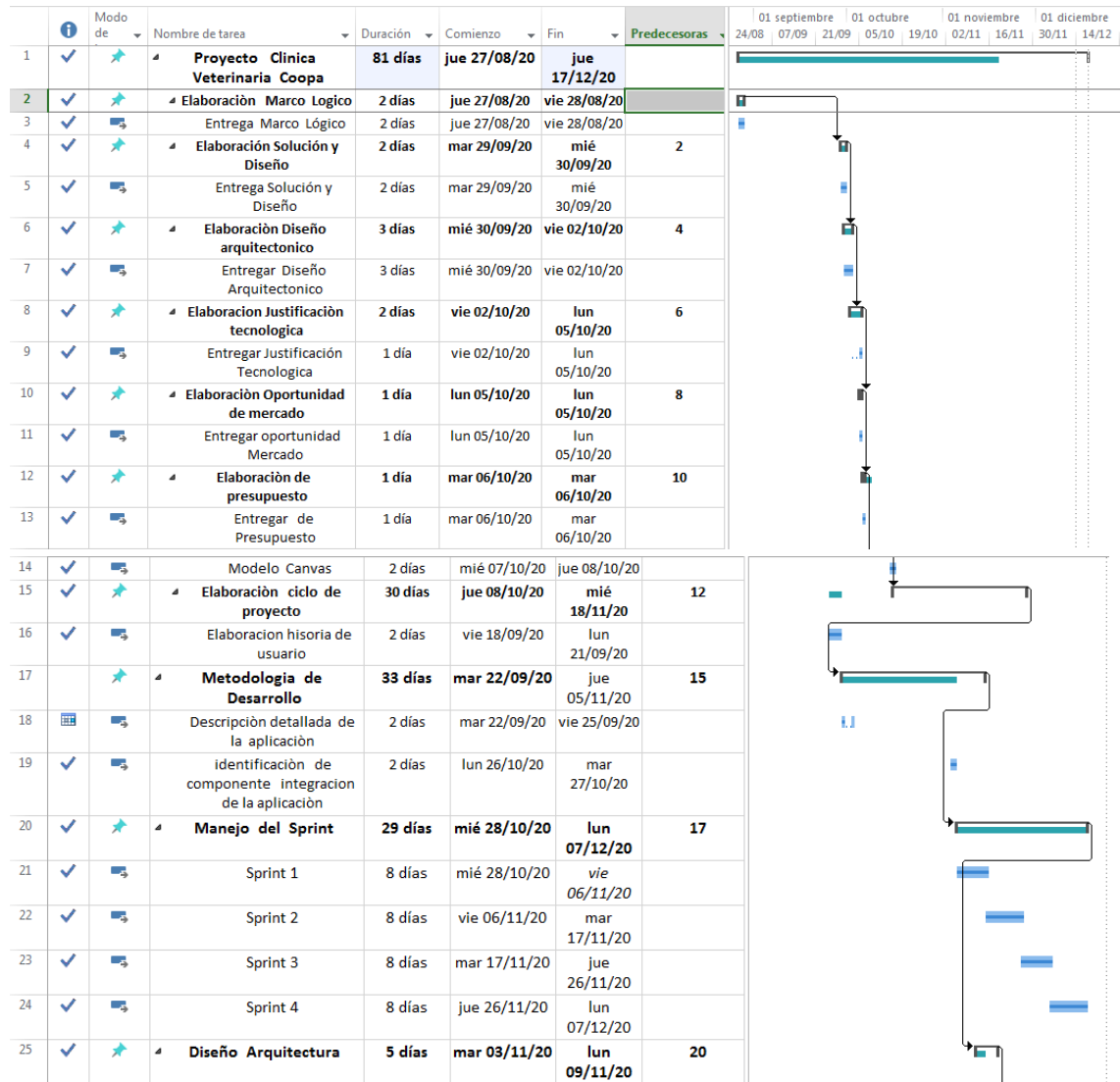


Ilustración 49: Árbol de análisis de soluciones

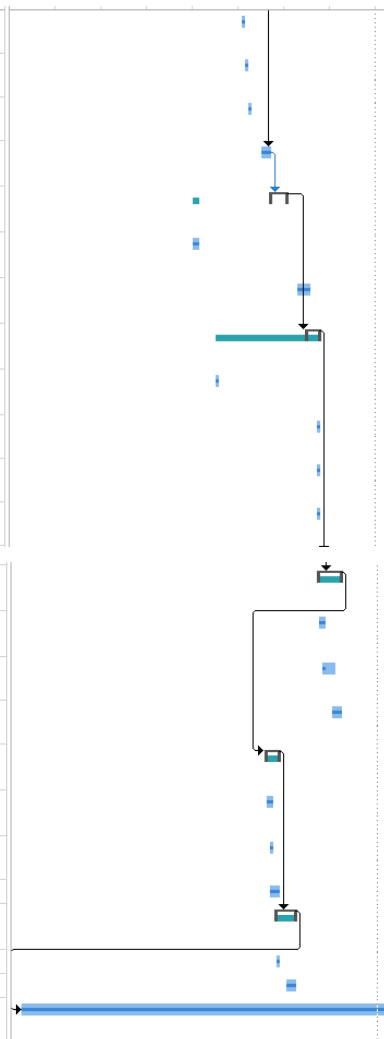
Elaboración propia

27. Cronograma de actividades.

El cronograma de actividades está diseñado para llevar un control de las actividades y cada entrega del proyecto.



26	✓	📄	Definición de la funcionalidad	1 día	mar 03/11/20	mar 03/11/20	
27	✓	📄	Definición de la impletación tecnología	1 día	mié 04/11/20	mié 04/11/20	
28	✓	📄	Diseño de los componente APP	1 día	jue 05/11/20	jue 05/11/20	
29	✓	📄	Maquetacion interfaz HTML/CSS	3 días	lun 09/11/20	mié 11/11/20	25
30	✓	🚀	Desarrollo de la app	3 días	jue 12/11/20	lun 16/11/20	29
31	✓	📄	Elaboración del código app	2 días	lun 19/10/20	mar 20/10/20	
32	✓	📄	Conectión BD	2 días	vie 20/11/20	lun 23/11/20	
33	✓	🚀	Prueba 1	4 días	lun 23/11/20	jue 26/11/20	30
34	✓	📄	Correcciones de errores	1 día	lun 26/10/20	lun 26/10/20	
35	✓	📄	validar formulario carrito de compra	1 día	jue 26/11/20	jue 26/11/20	
36	✓	📄	Agregar al carrito de compra	1 día	jue 26/11/20	jue 26/11/20	
37	✓	📄	El botón de guardar detalle de pedido	1 día	jue 26/11/20	jue 26/11/20	
38		🚀	prueba 2	5 días	jue 26/11/20	mié 02/12/20	33
39	✓	📄	correcciones de errores	2 días	jue 26/11/20	vie 27/11/20	
40		📄	formulario dealle de pedido	2 días	vie 27/11/20	lun 30/11/20	
41	✓	📄	Eliminar producto en inventario	3 días	lun 30/11/20	mié 02/12/20	
42	✓	🚀	prueba 3	4 días	mar 10/11/20	vie 13/11/20	38
43	✓	📄	correcciones de errores	2 días	mar 10/11/20	mié 11/11/20	
44	✓	📄	Resta existencia del producto	1 día	mié 11/11/20	mié 11/11/20	
45	✓	📄	Buscar Producto	3 días	mié 11/11/20	vie 13/11/20	
46	✓	🚀	prueba 4	4 días	vie 13/11/20	mié 18/11/20	42
47	✓	📄	Correcciones de errores	1 día	vie 13/11/20	vie 13/11/20	
48	✓	📄	Registro de usuario	3 días	lun 16/11/20	mié 18/11/20	
49	✓	📄	Proyecto funcional COOPA	81 días	jue 27/08/20	jue 17/12/20	46










28. Bibliografía

- [1] R. Idalgo Sanz, «papel del farmaceutico en la venta y distribucion de medicamentos veterinario,» Mnagua, managua, 2020, p. 20.
- [2] A. Chetman, «Analisis para identificar oportunidad de mercado,» mexico, editorial R.L, 2017.
- [3] A. Trejos, «Tecnologia educativa,» mexico, editoriaer r.l, 2013.
- [4] «GENBETA,» [En línea]. Available: <https://www.genbeta.com/desarrollo/phonegap>.
- [5] «4rsoluciones,» 27 05 2015. [En línea]. Available: <https://www.4rsoluciones.com/blog/aplicaciones-nativas-con-tecnologia-web-ventajas-y-desventajas-de-phonegap/>.
- [6] «irontec,» [En línea]. Available: <https://www.irontec.com/cursos/curso-de-desarrollo-de-aplicaciones-moviles-con-ionic-framework>.
- [7] «fundacion Jesuites educacio,» 2018. [En línea]. Available: <https://fp.uoc.fje.edu/blog/desarrollo-de-aplicaciones-hibridas-con-entorno-ionic/>.
- [8] «DELOITTE,» [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-react-native.html>.
- [9] «assertsoft,» [En línea]. Available: <https://www.assertsoft.com/entrada/3-desventajas-apps-con-react-native/>.
- [10] «GENVETA,» [En línea]. Available: <https://www.genbeta.com/desarrollo/jquerymobile-introduccion-al-desarrollo-web-para-moviles>.
- [11] GENBETA. [En línea]. Available: <https://www.genbeta.com/desarrollo/jquerymobile-introduccion-al-desarrollo-web-para-moviles>.
- [12] «maquetando,» [En línea]. Available: <https://maquetando.com/framework-7>.
- [13] «Aula formativa,» [En línea]. Available: <https://blog.aulaformativa.com/principales-ventajas-y-desventajas-de-emplear-una-framework-css/>.
- [14] [En línea].
- [15] «desarrolloweb,» [En línea]. Available: <https://desarrolloweb.com/home/angular>.

- [16] [En línea].
- [17] «koaliti,» [En línea]. Available: <https://blog.koalite.com/2013/06/angularjs-lo-bueno-y-lo-malo/>.
- [18] E. Business, «Tecnología ágiles,» Diciembre 2017. [En línea]. Available: <https://www.esic.edu/rethink/tecnologia/metodologia-agile-por-que-necesitas-implementarla-en-tu-empresa>.
- [19] Jorge, «Mintakastar,» 09 05 2007. [En línea]. Available: <https://mintakastar.wordpress.com>.
- [20] EUDE, «EUDE Business SCHOOL,» 27 04 2018. [En línea]. Available: <https://eude.es/blog/responsabilidad-product-owner>.
- [21] J. D. Hoyos, «incentro,» 29 07 2020. [En línea]. Available: <https://www.incentro.com/es-es/blog/stories/que-significa-scrum-master>.
- [22] Sara, «saraclip.com,» 24 10 2017. [En línea]. Available: <https://www.saraclip.com/roles-en-scrum>.
- [23] E. Abellan, «Metodología Scrum,» 05 marzo 2020. [En línea]. Available: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html#>.
- [24] kanvan tool, «porque utilizar la metodologia kanvan,» [En línea]. Available: <https://kanbantool.com/es/metodologia-kanban>.
- [25] i. d. c. S.A, «inco,» [En línea]. Available: <https://www.incosa.com.uy/nuestra-empresa/>.
- [26] «metodo,» [En línea]. Available: <https://blog.metodogrupo.com/scrum-gestion-proyectos/>.
- [27] «openwebinars,» [En línea]. Available: <https://openwebinars.net/blog/scrum-caracteristicas/>.

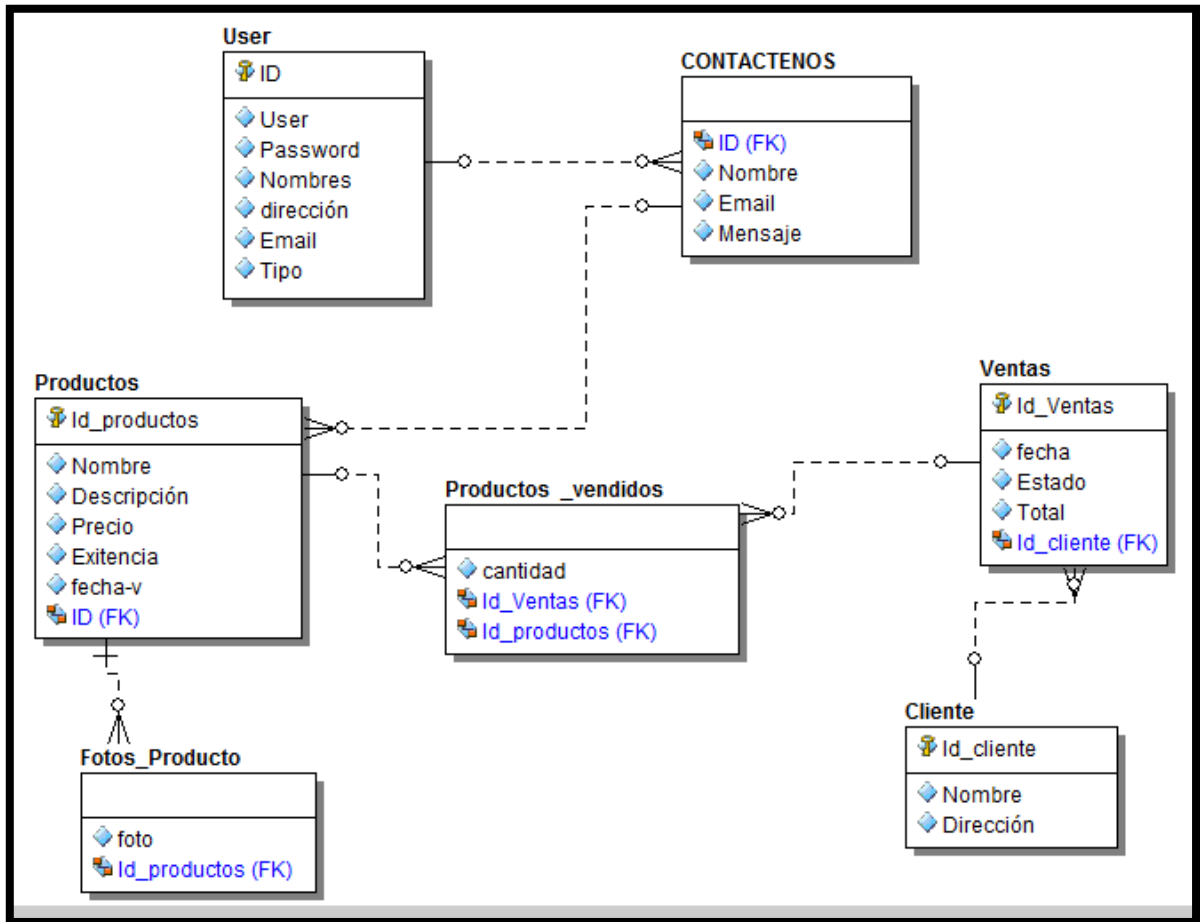
Anexos

Modelo canvas

<p>socios claves</p>  <ul style="list-style-type: none"> • Proveedor de Alojamiento para la aplicación web • Gestor de base de datos MariaDB 	<p>Actividades claves</p>  <ul style="list-style-type: none"> - Desarrollo de la App - Diseño de interfaz de usuarios administrador y cliente - Pruebas de software 	<p>Propuesta valor</p>  <ul style="list-style-type: none"> - Ventas en línea de productos veterinario de la farmacia COOPA -control de facturas - chat individual -pago en línea - Reportes de ventas 	<p>Relación con los clientes</p>  <p>-la relación con el cliente se dará mediante redes sociales como</p> <ul style="list-style-type: none"> -Facebook -WhatsApp 	<p>Segmento de cliente</p>  <ul style="list-style-type: none"> • Esta aplicación está dirigida a la farmacia veterinaria COOPA, y es adaptable a modelo de negocios con características similares. • De manera indirecta está dirigida a los clientes de dicha farmacia veterinaria 																			
<p>Estructura de costo</p>  <table border="1"> <thead> <tr> <th>Elementos</th> <th>costo</th> </tr> </thead> <tbody> <tr> <td>Equipo de desarrollo</td> <td>USD 1.800,00</td> </tr> <tr> <td>Tecnología</td> <td>USD 214.00</td> </tr> <tr> <td>Viatico</td> <td>USD 500.00</td> </tr> <tr> <td>Pruebas de software</td> <td>USD 138.00</td> </tr> <tr> <td>Total</td> <td>USD 2.652,00</td> </tr> </tbody> </table>		Elementos	costo	Equipo de desarrollo	USD 1.800,00	Tecnología	USD 214.00	Viatico	USD 500.00	Pruebas de software	USD 138.00	Total	USD 2.652,00	<p>Fuente de ingreso</p>  <table border="1"> <thead> <tr> <th colspan="2">Fuente de ingreso</th> </tr> </thead> <tbody> <tr> <td>Vender la aplicación</td> <td>\$800 por la compra</td> </tr> <tr> <td>Mantenimiento a la aplicación</td> <td>\$100 por mantenimiento</td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table>		Fuente de ingreso		Vender la aplicación	\$800 por la compra	Mantenimiento a la aplicación	\$100 por mantenimiento		
Elementos	costo																						
Equipo de desarrollo	USD 1.800,00																						
Tecnología	USD 214.00																						
Viatico	USD 500.00																						
Pruebas de software	USD 138.00																						
Total	USD 2.652,00																						
Fuente de ingreso																							
Vender la aplicación	\$800 por la compra																						
Mantenimiento a la aplicación	\$100 por mantenimiento																						

Base de datos

Diagrama entidad relación de la base de dato del proyecto (Farma-veterinaria)



Manual técnico



Guía técnica

**Desarrollo de una aplicación web para el control de ventas y facturación
para la farmacia veterinaria COOPA.**

**Versión 1.0
Diciembre 2020**

Índice de manual técnico

Objetivos.....	1
Objetivo general.....	1
Objetivos específicos	1
Introducción	1
Requerimientos técnicos.	1
Requerimientos mínimos de hardware.....	1
Requerimientos mínimos de software.....	2
Herramientas utilizadas para el desarrollo.....	2
Estructura de la aplicación web.	2
Método y autenticación de usuario para correo y contraseña.....	3
Conexión de la base de dato	3
Navegación:	3
Componentes funcionales	4
Componente de interfaz:	4
Estilos de la aplicación	5
Estado de pedidos:	5
Detalle de venta:	6
Pago en línea:	6
cambiar estado	7
Pago en línea	7
Chat individual:.....	8

Índice de ilustración

Ilustración 1: Autenticación de usuario	3
Ilustración: 2 Conexión de la base de datos	3
Ilustración 3: Navegación entre pantalla RouterLink	4
Ilustración 4: Componentes de interfaz.....	4
Ilustración: 5 Estilos de la aplicación	5
Ilustración 6: Estado de pedido.....	5
Ilustración 7: Detalle de ventas	6
Ilustración 8: detalles De ventas de producto	6
Ilustración 9: Cambiar estado.....	7
Ilustración 10: Pago en línea	7
Ilustración 11 Chat individual.....	8

Objetivos

Objetivo general

- Facilitar la comprensión técnica de la aplicación web

Objetivos específicos

- Desarrollar los requerimientos de hardware y software para el funcionamiento de la aplicación.
- Conocer las herramientas que se utilizó en la aplicación

Introducción

Este manual proporciona al lector la lógica con que se desarrolló la aplicación web, esta es propia de cada programador y se considera necesaria ser documentada. Se describe la información de los recursos utilizado en este proyecto y las características físicas y técnicas de cada elemento.

El manual no pretende ser un curso de aprendizaje de cada una de las herramientas utilizadas para el desarrollo del sitio, si no documentar su aplicación de desarrollo. Para mayor detalle acerca de cada una de las herramientas utilizadas, y su forma de operación y aplicación, se recomienda consultar los manuales respectivos de cada uno de ellos.

Se utiliza la herramienta Angular Cli para acelerar el desarrollo, mientras se adhiere a las recomendaciones de la guía de estilo que benefician a todos los proyectos de Angular.

Requerimientos técnicos.

Requerimientos mínimos de hardware

- **Procesador:** AMD E1-6015 with radion (TM) R2 Graphics 1.40GHz

- **Memoria RAM:** 4,00 GB
- **Tipo de sistema:** sistema operativo de 64 bits procesador x 64
- **Disco duro:** 465,64 GB

Requerimientos mínimos de software

- Conexión a internet.
- Cualquier sistema operativo desde XP hasta Windows 10
- Navegadores: Chrome, Firefox, Opera, Safari, Internet Explorer

Herramientas utilizadas para el desarrollo

La aplicación está basada en arquitectura web, utilizando la plataforma y el Framework Angular utilizada para escribir aplicaciones web en HTML (Hyper Text Markup Language), TypeScript, como editor de texto Visual Studio Code, y para el proceso de gestión MySQL

Node Js: Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript

Una de la gran ventaja de utilizar NodeJS es que trae integrado NodePackage Manager(NPM), esto facilita al desarrollador compartir y reutilizar código. En el desarrollo del proyecto NPM será utilizado para instalar la dependencia necesaria para el correcto funcionamiento de la aplicación de una manera sencilla y rápida utilizando en comando “npm install”.

MariaDB: permite a los desarrolladores y diseñadores realizar cambios en los sitios web con sólo cambiar un archivo, (sin necesidad de modificar todo el código web) para que se ejecuten en toda la estructura de datos que se comparte en la red. [25, p. 67]

Framework Angular: Se usa este framework porque permite crear aplicaciones web desarrolladas en TypeScript y HTML de manera dinámica y moderna además es de código abierto y consta con arquitectura (MVC), modelo, vista, control.

Estructura de la aplicación web.

La arquitectura que propone el framework de angular es: programación orientada a módulos, agrupación de componentes que se pertenecen.

El concepto de componentes es clave en Angular y se define como la interfaz de usuario que tiene funcionalidades y estado.

En cuanto al modelo de datos se usan los servicios que ofrece María DB, para la autenticación de usuario.

Método y autenticación de usuario para correo y contraseña.

```
<div class="login-wrapper" fxLayout="row" fxLayoutAlign="center center" style="display: flex; justify-content: center; align-items: center !important">
  <mat-card class="box">
    <mat-card-header>
      <mat-card-title>Inicio de sesión</mat-card-title>
      <br>
    </mat-card-header>
    <mat-card-content class="example-form">
      <mat-form-field class="example-full-width">
        <input matInput [(ngModel)]="txtuser" placeholder="Usuario" >
      </mat-form-field>
      <mat-form-field class="example-full-width">
        <input matInput [(ngModel)]="txtpassword" placeholder="Contraseña" >
      </mat-form-field>
    </mat-card-content>
    <button mat-stroked-button color="accent" class="btn-block" (click)="iniciarsesion()">Entrar</button>
  </mat-card>
</div>
```

Ilustración 1: Autenticación de usuario

Elaboración propia

Conexión de la base de dato

```
module.exports = mysql.createPool({
  host: 'localhost',
  user: 'root',
  password: 'root',
  database: 'cartbd',
  port: 3306
});
```

Ilustración: 2 Conexión de la base de datos

Elaboración propia

Navegación: para la creación de la navegación de la aplicación se empleó RouterLink, esta es una librería que permite crear rutas para navegar en los diferentes componentes de la aplicación

Como se puede observar se realiza la navegación entre los diferentes componentes de la aplicación que contiene el menú

```
<div class="contenedor-padre">
  <mat-toolbar class="barra" color="primary" >
    <mat-toolbar-row>
      <a routerLink="/tienda" style="color: □white; margin: 10px;" >
        <mat-icon color="primary" style="color: □white;"> home</mat-icon>
        Inicio
      </a>
      <a routerLink="/quienes" style="color: □white; margin: 10px;" >
        <mat-icon color="primary" style="color: □white;"> contact_support</mat-icon>
        ¿Quiénes somos?
      </a>
      <a routerLink="/contactenos" style="color: □white; margin: 10px;" >
        <mat-icon color="primary" style="color: □white;"> contact_mail</mat-icon>
        Contáctenos
      </a>
    </mat-toolbar-row>
  </mat-toolbar>
</div>
```

Ilustración 3: Navegación entre pantalla RouterLink

Elaboración propia

Componentes funcionales: se muestran los componentes funcionales que integra la aplicación web.

Componente de interfaz: en la siguiente imagen se observa la estructura de componente de interfaz padre, su estructura es la siguiente, importaciones de otros componentes, e importaciones de clases

```
import {NgModule} from '@angular/core';
import {Routes, RouterModule} from '@angular/router';
import {ProductosComponent} from './productos/productos.component';
import {ClientesComponent} from './clientes/clientes.component';
import {VentasComponent} from './ventas/ventas.component';
import {TiendaComponent} from './tienda/tienda.component';
import {AgregarProductoComponent} from './agregar-producto/agregar-producto.component';
import {DetalleDeProductoComponent} from './detalle-de-producto/detalle-de-producto.component';
import {TerminarCompraComponent} from './terminar-compra/terminar-compra.component';
import {DetalleDeVentaComponent} from './detalle-de-venta/detalle-de-venta.component';
import { ContactenosComponent } from './contactenos/contactenos.component';
import { AdminComponent } from './admin/admin.component';
import { QuienesComponent } from './quienes/quienes.component';
import { ReporteventasComponent } from './reporteventas/reporteventas.component';
import { CambiarEstadoComponent } from './cambiar-estado/cambiar-estado.component';
import { LoginComponent } from './login/login.component';
import { ListcontactoComponent } from './listcontacto/listcontacto.component';
```

Ilustración 4: Componentes de interfaz

Elaboración propia

Estilos de la aplicación: Esta imagen muestra alguno de los estilos utilizado en el componente padre de la aplicación

```

.contenedor-padre {
  display: flex;
  flex-flow: column;
  height: 100%;
}

.barra {
  flex: 0 1 auto;
}

.contenido {
  flex: 1 1 auto;
}

.padding-10 {
  padding: 10px;
}

.example-fill-remaining-space {
  /* This fills the remaining space, by using flexbox.
  | | Every toolbar row uses a flexbox row layout. */
  flex: 1 1 auto;
}

```

Ilustración: 5 Estilos de la aplicación

Elaboración propia

Estado de pedidos: Se muestra la función de administrar estado de los pedidos realizados en la aplicación.

```

<div class="contenedor-padre">
<mat-toolbar class="barra" color="primary">
  <mat-toolbar-row>
    <button (click)="cajon.toggle()" mat-icon-button>
      | <mat-icon>menu</mat-icon>
    </button>
    <span>
      | Farmacia veterinaria&nbsp;&||&nbsp;&Coopa
    </span>
    <span class="example-fill-remaining-space"></span>
    <button [matMenuTriggerFor]="menu" style="border-style:none !important; background: transparent; color:□white">
      <mat-icon>shopping_cart</mat-icon>
      <span style="margin-right: 10px; padding:0px; margin-top: 0px !important;" [matBadge]="productos.length+''" matBadgeOverlap="false"
      | | ></span>
    </button>
    <mat-menu #menu="matMenu">
      <button mat-menu-item style="height: 60px !important; line-height: 15px; border-bottom:1px solid ■dimgray !important;" *ngFor="let p of productos">
        <label style="color: ■#3f51b5">
          | <b>{{p.nombre}}</b>
        </label>
        <br>
        <label><b>Precio:</b> </> {{p.precio | currency:'C$':true:'1.2-2'}}</label>
        <br>
        <label><b>Cantidad:</b> </> {{p.cantidad}}</label>
        <br>
      </button>
    </mat-menu>
    <a style="margin-left: 10px" mat-raised-button color="primary"
      | href="#" target="_blank">
      | Iniciar sesión
    </a>
  </mat-toolbar-row>
</mat-toolbar>
<mat-sidenav-container class="contenido">
<mat-sidenav style="min-width: 300px;" #cajon opened mode="side">
  <mat-nav-list>
    <n style="margin-left: 4px;">Control de Administrador</n>

```

Ilustración 6: Estado de pedido

Elaboración propia

Detalle de venta: La imagen siguiente muestra el código para el componente detalle de ventas, la cual contiene los datos necesarios para el desarrollo de la interfaz que se muestra al cliente

Pago en línea: Funcionalidad del pago en línea, está estructurado con una variable que controla en pago en línea de Paypal e importaciones de clases y métodos de los componentes que depende la funcionalidad pago en línea.

```
import {Component, OnInit} from '@angular/core';
import {VentasService} from "../ventas.service";
import {ActivatedRoute} from "@angular/router";

@Component({
  selector: 'app-detalle-de-venta',
  templateUrl: './detalle-de-venta.component.html',
  styleUrls: ['./detalle-de-venta.component.css']
})
export class DetalleDeVentaComponent implements OnInit {

  constructor(private ventasService: VentasService, private activatedRoute: ActivatedRoute) {
  }

  public venta = {
    total: 0,
    nombre: "",
    direccion: "",
    productos: [],
  };

  public columnas = ['nombre', 'precio', 'cantidad', 'subtotal'];

  async ngOnInit() {
    const id = this.activatedRoute.snapshot.paramMap.get("id")
    this.venta = await this.ventasService.obtenerDetalleDeVenta(id);
  }
}
```

Ilustración 7: Detalle de ventas

Elaboración propia

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
import { AdminComponent } from './admin.component';
import { RouterTestingModule } from '@angular/router/testing';

describe('AdminComponent', () => {
  let component: AdminComponent;
  let fixture: ComponentFixture<AdminComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ AdminComponent ]
    })
    .compileComponents();
  }));

  beforeEach(() => {
    fixture = TestBed.createComponent(AdminComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Ilustración 8: detalles De ventas de producto

Elaboración propia

cambiar estado

```
| | <h1>Cambiar estado</h1>
<br>
<h2>Cliente: {{nombre}} </h2>
<h2>Fecha: {{fecha | date: 'shortDate'}} </h2>
<form>
  <mat-form-field appearance="fill">
    <mat-label>Cambiar estado</mat-label>
    <select matNativeControl [(ngModel)]="ventaModel.estado" name="car">
      <option *ngFor="let car of cars" [value]="car.value">
        {{car.viewValue}}
      </option>
    </select>
  </mat-form-field>
</form>

<app-loading-button texto="Actualizar estado" (click)="actualizarestado()"></app-loading-button>
<a routerLink="/ventas" mat-raised-button color="primary" style="margin-left: 10px">Volver</a>
```

Ilustración 9: Cambiar estado

Elaboración propia

Pago en línea

```
import {Component, OnInit, ViewChild, ElementRef} from '@angular/core';
import {CarritoService} from "../carrito.service";
import {DataSharingService} from "../data-sharing.service";
import {Cliente} from "../cliente";
import {MatStepper} from '@angular/material/stepper';
import Swal from 'sweetalert2';
import { StepperSelectionEvent } from '@angular/cdk/stepper';
import { CookieService } from 'ngx-cookie-service';
import {Router} from "@angular/router";

declare var paypal; //variable que controlara el boton y pagos de paypal

@Component({
  selector: 'app-terminar-compra',
  templateUrl: './terminar-compra.component.html',
  styleUrls: ['./terminar-compra.component.css']
})
export class TerminarCompraComponent implements OnInit {
  @ViewChild('paypal', {static: true}) paypalElement : ElementRef;
  @ViewChild('stepper') stepper;

  public txtcliente = "";
  public txtdireccion = "";
  public txtestado = "";
```

Ilustración 10: Pago en línea

Elaboración propia

Chat individual:

Para realizar la programación del chat en línea, se utilizó de Messenger, se realizó la petición para tener acceso a esta tecnología, cumpliendo con los siguientes pasos:

- Realizar la petición a Facebook.
- Enviar el dominio de la página que está en línea en este caso <http://coopa.site>.
- Luego ellos nos mandan un enlace y el código que necesitamos para programarlo en el proyecto.

```
window.fbAsyncInit = function() {
  FB.init({
    xfbml      : true,
    version    : 'v9.0'
  });
};

(function(d, s, id) {
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) return;
  js = d.createElement(s); js.id = id;
  js.src = 'https://connect.facebook.net/es_LA/sdk/xfbml.customerchat.js';
  fjs.parentNode.insertBefore(js, fjs);
})(document, 'script', 'facebook-jssdk');
```

Ilustración 11 Chat individual

Elaboración propia

Manual de usuario



Manual de usuario

Desarrollo de una aplicación web para el control de ventas y facturación para la farmacia veterinaria COOPA

Versión 1.0

Diciembre 2020

Índice de manual de usuario

Introducción	1
Requerimientos	1
Requerimientos de Hardware.....	1
Requerimientos de Software	1
Registro y sesión de usuario	2
Inicio y panel de navegaciones	3
funcionalidades de usuario cliente	4
carrito de compras.....	4
Modo de pago	5
Proceso de facturación	6
Compra realizada	8
Funcionalidad del usuario administrador	8
Registrar productos en inventario	8
Mostrar productos guardados.....	9
ventas realizadas	10
Detalle del producto vendido	10
Reporte de venta	11
cambiar estado de los productos	11
Formulario de políticas de envío y políticas de devoluciones	12
Chat individual.....	13
Proceso de chat con el cliente	14
Facturación.....	14

Índice de ilustración

Ilustración 1: Registro de cliente	2
Ilustración 2: Inicio de sesión	2
Ilustración 3: Inicio y panel de navegación	3
Ilustración 4: Panel de navegación del administrador	3
Ilustración 5: Panel de navegación del cliente	4
Ilustración 6: Opciones de carrito de compra	4
Ilustración 7: Proceso de compra	5
Ilustración 8: Modo de pago	5
Ilustración 9: Modo de pago en PayPal	6
Ilustración 10: Ingresar cuenta	6
Ilustración 11: Ingresar usuario y contraseña	7
Ilustración 12: Factura de PayPal	7
Ilustración 13: Compras realizadas	8
Ilustración 14: Registro de productos	8
Ilustración 15: Registro de productos	9
Ilustración 16: Productos en inventario	9
Ilustración 17: Ventas realizadas	10
Ilustración 18: Detalle de ventas	10
Ilustración 19: Reportes de ventas	11
Ilustración 20: Cambiar estado	11
Ilustración 21: Contáctenos	12
Ilustración 22: Política de envío y devoluciones	12
Ilustración 23: Burbuja de chat	13
Ilustración 24: Chatear con COOPA	13
Ilustración 25: Proceso de chat con el cliente	14
Ilustración 26: Facturación	14

Introducción

El manual pretende otorgar soporte para el uso y funcionamiento de los usuarios de la aplicación web

Esta aplicación ha sido desarrollada con el objetivo de aportar al conocimiento de los interesados (propietario de farmacias veterinarias), de manera que puedan mejorar sus ventas y establecer una mejor relación con el cliente, y así generar mejores ganancias.

Requerimientos

Para el correcto funcionamiento de la aplicación web, es necesario de requisitos de hardware y de software

Requerimientos de Hardware

- **Procesador:** AMD E1-6015 with radion (TM) R2 Graphics 1.40GHz
- **Memoria RAM:** 4, 00 GB
- **Tipo de sistema:** sistema operativo de 64 bits procesador x 64
- **Disco duro:** 465,64

Requerimientos de Software

- Conexión a internet
- Cualquier sistema operativo desde XP hasta Windows 10
- Navegadores: Chrome, Firefox, Opera, Safari, Internet Explorer

Registro y sesión de usuario

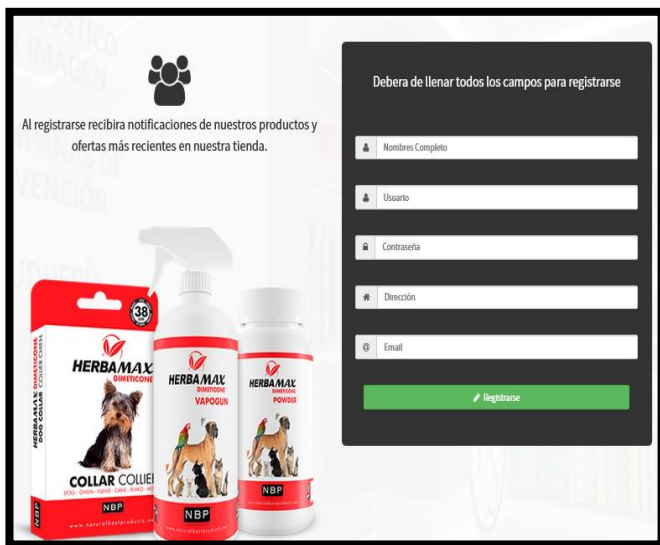


Ilustración 1: Registro de cliente

elaboración propia

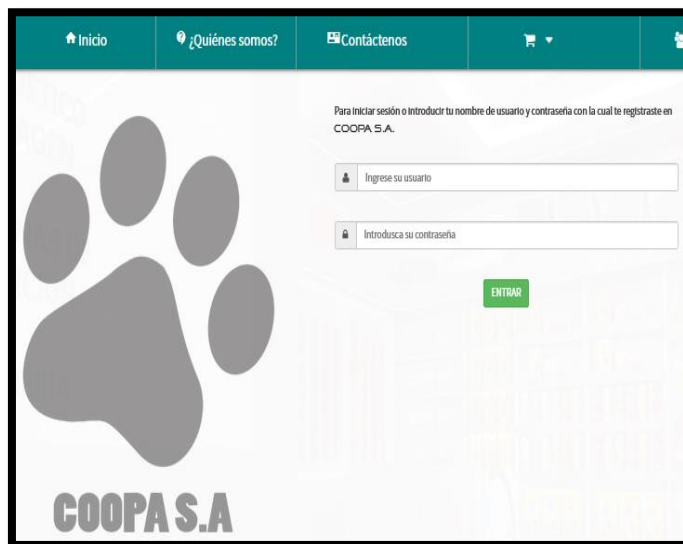


Ilustración 2: Inicio de sesión

elaboración propia

La interfaz de la aplicación ofrece una vista para el acceso de la misma mediante un Login, pero antes el usuario requiere un registro para ingresar a la aplicación, por lo tanto, se ofrece una pantalla con los datos necesarios, una vez lleno los datos deberá presionar **registrarse** y el usuario será registrado en la aplicación.

Una vez el usuario registrado, deberá proporcionar un usuario y una contraseña, para acceder a la aplicación, luego debe presionar el botón **entrar**

Inicio y panel de navegaciones

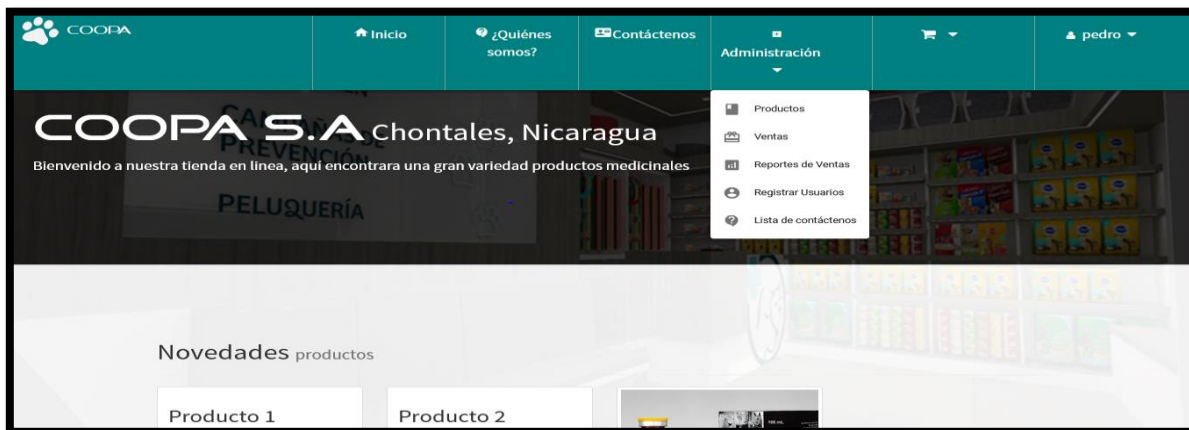


Ilustración 3: Inicio y panel de navegación

Elaboración propia

Una vez el usuario haya sido autenticado podrá tener acceso al inicio y toda las funcionalidades de la aplicación web. En esta imagen muestra el panel de navegación del usuario administrador.

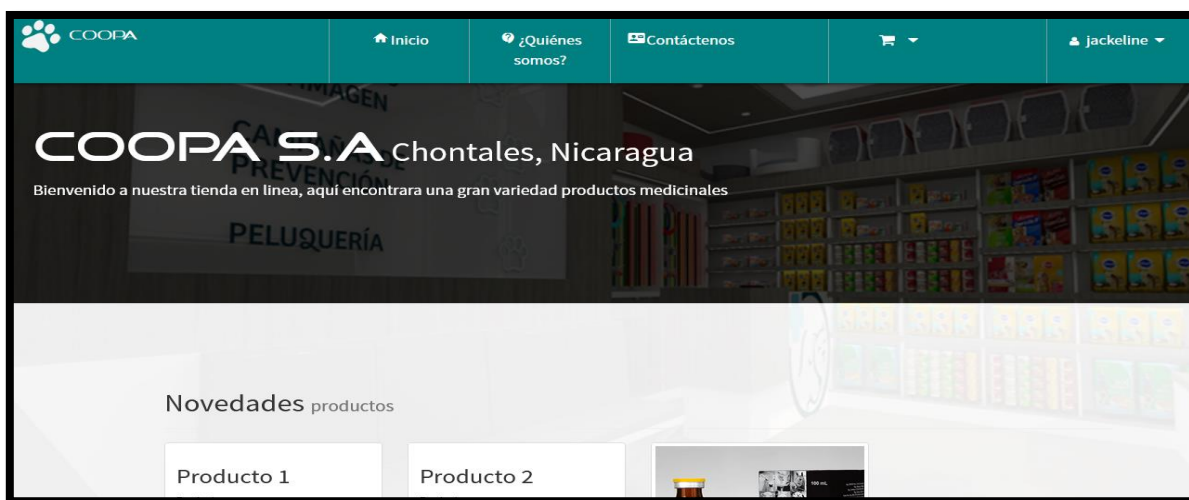


Ilustración 4: Panel de navegación del administrador

Elaboración propia

Luego que el usuario cliente inicia sesión, puede tener acceso a las opciones de navegación las cuales son: inicio donde se encuentra la tienda de productos, quiénes somos que es la información de la empresa, contactenos, carrito de compra, chat, cerrar sesión

funcionalidades de usuario cliente

carrito de compras

con esta opción el usuario puede ver los productos veterinarios que están en tienda, y que han sido agregado al carrito de compra.

Ilustración 5: Panel de navegación del cliente

Elaboración propia



Luego que el usuario halla añadido un producto al carrito de compra, tiene las siguientes opciones.



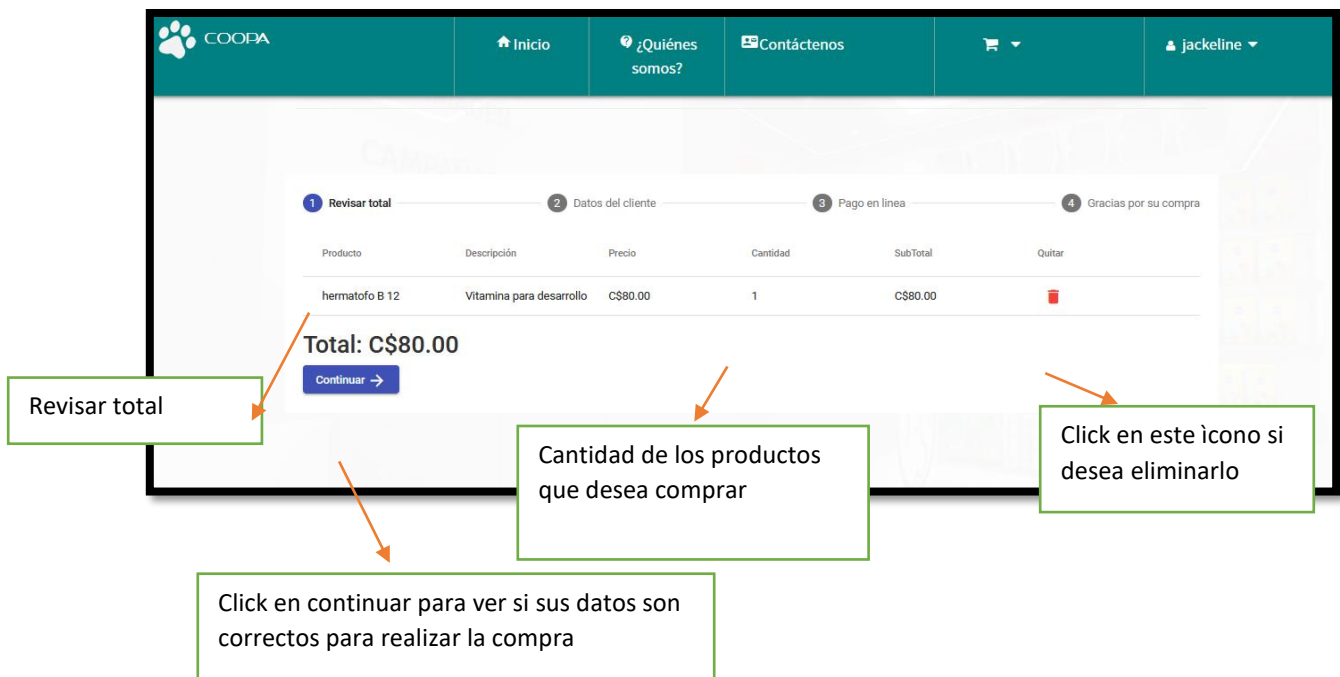
Ilustración 6: Opciones de carrito de compra

Elaboración propia

después de a ver dado click en terminar de comprar muestra el siguiente formulario, que permite mostrar paso a paso el proceso de compras, el usuario para realizar su compra es necesario que lea la información de políticas de envío, política de pago, y política de devoluciones que se encuentra en la página principal de la aplicación web.

Ilustración 7: Proceso de compra

Elaboración propia



Modo de pago

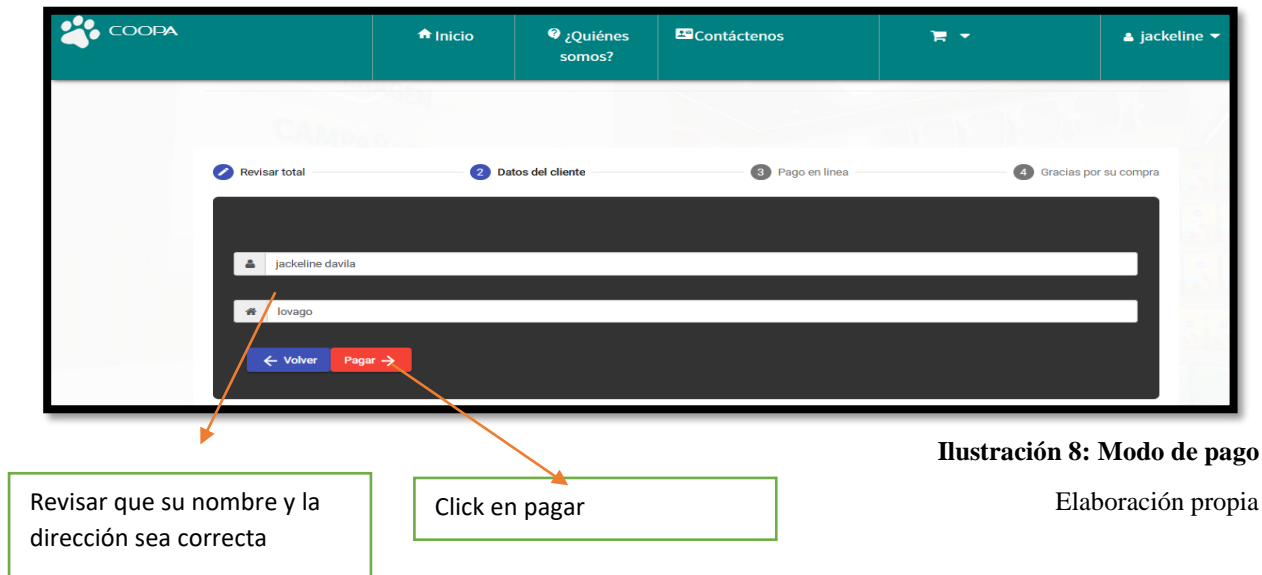


Ilustración 8: Modo de pago

Elaboración propia

Proceso de facturación

Luego de realizar su compra, procede a pagar en línea, en este caso el pago se realiza en PayPal, ya que brinda total seguridad al momento de comprar por internet, lo que hace es enlazar tu tarjeta con la cuenta que creaste en PayPal, de esa forma no tendrás que volver a ingresar los datos de tu tarjeta, lo único que tendrás que hacer para efectuar el pago es colocar tu usuario y contraseña de PayPal, de esa manera evitamos el robo de datos de nuestra tarjeta.

Por otro lado, PayPal, otorga 60 días de garantía para cualquier producto que compremos, es decir, si por un motivo no recibe el producto, puede solicitar la devolución total de su inversión.

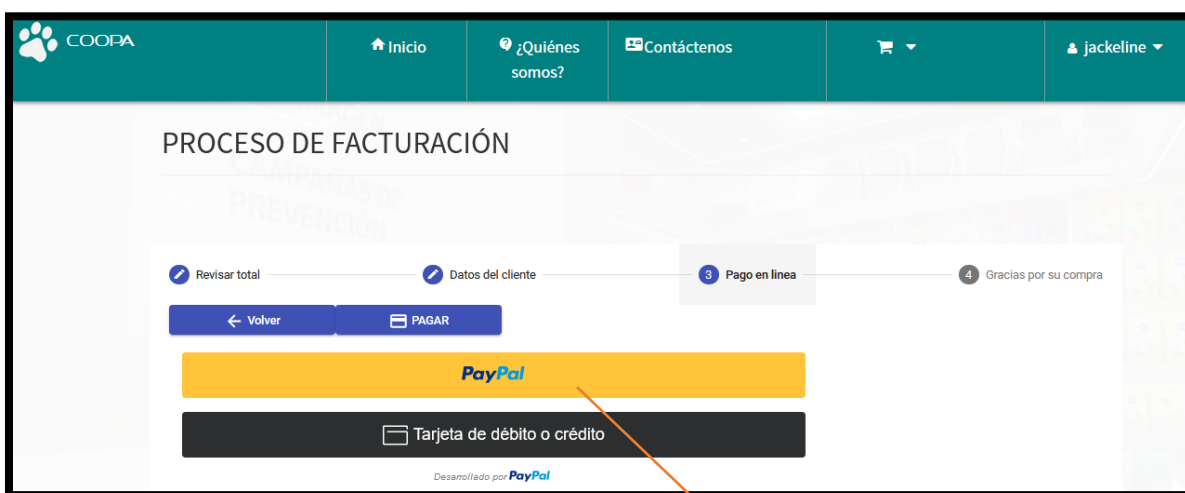


Ilustración 9: Modo de pago en PayPal

Elaboración propia

Click en PayPal

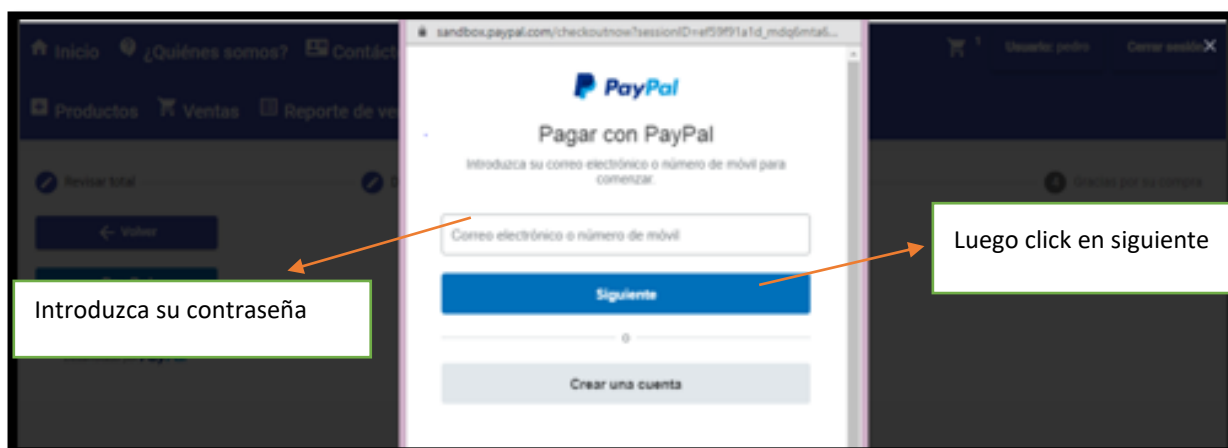


Ilustración 10: Ingresar cuenta

Elaboración propia

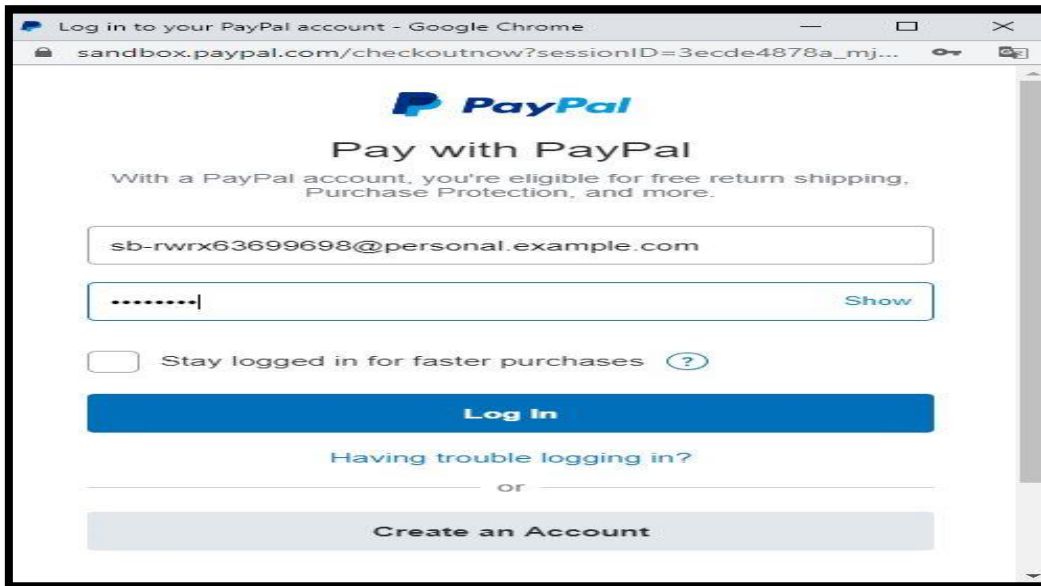


Ilustración 11: Ingresar usuario y contraseña

Elaboración propia

Esta es la factura que proporciona PayPal.

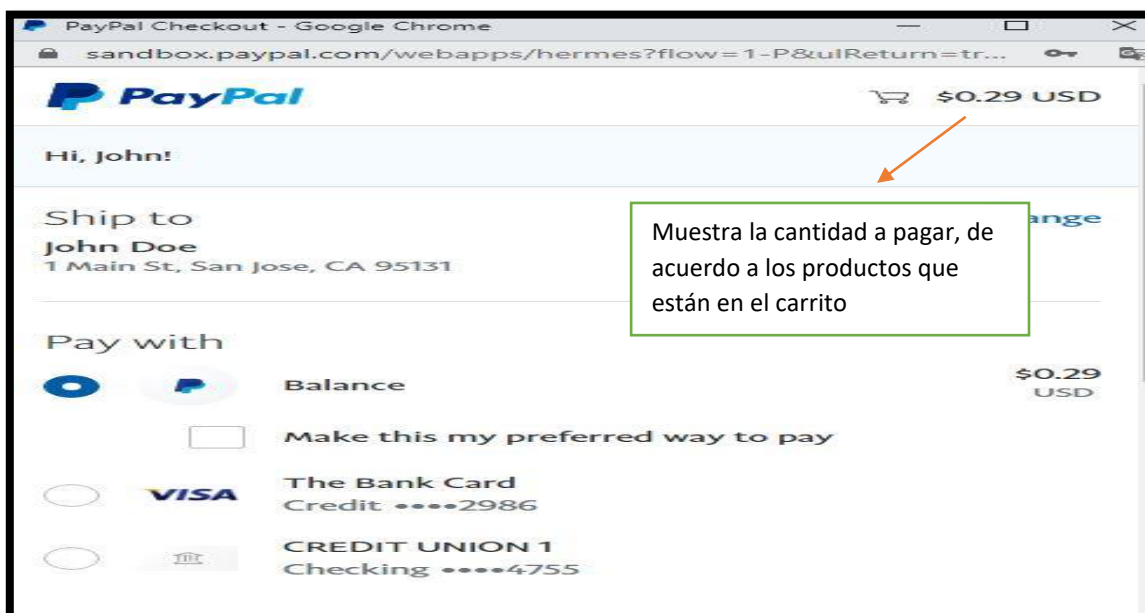


Ilustración 12: Factura de PayPal

Elaboración propia

Compra realizada



Ilustración 13: Compra realizadas

Elaboración propia

Funcionalidad del usuario administrador

Registrar productos en inventario

El usuario administrador podrá agregar nuevos productos en inventario

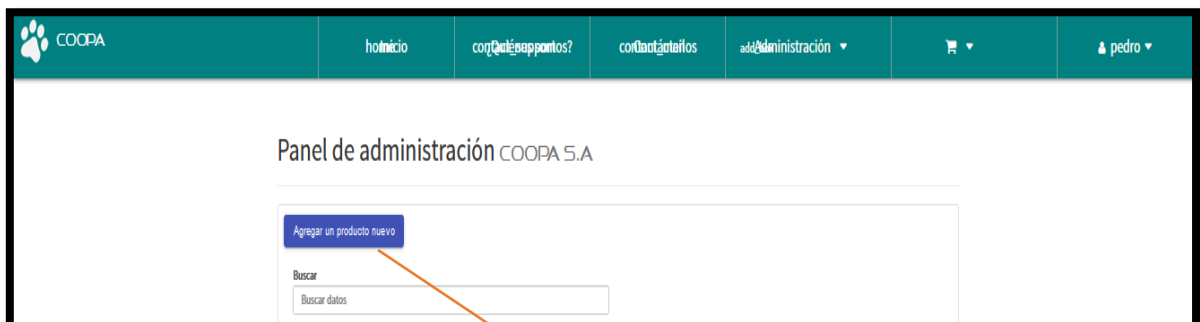


Ilustración 14: Registro de productos

Elaboración propia

Click en agregar producto, luego se desplegará un formulario con los datos correspondiente, para registrar un nuevo producto.

En este formulario podrá registrar un nuevo producto

Panel de administración COOPA S.A.

+ Agregar un producto nuevo

Nombre de producto

Nombre

Descripción

Descripción (301 caracteres)

Fecha de Vencimiento

dd / mm / aaaa

Precio

Existencia

o

Imagen de producto

Examinar... No se han seleccionado archivos.

Formato de imágenes admitido png, jpg, gif, jpeg

Guardar Volver

Click en examinar para agregar imagen

Luego click en guardar

Ilustración 15: Registro de productos

Elaboración propia

Mostrar productos guardados

Panel de administración COOPA S.A.

Agregar un producto nuevo

Buscar

Buscar datos

Nombre	Descripción	Precio	Existencia	Fecha de Vencimiento	Modificar	Eliminar
Producto 1	Producto 1	C\$100.00	18	2025-11-09	edit	delete
Producto 2	Producto 2	C\$300.00	9	2030-02-09	edit	delete
Vitamina para desarrollo	C\$80.00	598	2022-05-08	edit	delete	
es una de los mejores desparacitantes	C\$160.00	199	2023-02-10	edit	delete	

Buscar por fecha o nombre el producto

Eliminar

Editar

Ilustración 16: Productos en inventario

Elaboración propia

Ventas realizadas: En este formulario el usuario administrador puede observar las ventas realizadas.

The screenshot shows a web interface for managing sales. At the top, there is a search bar labeled 'Buscar' with the placeholder text 'Buscar datos'. Below it is a table with the following columns: 'Cliente', 'Dirección de envío', 'Total', 'Fecha', 'Estado', and 'Detalles'. The table contains eight rows of sales data. Three callout boxes with arrows point to specific elements: 'Buscar por fecha las ventas' points to the search bar; 'Revisar estado' points to the 'Estado' column; and 'Detalles de venta' points to the 'Detalles' column.

Cliente	Dirección de envío	Total	Fecha	Estado	Detalles
Juan Manuel	Hospital 3 al sur	CS\$2.597.00	1/21/21	Entregado	[Q]
Teodoro Ruiz	inta 5 al 2 oeste	CS\$200.00	1/22/21	En espera	[Q]
Jackeline Dávila Oporta	Lovago	CS\$150.00	1/22/21	En espera	[Q]
Bryan davila	acoyapa	CS\$150.00	1/22/21	En espera	[Q]
Jackeline Dávila Oporta	Lovago	CS\$350.00	1/22/21	En espera	[Q]
Jackeline Dávila Oporta	Lovago	CS\$150.00	1/28/21	En espera	[Q]
Jackeline Dávila Oporta	Lovago	CS\$500.00	2/3/21	En espera	[Q]
		CS\$800.00	2/3/21	En espera	[Q]

Buscar por fecha las ventas

Revisar estado

Detalles de venta

Ilustración 17: Ventas realizadas

Elaboración propia

Detalle del producto vendido

Al dar click en el botón detalle de la venta se desplegará el siguiente formulario con la información del cliente y del producto

The screenshot shows a detail view of a sale. At the top, there is a navigation bar with links: 'Inicio', '¿Quiénes somos?', 'Contáctenos', and 'Administración'. Below the navigation bar, the customer information is displayed: 'pedro gonzalez' (with a person icon), 'ninguna' (with a truck icon), and '\$3,100.00' (with a dollar sign icon). Below this is a table with the following columns: 'Nombre', 'Precio', 'Cantidad', and 'Subtotal'. The table contains two rows of product information. At the bottom left, there is a 'Volver' button.

Nombre	Precio	Cantidad	Subtotal
Producto 1	CS\$100.00	CS\$1.00	CS\$100.00
Producto 2	CS\$300.00	CS\$10.00	CS\$3,000.00

Volver

Ilustración 18: Detalle de ventas

Elaboración propia

Reporte de venta

Cliente	Dirección de envío	Total	Fecha/vta
pedro gonzalez	ninguna	C\$3,100.00	1/9/21
pedro gonzalez	ninguna	C\$300.00	1/22/21
jackeline davila	lovago	C\$100.00	2/1/21
jackeline davila	lovago	C\$0.00	2/1/21
jackeline davila	lovago	C\$0.00	2/1/21
jackeline davila	lovago	C\$80.00	2/3/21
jackeline davila	lovago	C\$80.00	2/8/21
jackeline davila	lovago	C\$160.00	2/9/21

Ilustración 19: Reportes de ventas

Elaboración propia

cambiar estado de los productos

Inicio

Cambiar estado

Cliente: pedro gonzalez
Fecha: 1/9/21

Cambiar estado
En espera

Actualizar estado Volver

Ilustración 20: Cambiar estado

Elaboración propia

La aplicación web cuenta con un área de contáctenos, la que permite que el usuario administrador ver lista de los usuarios que han enviado sugerencia a la aplicación web.

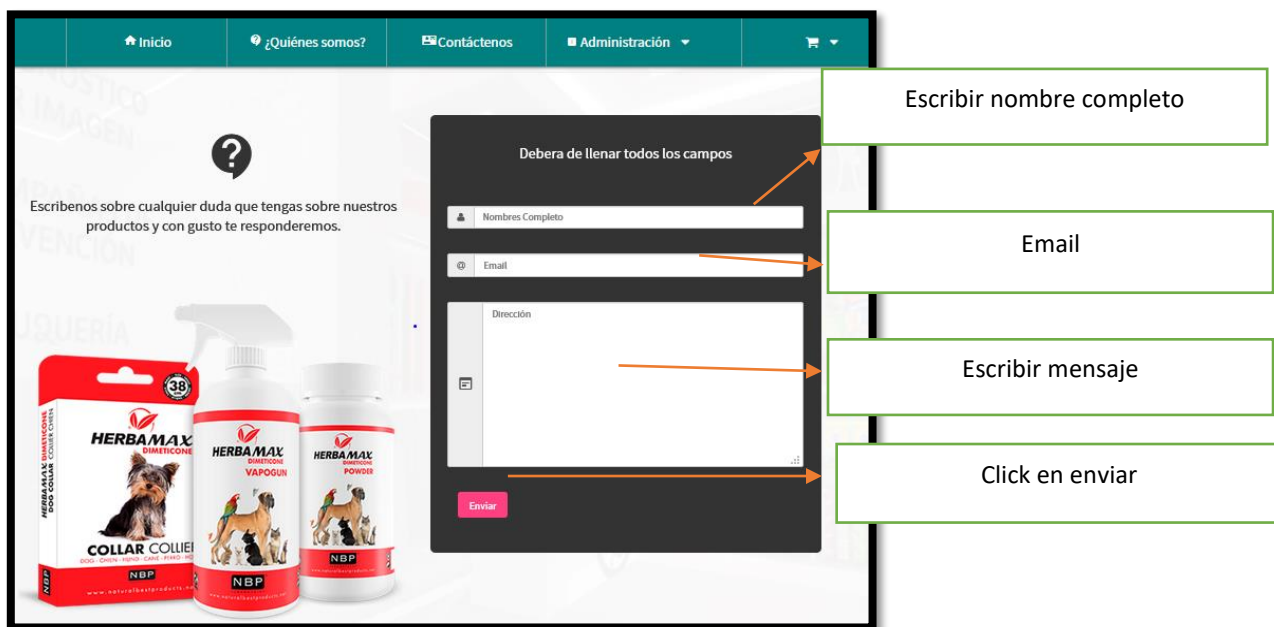


Ilustración 21: Contáctenos

Elaboración propia

Formulario de políticas de envío y políticas de devoluciones

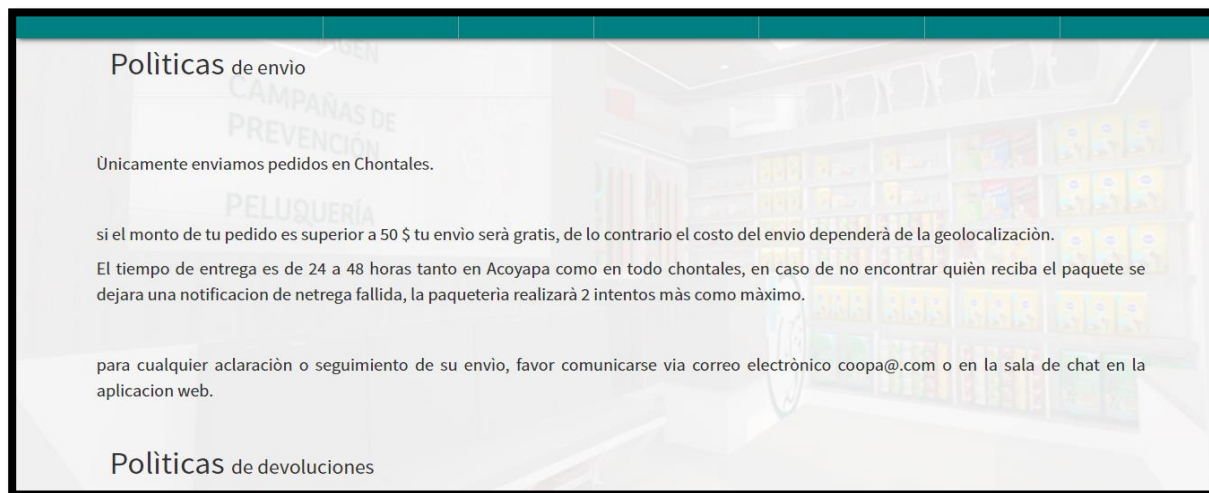


Ilustración 22: Política de envío y devoluciones

Elaboración propia

Chat individual

En el formulario principal, en la parte inferior derecha muestra el icono de chat, si quieres comunicarte con la farmacia, puedes enviar un mensaje



Ilustración 23: Burbuja de chat

Elaboración propia

Click en el icono

luego de dar click en el icono de chat, se desplegará el siguiente formulario



Continuar el chat como usuario registrado de Messenger

También está la opción de continuar como invitado

Ilustración 24: Chatear con COOPA

Elaboración propia

Proceso de chat con el cliente

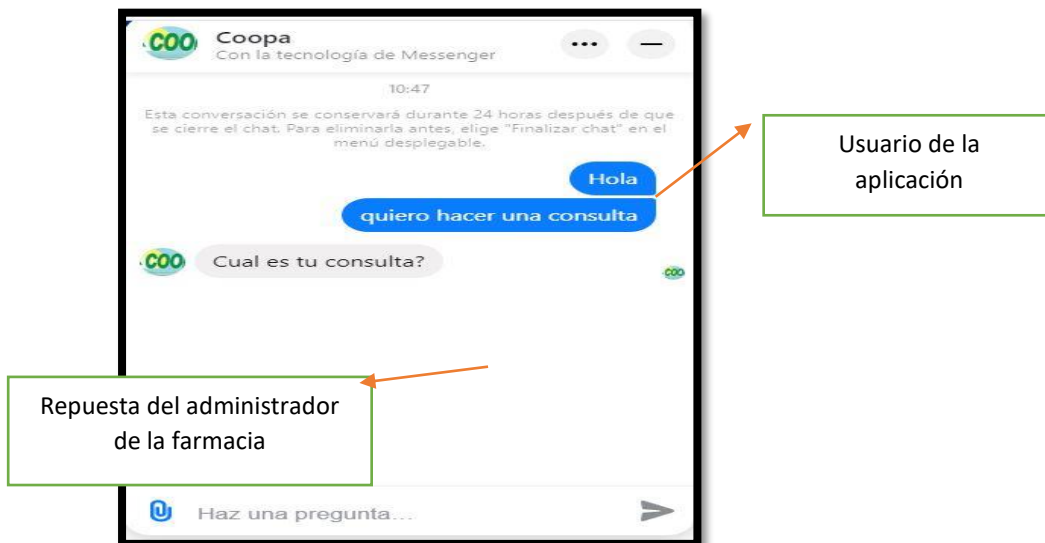


Ilustración 25: Proceso de chat con el cliente

Elaboración propia

Facturación

luego de realizar la compra al cliente se le mostrará que la factura con los datos, el cliente tendrá la opción de imprimir su factura.

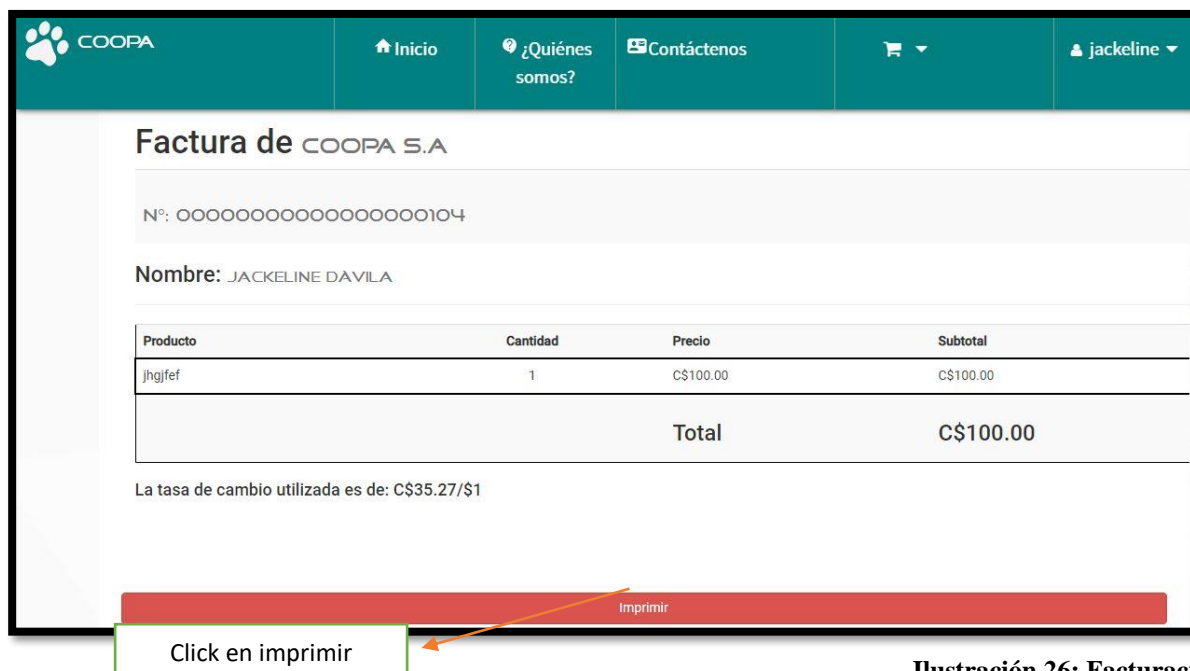


Ilustración 26: Facturación

Elaboración propia



Cooperativo de servicios
agropecuarios

PLAN DE PRUEBAS (Farm-veterinaria)



PLAN DE PRUEBAS

Cooperativa de servicios agropecuarios

Juigalpa, Chontales, diciembre 2020



Bitácora de cambio

Fecha	Modificado por	Versión	Descripción
06/11/2020	Br. Jackeline Dávila	0.1	Inicio del plan de prueba para el proyecto Farm-veterinaria planteando los casos de prueba para cada sprint
12/11/2020	Br. Jackeline Dávila	0.1	selección y revisión de los tipos de pruebas para los componentes, y la funcionalidad de cada componente de Farm-veterinaria
12/12/2020	Br. Jackeline Dávila	0.1	Realizar pruebas de caja blanca



Formato preliminar al documento

Título:	Plan de prueba Farm-veterinaria		
ID:	Plan de prueba_ 01		
Fecha elaboración:	06/11/2020		
Sumario	El objetivo del plan de pruebas es verificar que los módulos, componentes y funcionalidades de la aplicación web funcionen correctamente, así como también detectar y corregir fallos encontrado durante el periodo de prueba.		
Palabras Claves:	Pruebas, fallos, aplicación, componente, requerimiento, funcionamiento.		
Formato:	Documento	Lenguaje:	Español
Dependencia:	COOPA		
Versión:	0.2	Estado :	Se colocará el estatus del plan de prueba, puede ser aprobado o no aprobado
Autor (es):	Br. Jackeline Margarita Dávila Oporta Br. Keylin López	Firma s:	
Revisó:	Msc. Saira Urbina		
Aprobó:	Msc. Saira Urbina		
Información Adicional:			



Tabla de contenido

1 introducción	1
1.1 Objetivo general.....	1
1.2Estrategia de pruebas	1
1.3 Alcance	2
1.4 Propósito	2
2Entregables.....	2
2.1 Documentación a entregar	2
3 características a ser probadas por sprint	3
3.1 Característica a ser probadas del sprint 1.....	3
3.2 Característica a ser probadas del sprint 2.....	4
3.3 Característica a ser probadas del sprint 3.....	4
3.4 Característica a ser probadas del sprint 4.....	5
4 Características a no ser probadas por sprint	5
4.1 Característica a no ser probado del sprint 1	5
4.2 Características a no ser probada del sprint 2.....	6
4.3Característica a no ser probadas del sprint 3.....	7
4.4 Característica a no ser probada del sprint 4	8
5. Criterios de aprobación y fallo.....	9
6. Criterios de suspensión y reanudación.....	10
7. Tareas de las pruebas.....	11
8. Necesidades ambientales.....	12
8.1 Hardware.....	12
8.2 Planeación de costo.....	12
8.3 SUT (sistema bajo pruebas)	13
9. Planificación de Scrum	14
9.1 Planificación de los Sprint	14
10. Informe de pruebas realizadas	15



10.1 casos de prueba del sprint 1	15
10.2 script correspondiente a cada caso de prueba del sprint 1	18
10.3 resultado de las pruebas ejecutadas del sprint 1.....	20
11. Casos de pruebas del sprint 2.....	21
11.1 Script correspondiente a cada caso de prueba del sprint 2.....	23
11.2 resultado de las pruebas ejecutadas del sprint 2.....	24
12. Caso de prueba del sprint 3	25
12.1 script correspondiente a cada caso de prueba del sprint 3	27
12.2 resultado de las pruebas ejecutadas del sprint 3.....	29
13. Caso de prueba del sprint 4.....	30
13.1 script correspondiente al caso de prueba del sprint 4	31
13.2 resultado de la prueba ejecutada del sprint 4	31

Índice de Ilustración

Ilustración 1: Script del caso de prueba 0.1 del sprint 1.....	18
Ilustración 2: Script del caso de prueba 0.2 del sprint 1.....	19
Ilustración 3: Script del caso de prueba 0.3 del sprint 1.....	20
Ilustración 4: Resultado de prueba 0.1, sprint 1.....	20
Ilustración 5: Resultado de prueba 0.2, sprint 1.....	20
Ilustración 6: Resultado del caso de prueba 0.3, sprint 1	20
Ilustración 7: Script del caso de prueba 0.4 del sprint 2.....	23
Ilustración 8: Script del caso de prueba 0.5 del sprint 2.....	24
Ilustración 9: Resultado del caso de prueba 0.4, sprint 2	24
Ilustración 10: Resultado del caso de prueba 0.5, sprint 2	24
Ilustración 11: Script del caso de prueba 0.6 del sprint 3.....	27
Ilustración 12: Script del caso de prueba 0.7 del sprint 3.....	28
Ilustración 13: Resultado del caso de prueba 0.6, sprint 3	29



Ilustración 14: Resultado del caso de prueba 0.7, sprint 3	29
Ilustración 15: Script del caso de prueba 0.8 del sprint 4.....	31
Ilustración 16: Resultado de prueba 0.8, del sprint 4	31

Índice de tablas

Tabla 1: Documentación a entregar.....	3
Tabla 2: Características probadas Sprint 1.	4
Tabla 3: Características probadas Sprint 2.	4
Tabla 4: Características probadas Sprint 3.	5
Tabla 5: Características probadas Sprint 4.	5
Tabla 6: Sprint 1 características a no ser probadas.	6
Tabla 7: sprint 2 características no probadas.....	7
Tabla 8: sprint 3 características no probadas.....	8
Tabla 9: sprint 4 características no probadas.....	8
Tabla 10: Criterios de aprobación y fallo.....	10
Tabla 11: Criterios de suspensión y reanudación.....	10
Tabla 12: Tareas de las pruebas.....	11
Tabla 13: Necesidades ambientales de hardware.	12
Tabla 14: Planeación de costos.	12
Tabla 15: Sistema bajo pruebas.	13
Tabla 16: planificación de Scrum.	14
Tabla 17: caso de prueba 0.1	15
Tabla 18: caso de prueba 0.2	16
Tabla 19: caso de prueba 0.3	17



PLAN DE PRUEBAS (Farm-veterinaria)



Tabla 20: caso de prueba 0.4	21
Tabla 21: caso de prueba 0.5.	22
Tabla 22: caso de prueba 0.6	25
Tabla 23: caso de prueba 0.7	26
Tabla 24: caso de prueba 0.8	30

1 introducción

El plan de prueba está elaborado con el fin de obtener un software funcional, elaborando pruebas unitarias a los componentes y funcionalidades del sistema, para eso se desarrolló pruebas con herramientas de Angular como, Karma y Jasmine, y también pruebas de caja blanca, esto permite conocer los fallos que puede estar presentando el software.

El plan de pruebas contiene 5 etapas las cuales son:

- Planeación de pruebas
- Diseño de pruebas
- Implementación de pruebas
- Evaluación del criterio de salidas
- Cierre del proceso.

1.1 Objetivo general

Desarrollar un plan de pruebas que verifique el funcionamiento de los componentes de la aplicación web para el manejo correcto de la gestión de compras, ventas, e inventario de la farmacia veterinaria COOPA.

1.2 Estrategia de pruebas

Para la verificación de código de la aplicación Farm-veterinaria se utiliza pruebas unitarias las cuales verificaran que los módulos y componentes de la aplicación funcionan correctamente, de igual forma se realizan test creados para cubrir requisitos del código y verificar los resultados.

Los test unitarios son una herramienta excelente para prevenir bugs en ejecución y para describir el comportamiento que debe tener el código, depende de los resultados, se valida y continúa, o se repara de forma que el error desaparezca.

Para la realización de pruebas unitarias se utilizan las herramientas de Karma y Jasmine, que facilita el framework de angular.

Las técnicas que se realizan son:

Caja blanca: se verifica la estructura interna del componente con independencia de las funcionalidades establecidas para el mismo, por lo tanto, o se comprueba las correcciones del resultado si se producen.

Se realizan historias de usuarios que sirve como guía, de acuerdo a las funcionalidades de la aplicación,

La prueba unitaria se da por finalizada cuando se realizan todas las verificaciones establecida y no se encuentre ningún defecto o bien se determine su suspensión.

1.3 Alcance

El plan de prueba, actúa como una guía para el desarrollo de software, ya que ordena y estructura de una mejor manera las actividades que se realizan en el proyecto Farm-veterinaria, el plan de pruebas permitirá al equipo a evaluar aspectos como: funcionalidad, usabilidad, encontrar errores y defectos que puedan existir en el software a fin de corregirlos.

Verificar que los validadores de datos funcionen y limiten el ingreso de información, para que no registre datos que no estén permitido

Se quiere comprobar además que la aplicación cumple con los requerimientos establecidos por el usuario.

1.4 Propósito

El propósito es organizar las características a ser probadas, para encontrar errores y defectos, para ello es necesario de un plan de pruebas para coordinarlas, a fin de asegurar la calidad del producto, durante el ciclo de vida del proyecto se escogió como ámbito las pruebas, todas las interfaces involucradas, teniendo como aspecto principal las historias de usuarios, para comprobar el rendimiento y la capacidad del software.

Con ello se verifica el cumplimiento de las especificaciones del diseño, los requisitos de análisis y a su vez se esperan encontrar los problemas percibidos del software, con la finalidad de entregar un software de calidad para el usuario

2Entregables

2.1 Documentación a entregar

DOCUMENTO	PERSONA QUIEN ENTREGA	PERSONA QUIEN RECIBE	FECHA PLANEADA	FECHA DE ENTREGA
Historias de usuario	Jackeline Davila Keylin López	Msc. Saira Urbina	12/11/2020	18/11/2020
Casos de pruebas del Sprint1	Jackeline Davila Keylin López	Msc. Saira Urbina	80/10/2020	04/11/2020
Casos de prueba del Sprint 2	Jackeline Davila Keylin López	Msc. Saira Urbina	05/11/2020	11/11/2020
Casos de prueba del Sprint 3	Jackeline Davila Keylin López	Msc. Saira Urbina	23/11/2020	04/12/2020
Casos de prueba del Sprint 4	Jackeline Davila Keylin López	Msc. Saira Urbina	07/12/2020	16/12/2020
Casos de pruebas	Jackeline Davila Keylin López	Msc. Saira Urbina	12/11/2020	18/11/2020
Plan de pruebas	Jackeline Davila Keylin López	Msc. Saira Urbina	12/11/2020	18/11/2020
Informes de pruebas realizadas	Jackeline Davila Keylin López	Msc. Saira Urbina	22/12/2020	22/12/2020

Tabla 1: Documentación a entregar.

3 características a ser probadas por sprint

3.1 Característica a ser probadas del sprint 1

CARACTERISTICA	DESCRIPCIÓN	MODULO	Código de prueba
Validar el formulario del uso del carrito de compra	Verificar que el componente principal de carrito de compra se haya realizado correctamente con sus módulos	Componente #1 Sprint 1	0.1
Funcionalidad para agregar al carrito de compra	El usuario dará clic en el botón agregar un producto al carrito de compra y este sumará la cantidad correspondiente a su compra.	Componente #1 Sprint1	0.2
Funcionalidad en el botón guardar detalle de pedido	El usuario comprobará el botón guardar realiza los cambios en el carrito de compra	Componente #1 Sprint 1	0.3

Tabla 2: Características probadas Sprint 1.

3.2 Característica a ser probadas del sprint 2

CARACTERISTICA	DESCRIPCIÓN	MODULO	Código de prueba
Formulario Detalles de pedido	El usuario registra los productos necesarios y verifica es el producto está en inventario	Componente #2 Sprint 2	0.4
Eliminar productos en inventario	El usuario podrá eliminar el producto deseado	Componente #2 Sprint 2	0.5

Tabla 3: Características probadas Sprint 2.

3.3 Característica a ser probadas del sprint 3

CARACTERÍSTICA	DESCRIPCIÓN	MODULO	Código de prueba
Restar existencia del producto	El usuario administrador dentro de la aplicación podrá ver la resta de sus productos en inventario	Componente #3 Sprint #3	0.6
Buscar producto	Buscar por filtro los productos en inventario	Componente #3 Sprint #3	0.7

Tabla 4: Características probadas Sprint 3.

3.4 Característica a ser probadas del sprint 4

CARACTERÍSTICA	DESCRIPCIÓN	MODULO	Código de prueba
Registro de usuario	Después de haberse registrado el usuario podrá iniciar sesión	#5 sprint 4	0.8

Tabla 5: Características probadas Sprint 4.

4 Características a no ser probadas por sprint

4.1 Característica a no ser probado del sprint 1

CARACTERÍSTICA	DESCRIPCIÓN	JUSTIFICACIÓN	RIESGO
Ver tienda de productos	Inicia la aplicación y el usuario registrado y no registrado tiene acceso al catálogo de producto.	No se realiza prueba porque el framework de angular permite la reutilización de código, de manera que utilizamos los mismos componentes para la	Impedir acceso a la app

		estructura del catálogo de producto.	
Ver carrito de compra	El usuario puede ver la cantidad de los productos que ha comprado.	No se realizó prueba en ver carrito de compra, porque, el usuario de la aplicación ve el carrito de compra a la hora de realizar su pedido, esta funcionalidad ya ha sido probada.	Que alguno de los componentes sufra cambios en la aplicación
Ver mis pedidos	Dar clic en botón carrito de compra, en el cual visualiza los pedidos realizados.	No se realizó prueba al botón ver mi pedido porque esta funcionalidad depende del componente carrito de compra	Que los productos no carguen
Pago en línea	Después que realiza pedido, da clic en modo de pago	No se efectúa una prueba porque PayPal es quien nos da el enlace a la aplicación.	Desconocimiento de pago en línea

Tabla 6: Sprint 1 características a no ser probadas.

4.2 Características a no ser probada del sprint 2

CARACTERÍSTICA	DESCRIPCIÓN	JUSTIFICACIÓN	RIESGO
Ver información de la farmacia	Inicia sesión como usuario cliente, y da click en el menú, luego en quiénes somos.	No se desarrolló una prueba porque no era prioritario hacerla ya que se realizaron pruebas con valores.	Impedir acceso a la app

Restar productos	El usuario administrador, puede ver que los productos en existencia sean restados si se ha realizado una venta	Se ha desarrollado, una prueba para la funcionalidad restar existencia de producto lo que nos permite ver que el producto sea restado, entonces realizar una prueba, de ver producto, no es necesario hacerla.	Que la función restar existencia de producto, no esté funcionando correctamente
Editar producto	Inicia la aplicación como usuario administrador, luego click en productos, se mostrará la lista de productos y luego se encuentra el botón editar	No se realizó prueba para esta funcionalidad, ya que editar producto depende del componente producto, y se estaría redundando en el código ya que cada prueba es realizada independiente.	Que los productos no carguen

Tabla 7: sprint 2 características no probadas

4.3 Característica a no ser probadas del sprint 3

CARACTERÍSTICA	DESCRIPCIÓN	JUSTIFICACIÓN	RIESGO
Reporte de ventas diarias	Inicia sesión como usuario administrador, y da Click en el menú, luego en el botón ventas.	No se desarrolló una prueba de reporte de ventas, porque a la hora de elaborar los test de prueba. Marcaban errores, ya que dependía de varios componentes, que ya había sido probado.	Impedir acceso a la app
Administrar estado de pedido	El usuario administrador, puede administrar	Administrar estado de pedido es una funcionalidad de ver pedido, y no se realizó	Que la función restar existencia de producto, no

	el estado de pedido	pruebas por la complejidad de su dependencia	esté funcionando correctamente
Ver existencia de producto	Inicia la aplicación como usuario administrador, luego click en productos, se mostrará la lista de productos que existe en inventario	No se realizó prueba para esta funcionalidad, ya que existe una prueba restar existencia de producto.	Que no existen productos en inventario.

Tabla 8: sprint 3 características no probadas

4.4 Característica a no ser probada del sprint 4

CARACTERÍSTICA	DESCRIPCIÓN	JUSTIFICACIÓN	RIESGO
Ver cliente	Inicia sesión como usuario administrador, y da Click en el menú, luego en el botón ventas y luego en detalles, cargará una lista de los clientes	No se desarrolló una prueba para esta función ya que no se considera tan necesaria a la hora de escribir una prueba en angular, de modo que solo se observa que exista cliente es una funcionalidad que depende de un usuario registrado	Que no halla registros de clientes o usuarios en la aplicación
Chat individual	El usuario cliente y administrador tendrá la opción de chat individual en la pantalla inicial de la aplicación	No se desarrolló una prueba para este componente porque, solo se utilizó el código que nos facilitó Facebook, para enlazar el chat a la aplicación	Que la página no esté disponible en línea

Tabla 9: sprint 4 características no probadas

5. Criterios de aprobación y fallo

ID CRITERIO	DESCRIPCIÓN	APROBACIÓN	FALLO
0.1	Se evaluará la característica del componente del carrito de compra se espera que el módulo carrito de compra funcione, y que el usuario pueda realizar las tareas dentro del formulario	Una vez obtenido el resultado del módulo, se determina que el módulo carrito de compras está correcto.	Fallo que el usuario introduzca datos no válidos en el módulo
0.2	Se evalúa agregar productos en el módulo carrito de compra con un contador.	Una vez obtenido el valor final del contador, si este alcanza una cantidad mínima de 1 y el máximo menor a la existencia de productos	retorno a un valor que no esté en la existencia
0.3	Se evaluará el botón guardar producto	Una vez obtenida esta función se verifica que el botón funciona	Permanencia de los productos eliminados
0.4	Se evalúa la funcionalidad detalles de producto	Una vez obtenido la lista de productos se evalúan los detalles de productos	Que al darle Click al botón no funcione como se espera
0.5	Eliminar producto de inventario	Una vez realizada la función eliminará productos en inventario se verifica que si existe	Que no sea eliminado el producto de la base de datos
0.6	Restar existencia de producto	Una vez vendido un producto el sistema restará la existencia en inventario	Que siga permaneciendo en inventario

0.7	Buscar producto en inventario	El usuario podrá buscar su producto mediante filtros	Que el producto que se busca no exista en inventario
0.8	Registro de usuario	Podrá iniciar sesión en la aplicación después de registrarse	Que los datos del usuario no se hayan registrado correctamente

Tabla 10: Criterios de aprobación y fallo.

6. Criterios de suspensión y reanudación

CRITERIO DE SUSPENSIÓN	CRITERIO DE REANUDACIÓN
Hubo pruebas de fallo aproximadamente el 20%	Se realizó revisión de depuración mediante la revisión de código
Fallos que existieron en las pruebas fueron por inconvenientes con integración de librerías en Angular click, y conexión a Node.js por ser un componente de angular click	Aislar funciones para corregir inconvenientes durante es test de funcionalidades

Tabla 11: Criterios de suspensión y reanudación.

7. Tareas de las pruebas

TAREA	DESCRIPCIÓN	FECHA INICIO	FECHA FIN	DURACIÓN (HRS)	RESPONSABLE	ROL
Prueba de componentes	Probar la carga correcta de los componentes de la aplicación	12/11/2020	14/11/2020	12 hrs	Br. Jackeline Dávila	Desarrollador
Prueba de funciones	Probar las funciones principales de los componentes de la aplicación	16/11/2020	17/11/2020	24 hrs	Br. Jackeline Dávila	Desarrollador
Corregir errores encontrados	Resolver errores de las pruebas de funciones	17/11/2020	18/11/2020	24 hrs	Br. Keylin López	Diseñador
Configuración de coverage	Configuración de herramienta coverage para visualizar los resultados de forma gráfica.	18/11/2020	19/11/2020	24 hrs	Br. Keylin López	Diseñador

Tabla 12: Tareas de las pruebas.

8. Necesidades ambientales

8.1 Hardware

DISPOSITIVO	MARCA	CARACTERISTICAS	¿TENEMOS EL EQUIPO?
Pc	HP	<ul style="list-style-type: none"> Procesador: AMD-E16015 Memoria instalada(RAN): 4,00 GB Tipo de sistema: sistema operativo de 64bits procesador x 64. 	si

Tabla 13: Necesidades ambientales de hardware.

8.2 Planeación de costo

ID RECURSO	TIPO DE RECURSO	UNIDAD	COSTO DE RECURSO X UNIDAD	HORAS	CANTIDAD	TOTAL
1	Desarrollador	2	USD 40	8	16	USD 80
2	pc	2	USD 20	15	30	USD 40
3	internet	1	USD 10	15	15	USD 10
4	Materiales					
	Lápiz	2	USD 2	6	4	USD 4
	cuadernos	2	USD 2	6	4	USD 4
Total						USD 138

Tabla 14: Planeación de costos.

8.3 SUT (sistema bajo pruebas)

MODULO	¿DEPENDENCIA ENTRE MODULOS?	REQUERIMIENTOS
Carrito de compra	no	El módulo carrito de compra y sus características están diseñado correctamente
Registro de producto	no	Carga de los datos generados por el usuario
Existencia de producto en inventario	si	Calcular la cantidad de productos en el carrito de compra

Tabla 15: Sistema bajo pruebas.

9. Planificación de Scrum

9.1 Planificación de los Sprint

Planificación inicial de las iteraciones			
id	Historias de usuarios	Sprint programado	Fecha de entrega
2	Ver tienda de producto	1	04/11/2020
5	Agregar carrito de compra	1	
7	Actualizar carrito de compra	1	
6	Ver carrito de compra	1	
8	Eliminar producto del carrito de compra	1	
10	Pago en línea	1	
11	Detalle de pedido	1	
9	Realizar pedido	1	
1	Ver información de la farmacia	2	20/11/2020
13	Contáctenos	2	
14	Registrar producto	2	
15	Editar producto	2	
16	Eliminar producto	2	
20	Ver pedido	2	
19	Reporte de venta diarias	3	04/12/2020
21	Administrar estado de pedido	3	
23	Existencia de producto	3	
24	Restar existencia de producto	3	
18	Ver lista de producto en inventario	3	
17	Buscar producto en inventario	3	
22	Ver cliente	4	16/12/2020
12	Chat individual	4	
3	Registrar cliente	4	
4	Iniciar sesión	4	
25	Facturación	4	

Tabla 16: planificación de Scrum.

10. Informe de pruebas realizadas

10.1 casos de prueba del sprint 1

Test Case ID	0.1	Test Case Description	Validar el componente carrito de compra (funcionalidad para validar los datos de la compra)		
Type of test	Caja blanca		Cobertura de decisión		
Created By	Keylin López	Reviewed By		Version	1.2

QA Tester's Log

Tester's Name	Jackeline Dávila	Date Tested	29/10/2020	Test Case (Pass/Fail/Not Executed)	pass
----------------------	------------------	--------------------	------------	---	------

S #	Prerequisites:
1	Programar función que verifique el valor de los datos de entrada del formulario.
2	Escribir pruebas para verificar los datos de corrección introducidos

S #	Test Data
1	CarritoService = { }
2	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Ejecutar prueba	Inicia el tiempo de ejecución	Test de prueba ejecutado	pass
2	El test permite crear variable al ejecutar la sentencia, y guarda el valor de error según cada estado	Se crear variable por cada estado con el valor de falso, porque ningún estado indica un error	Las variables que inicia en error se han creado	pass
3	El test permite comprobar que la variable total contiene solo número	Valida la cadena de caracteres	Devuelve verdadero si el valor es correcto o falso si el caso es contrario	pass
4	Inicia sesión como usuario cliente	Muestra toda la interfaz del carrito de compra	Muestra la funcionalidad correcta del carrito de compra	pass

Tabla 17: caso de prueba 0.1

Test Case ID	0.2	Test Case Description	Componente Agregar al carrito de compra (funcionalidad para validar los datos de agregar producto)		
Type of test	Caja blanca		Cobertura de decisión		
Created By	Keylin López	Reviewed By		Version	1.2

QA Tester's Log

Tester's Name	Jackeline Dávila	Date Tested	29/10/2020	Test Case (Pass/Fail/Not Executed)	pass
----------------------	------------------	--------------------	------------	---	------

S #	Prerequisites:
1	Programar función que verifique el valor de los datos de entrada agregar producto.
2	Escribir pruebas para verificar los datos de son correctos

S #	Test Data
1	Appcomponet = []; CarritoService =[];
2	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Ejecutar prueba	Inicia el tiempo de ejecución	Test de prueba ejecutado	pass
2	El test permite recorrer el arreglo de componente y el de producto	Hay iteración de arreglo de manera que el arreglo del componente se itera dentro del arreglo de producto	Iteración realizada	pass
3	Inicia sesión como usuario cliente	Muestra la interfaz de la farmacia veterinaria	Se muestra el componente e interfaz de la farmacia veterinaria correctamente	pass
4	Agregara al carrito de compra	Muestra la cantidad de los productos agregado al carrito de compra	Muestra el resultado de compra	pass

Tabla 18: caso de prueba 0.2

Test Case ID	0.3	Test Case Description	Guardar producto (funcionalidad para validar los datos de guardar producto)		
Type of test	Caja blanca		Cobertura de decisión		
Created By	Keylin López	Reviewed By		Version	1.2

QA Tester's Log

Tester's Name	Jackeline Dávila	Date Tested	29/10/2020	Test Case (Pass/Fail/Not Executed)	pass
----------------------	------------------	--------------------	------------	---	------

S #	Prerequisites:
1	El formulario de producto debe estar creado para adicionar nuevas estructuras
2	Escribir pruebas para verificar que los datos son correctos

S #	Test Data
1	NombreProd: “”; Descripción: “”; Cantidad: “”; Precio;
2	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Ejecutar prueba	Inicia el tiempo de ejecución	Test de prueba ejecutado	pass
2	Introducir usuario y contraseña como administrador	Se muestra la aplicación con el panel de administrador	Ejecución finalizada	pass
3	Click en el botón productos	Se despliega el formulario de registro de productos	Los campos del formulario se han llenado correctamente	pass
4	Click en guardar producto	El test recorre el arreglo producto, con los datos necesarios	Producto guardado	pass

Tabla 19: caso de prueba 0.3

10.2 script correspondiente a cada caso de prueba del sprint 1

Para realizar los test de prueba se utilizó la estructura que nos facilita el framework de angular, con la tecnología de Karma y Jasmine, en esta prueba se observa los valores asignados a las pruebas, las importaciones que necesita de otro componente para ser probada, luego que se escribir la prueba unitaria correspondiente a cada caso de prueba, se ejecuta el comando “ng test” la cual muestra que la prueba esta correcta.

```
import { TestBed } from '@angular/core/testing';

import { CarritoService } from './carrito.service';

describe('CarritoService', () => {
  let service: CarritoService;
  total: int;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(CarritoService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});
```

Ilustración 1: Script del caso de prueba 0.1 del sprint 1

```
//prueba de componente
describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule
      ],
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  });

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'angular-veterinaria'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('angular-veterinaria');
  });
});
```

```
it('should render title', () => {
  const fixture = TestBed.createComponent(AppComponent);
  fixture.detectChanges();
  const compiled = fixture.nativeElement;
  expect(compiled.querySelector('.content span').textContent).toContain('angular-veterinaria app is running!');
});
});
```

Ilustración 2: Script del caso de prueba 0.2 del sprint 1


```
//boton guardar producto
describe('AppComponent' , ()=>{
  it('Guardar-productto', ()=>{
    expect(true).toBe(true);
  });
});
```

Ilustración 3: Script del caso de prueba 0.3 del sprint 1

10.3 resultado de las pruebas ejecutadas del sprint 1

Cobertura del proyecto: al ejecutar el código “ng tests--codeCoverage” se creará en la carpeta src, otra llamada coverage, lo que nos permite abrir index.html, que muestra gráficamente como tenemos cada archivo Spec.ts en cuanto a cobertura de código

Archivo	Declaraciones	Ramas	Funciones	Líneas
Fuente	100%	3/3	100%	0/0
src/app	100%	3/3	100%	0/0

Ilustración 4: Resultado de prueba 0.1, sprint 1

File	Statements	Branches	Functions	Lines
polyfills.ts	0%	0/0	0%	0/0
test.ts	100%	3/3	100%	0/0

Ilustración 5: Resultado de prueba 0.2, sprint 1

File	Statements	Branches	Functions	Lines
polyfills.ts	0%	0/0	0%	0/0
test.ts	100%	3/3	100%	0/0

Ilustración 6: Resultado del caso de prueba 0.3, sprint 1

11. Casos de pruebas del sprint 2

Test Case ID	0.4	Test Case Description	Detalles producto (funcionalidad para validar el detalle de producto)		
Type of test	Caja blanca		Cobertura de decisión		
Created By	Keylin López	Reviewed By		Version	1.2

QA Tester's Log

Tester's Name	Jackeline Dávila	Date Tested	11/11/2020	Test Case (Pass/Fail/Not Executed)	pass
----------------------	------------------	--------------------	------------	---	------

S #	Prerequisites:	S #	Test Data
1	El formulario de producto debe estar creado para adicionar nuevas estructuras	1	Verdetalle= [];
2	Escribir pruebas para verificar los datos de son correctos	2	
3	Iniciar como usuario administrador		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Ejecutar prueba	Inicia el tiempo de ejecución	Test de prueba ejecutado	pass
2	Dentro del formulario producto. Click en el botón detalles	Se muestral el detalle de los productos con los datos necesarios	Ejecución finalizada	pass
3	El test recorre el arreglo Detalleproducto	Se realiza la iteración y verifica si los datos son correctos	Iteración finalizada	pass

Tabla 20: caso de prueba 0.4

Test Case ID	0.5	Test Case Description	Eliminar producto de inventario (funcionalidad para validar el detalle de producto)		
Type of test	Valor de equivalencia		Cobertura de decisión		
Created By	Keylin López	Reviewed By		Version	1.2

QA Tester's Log

Tester's Name	Jackeline Dávila	Date Tested	29/11/2020	Test Case (Pass/Fail/Not Executed)	pass
----------------------	------------------	--------------------	------------	---	------

S #	Prerequisites:
1	El formulario de producto debe estar creado para adicionar nuevas estructuras
2	Escribir pruebas para verificar los datos de son correctos
3	Iniciar como usuario administrador

S #	Test Data
1	1,1
2	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Ejecutar prueba	Inicia el tiempo de ejecución	Test de prueba ejecutado	pass
2	El test permite recorrer el arreglo de productos	El test verifica si hay un producto agregado, si lo hay, devuelve un valor verdadero de lo contrario dará error	Se comprueba que si funciona correctamente	
3	Dentro del formulario producto. Click en el botón eliminar producto	Se muestrial el detalle de los productos con los datos necesarios	Producto eliminado de inventario	pass

Tabla 21: caso de prueba 0.5.

11.1 Script correspondiente a cada caso de prueba del sprint 2

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
import { DetalleDeProductoComponent } from './detalle-de-producto.component';

describe('DetalleDeProductoComponent', () => {
  let component: DetalleDeProductoComponent;
  let fixture: ComponentFixture<DetalleDeProductoComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ DetalleDeProductoComponent ]
    })
    .compileComponents();
  }));

  beforeEach(() => {
    fixture = TestBed.createComponent(DetalleDeProductoComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Ilustración 7: Script del caso de prueba 0.4 del sprint 2

```
describe('AppComponent' , ()=>{
  let app: AppComponent;
  beforeEach(async(()=> {
    //app = new AppComponent();
  }));

  it('la suma de los productos debe dar 2', async(()=>{
    expect (app.add(1,1)).toEqual(2)
  }));
});
```

Ilustración 8: Script del caso de prueba 0.5 del sprint 2

11.2 resultado de las pruebas ejecutadas del sprint 2

Podemos observar la cobertura del proyecto realizando los testing de pruebas del sprint 2, del caso de prueba 0.4 y 0.5

Archivo	Declaraciones	Ramas	Funciones	Lineas
Fuente	100%	3/3	100%	0/0
src/app	100%	3/3	100%	0/0

Ilustración 9: Resultado del caso de prueba 0.4, sprint 2

File	Statements	Branches	Functions	Lines
src	100%	3/3	100%	0/0
src/clases	50%	2/4	33.33%	1/3

Ilustración 10: Resultado del caso de prueba 0.5, sprint 2

La barra amarilla muestra la cobertura de código del archivo spec, podemos observar que la franja verde nos da la cobertura de código completa.

12. Caso de prueba del sprint 3

Test Case ID	0.6	Test Case Description	Restar existencia de producto (funcionalidad para validar la existencia de producto)		
Type of test	Caja blanca		Cobertura de decisión		
Created By		Reviewed By		Version	1.2

QA Tester's Log

Tester's Name	Jackeline Dàvila	Date Tested	30/11/2020	Test Case (Pass/Fail/Not Executed)	pass
----------------------	------------------	--------------------	------------	---	------

S #	Prerequisites:
1	El formulario de producto debe estar creado para adicionar nuevas estructuras
2	Escribir pruebas para verificar los datos de son correctos
3	Iniciar como usuario administrador
4	Tiene que realizarse ventas en la aplicación

S #	Test Data
1	Productocomponent =[]; ventascomponent =[]
2	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Ejecutar prueba	Inicia el tiempo de ejecución	Test de prueba ejecutado	pass
2	El test permite recorrer el arreglo de productos y ventas	El test verifica si hay un producto eliminado, si lo hay, devuelve un valor verdadero de lo contrario dará error	Se comprueba que si funciona correctamente	
3	Dentro del formulario producto. Click en el botón producto	Se muestral el detalle de los productos en existencia.	Se resta la existencia del producto	pass

Tabla 22: caso de prueba 0.6

Test Case ID	0.7	Test Case Description	Buscar producto en inventario (funcionalidad para validar la los datos de producto)		
Type of test	Caja blanca		Cobertura de decisión		
Created By		Reviewed By		Version	1.2

QA Tester's Log

Tester's Name	Jackeline Dàvila	Date Tested	20/12/2020	Test Case (Pass/Fail/Not Executed)	pass
----------------------	------------------	--------------------	------------	---	------

S #	Prerequisites:
1	El formulario de producto debe estar creado para adicionar nuevas estructuras
2	Escribir pruebas para verificar los datos de son correctos
3	Iniciar como usuario administrador

S #	Test Data
1	Targetaproductocomponent[];
2	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Ejecutar prueba	Inicia el tiempo de ejecución	Test de prueba ejecutado	pass
2	El test permite recorrer los datos del arreglo,	El test verifica si los elementos dentro del arreglo y captura los datos de la base de dato	Se muestreal los datos correspondientes a la iteración del arreglo	
3	Dentro del formulario producto. Click en el botón producto	Se observa la lista de los productos en inventario.	El test de prueba verifica los datos en la lista	pass
4	Click en buscar producto	El test permite buscar por filtro la lista de producto	Se realiza la búsqueda mediante nombre y fecha de producto	pass

Tabla 23: caso de prueba 0.7

12.1 script correspondiente a cada caso de prueba del sprint 3

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { VentasComponent } from './ventas.component';

describe('VentasComponent', () => {
  let component: VentasComponent;
  let fixture: ComponentFixture<VentasComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ VentasComponent ]
    })
    .compileComponents();
  }));

  beforeEach(() => {
    fixture = TestBed.createComponent(VentasComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Ilustración 11: Script del caso de prueba 0.6 del sprint 3


```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { TarjetaProductoComponent } from './tarjeta-producto.component';

describe('TarjetaProductoComponent', () => {
  let component: TarjetaProductoComponent;
  let fixture: ComponentFixture<TarjetaProductoComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ TarjetaProductoComponent ]
    })
    .compileComponents();
  }));

  beforeEach(() => {
    fixture = TestBed.createComponent(TarjetaProductoComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Ilustración 12: Script del caso de prueba 0.7 del sprint 3

12.2 resultado de las pruebas ejecutadas del sprint 3

File ▲	Statements ⇅	Branches ⇅	Functions ⇅	Lines ⇅
src		100% 3/3	100% 0/0	100% 0/0 100% 3/3
src/clases		75% 3/4	100% 0/0	66.67% 2/3 75% 3/4

Ilustración 13: Resultado del caso de prueba 0.6, sprint 3



File ▲	Statements ⇅	Branches ⇅	Functions ⇅	Lines ⇅
src		100% 3/3	100% 0/0	100% 0/0 100% 3/3
src/clases		50% 2/4	100% 0/0	33.33% 1/3 50% 2/4

Ilustración 14: Resultado del caso de prueba 0.7, sprint 3

13. Caso de prueba del sprint 4

Test Case ID	0.8	Test Case Description	Validar el formulario registro de usuario cliente (funcionalidad para validar la los datos inicio de usuario)		
Type of test	Caja blanca		Cobertura de decisión		
Created By	Keylin López	Reviewed By		Version	1.2

QA Tester's Log

Tester's Name	Jackeline Dávila	Date Tested	21/11/2020	Test Case (Pass/Fail/Not Executed)	pass
----------------------	------------------	--------------------	------------	---	------

S #	Prerequisites:
1	Programar las funciones que verifiquen el valor de los datos del formulario
2	Escribir pruebas para verificar los datos que han sido introducido
3	

S #	Test Data
1	LoginComponent = { };
2	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Ejecutar prueba	Inicia el tiempo de ejecución	Test de prueba ejecutado	pass
2	El test permite recorrer los datos del arreglo del componente login,	El test verifica si los elementos dentro del arreglo	Se muestran los datos correspondientes a la iteración del arreglo	
3	Iniciar en la aplicación	Se espera que muestre el formulario del registro de cliente	El formulario está correcto, para registrar datos	pass
4	Click en registro de usuario	Llenar los datos correspondientes en el formulario,	Se registró un nuevo usuario en la aplicación	pass

Tabla 24: caso de prueba 0.8

13.1 script correspondiente al caso de prueba del sprint 4

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { LoginComponent } from './login.component';

describe('LoginComponent', () => {
  let component: LoginComponent;
  let fixture: ComponentFixture<LoginComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ LoginComponent ]
    })
    .compileComponents();
  }));

  beforeEach(() => {
    fixture = TestBed.createComponent(LoginComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Ilustración 15: Script del caso de prueba 0.8 del sprint 4

13.2 resultado de la prueba ejecutada del sprint 4

Expediente	Declaraciones	Ramas	Funciones	Lineas
src	100%	3/3	100%	0/0
src / app	84.62%	11/13	45.45%	10/22
src / app / form	94.74%	18/19	55.17%	16/29

Ilustración 16: Resultado de prueba 0.8, del sprint 4