

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

CARRERA SISTEMAS DE INFORMACIÓN



**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN**

**DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN
WEB PARA LA GESTIÓN DE TIENDAS DE MASCOTAS. CASO
DE ESTUDIO: ZEUS PELUQUERÍA CANINA**

AUTOR:

ALVEAR MENA ANDRÉS DAVID

QUITO DM, 25 DE MAYO DE 2023

DEDICATORIA

Este trabajo se lo dedico a mi familia, cada integrante ha sido un impulso importante en algún momento de la carrera, por todo su apoyo durante el proceso de formación académica y siempre brindarme el soporte necesario para lograr alcanzar una meta más en mi vida.

AGRADECIMIENTO

Primero, agradezco a Dios por darme la fuerza para seguir adelante y poder cumplir con una meta más en mi vida.

Agradezco a mis padres por haberme brindado la oportunidad de conseguir otra meta más en vida, con el esfuerzo que ha significado para ellos permitirme llegar a este punto en mi vida.

A mis compañeros de la carrera, gracias a ellos pude seguir adelante en momentos difíciles durante los semestres que compartí con ellos y su apoyo en cada momento que se los necesitó.

A la Pontificia Universidad Católica del Ecuador y a la Facultad de Ingeniería, que me han permitido culminar mis estudios y lograr alcanzar una meta muy importante en mi vida, a todos los profesores que se han esforzado para brindarme un poco de conocimiento, especialmente a Fabián De la Cruz por el contratiempo que significó este trabajo y que gracias a él ha sido posible la elaboración del presente proyecto.

ÍNDICE DE CONTENIDO

1	CAPÍTULO I: INTRODUCCIÓN	1
1.1	Justificación.....	1
1.2	Planteamiento del problema	1
1.3	Objetivos	2
1.3.1	Objetivo general	2
1.3.2	Objetivos específicos.....	2
1.4	Alcance.....	2
1.5	Contenido	2
2	CAPÍTULO II: FUNDAMENTOS TEÓRICOS.....	4
2.1	Situación Actual	4
2.1.1	Antecedentes o marco referencial	4
2.1.2	Marco teórico	4
2.1.3	Marco conceptual	4
2.2	Metodología de desarrollo.....	5
2.2.1	Roles.....	6
2.2.2	Herramientas de Scrum	6
2.3	Herramientas de desarrollo.....	7
3	CAPÍTULO III: CASO DE ESTUDIO	10
3.1	Análisis y levantamiento de requerimientos	10
3.2	Diseño de la base de datos y arquitectura del sistema.....	13
3.2.1	Base de Datos	13
3.2.2	Arquitectura.....	14
3.3	Primer ciclo de desarrollo	15
3.3.1	Diseño de interfaces	15
3.3.2	Configuración del ambiente de desarrollo.....	16
3.3.3	Creación de base de datos en Firebase	17
3.3.4	Sistema de autenticación	18
4	CAPÍTULO IV: IMPLEMENTACIÓN	19
4.1	Segundo ciclo de desarrollo	19
4.1.1	CRUD de clientes.....	19
4.1.2	CRUD de mascota.....	20
4.1.3	CRUD de servicios.....	21
4.1.4	CRUD de productos	22
4.1.5	CRUD de citas.....	23

4.1.6	CRUD de facturación	24
4.2	Tercer ciclo de desarrollo	26
4.3	Pruebas del sistema	30
4.3.1	Pruebas Unitarias.....	30
4.3.2	Pruebas de aceptación	33
5	CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	35
5.1	Conclusiones	35
5.2	Recomendaciones.....	36
6	REFERENCIAS BIBLIOGRÁFICAS	37
7	ANEXOS.....	39
7.1	Historias de Usuario	39
7.2	Product Backlog	46
7.3	Sprint Backlog.....	46
7.4	Pruebas de Aceptación	49

ÍNDICE DE TABLAS

Tabla 1: Asignación de roles	6
Tabla 2: Ejemplo de Historia de Usuario	7
Tabla 3: Herramientas de desarrollo	8
Tabla 4: Requerimientos Funcionales	11
Tabla 5: Planificación de sprints	12
Tabla 6: Historia de Usuario de Registro de Cliente.....	13
Tabla 7: Versiones de Herramientas	17
Tabla 8: Prueba de aceptación relacionada con la HU01	33
Tabla 9: HU01 Iniciar y Cerrar sesión	39
Tabla 10: HU02 Crear nuevo cliente.....	39
Tabla 11: HU03 Editar cliente	39
Tabla 12: HU04 Visualizar listado de clientes	40
Tabla 13: HU05 Crear nueva mascota	40
Tabla 14: HU06 Editar mascota.....	41
Tabla 15: HU07 Visualizar listado de mascotas	41
Tabla 16: HU08 Crear nuevo servicio.....	41
Tabla 17: HU09 Editar servicio	42
Tabla 18: HU10 Visualizar listado de servicios.....	42
Tabla 19: HU11 Crear nuevo producto	42
Tabla 20: HU12 Editar producto.....	43
Tabla 21: HU13 Visualizar listado de productos	43
Tabla 22: HU14 Crear nueva cita.....	43
Tabla 23: HU15 Editar cita	44
Tabla 24: HU16 Visualizar listado de citas.....	44
Tabla 25: HU17 Crear nueva factura	45
Tabla 26: HU18 Visualizar factura	45
Tabla 27: HU19 Recuperar contraseña	45
Tabla 28: Product Backlog.....	46
Tabla 29: Sprint Backlog.....	46
Tabla 30: PA01 Iniciar y cerrar sesión.....	49
Tabla 31: PA02 Crear nuevo cliente	49
Tabla 32: PA03 Editar cliente	50
Tabla 33: PA04 Visualizar listado de clientes.....	50
Tabla 34: PA05 Crear nueva mascota	50
Tabla 35: PA06 Editar mascota.....	51
Tabla 36: PA07 Visualizar listado de mascotas	51
Tabla 37: PA08 Crear nuevo servicio	52
Tabla 38: PA09 Editar servicio	52
Tabla 39: PA10 Visualizar listado de servicios.....	52
Tabla 40: PA11 Crear nuevo producto.....	53
Tabla 41: PA12 Editar producto.....	53
Tabla 42: PA13 Visualizar listado de productos	54
Tabla 43: PA14 Crear nueva cita	54
Tabla 44: PA15 Editar cita	54
Tabla 45: PA16 Visualizar listado de citas	55
Tabla 46: PA17 Crear nueva factura	55
Tabla 47: PA18 Visualizar factura	55
Tabla 48: PA19 Recuperar contraseña	56

ÍNDICE DE FIGURAS

Figura 1: Ficha Tienda	10
Figura 2: Base de datos	14
Figura 3: Arquitectura	15
Figura 4: Prototipo de inicio de sesión.....	16
Figura 5: Prototipo de módulo citas	16
Figura 6: Creación de base de datos en Firebase.....	17
Figura 7: Autenticación de Firebase.....	18
Figura 8: Pantalla de iniciar sesión	18
Figura 9: Formulario de creación de clientes	19
Figura 10: Vista de módulo clientes.....	20
Figura 11: Vista editar cliente	20
Figura 12: Pantalla de Mascotas.....	21
Figura 13: Vista módulo mascotas	21
Figura 14: Pantalla de Servicios.....	22
Figura 15: Vista módulo de servicios.....	22
Figura 16: Pantalla de Productos.....	23
Figura 17: Formulario Cita Parte 1	24
Figura 18: Formulario Cita Parte 2	24
Figura 19: Vista módulo de citas.....	24
Figura 20: Pantalla de citas de la fecha actual.....	25
Figura 21: Pantalla para realizar el pago	26
Figura 22: Cita que ha sido cancelada.....	26
Figura 23: Pantalla de inicio.....	27
Figura 24: Formulario de recuperación de contraseña	27
Figura 25: Mensaje de cambio de contraseña en el correo.....	28
Figura 26: Cambio de contraseña.....	28
Figura 27: Verificación del cambio de contraseña realizado	28
Figura 28: Interfaz de Cypress	31
Figura 29: Código para realizar la prueba de renderizado de la página de inicio de sesión.....	31
Figura 30: Prueba de renderización finalizada	32
Figura 31: Prueba de renderización e inicio de sesión en la página inicial	32
Figura 32: Prueba de ingreso de un nuevo cliente y almacenamiento en la base de datos.....	33

RESUMEN

Este informe técnico presenta el desarrollo de una aplicación web para la gestión de la tienda de mascotas “Zeus peluquería canina.”

El proyecto se ha desarrollado con el fin de automatizar los procesos de gestión de clientes, mascotas, productos, servicios, citas y facturación dentro del negocio. La parte interna de la aplicación web se ha desarrollado con el *framework React JS* y el uso de *Firebase* como gestor de base de datos. El proyecto utiliza la metodología *Scrum*, debido al uso de varios ciclos de desarrollo y poder llevar un control de los avances para lograr terminar el software en el tiempo estimado.

El documento está estructurado de la siguiente manera: la primera sección corresponde a la introducción dando a conocer el contexto del problema, objetivos y alcance del proyecto. En la segunda sección se describe la metodología de desarrollo y las herramientas que se utilizarán para la implementación del proyecto. En la tercera sección se detalla el análisis y requerimientos de la aplicación web, el diseño de la base de datos, la arquitectura del sistema y el primer ciclo de desarrollo. En la cuarta sección se muestra los resultados obtenidos en el resto de los ciclos de desarrollo y las diferentes pruebas realizadas. Finalmente, en la última sección se detallan las conclusiones y recomendaciones obtenidas al finalizar el desarrollo del proyecto.

PALABRAS CLAVE: *Framework, React, Firebase, JavaScript, Scrum, Aplicación web y tienda de mascotas.*

ABSTRACT

This technical report presents the development of a web application for the management of the pet store “Zeus peluquería canina.”

The project has been developed to automate the management of customers, pets, products, services, appointments, and billing within the business. The internal part of the web application has been developed with the React JS framework and the use of Firebase as a database manager. The project uses the Scrum methodology, due to the use of various development cycles and being able to keep track of the progress to finish the software in the estimated time.

The document is structured as follows: the first section corresponds to the introduction, disclosing the context of the problem, objectives, and scope of the project. The second section describes the development methodology and the tools that will be used for the implementation of the project. The third section details the analysis and requirements of the web application, the database design, the system architecture and the first development cycle. The fourth section shows the results obtained in the rest of the development cycles and the different tests carried out. Finally, the last section details the conclusions and recommendations obtained at the end of the development of the project.

KEYWORDS: Framework, React, Firebase, JavaScript, Scrum, Aplicación web y tienda de mascotas.

1 CAPÍTULO I: INTRODUCCIÓN

En el presente capítulo se abordará la justificación del por qué se va a desarrollar una aplicación enfocada en la gestión de tiendas de mascotas. Además, se presentará el planteamiento del problema, en el que se identificará un caso específico para la implementación de una aplicación web que ayude con las tareas dentro del negocio. También se expondrán los objetivos del desarrollo de la aplicación y el alcance que tendrá la misma, es decir, hasta el punto en el que corresponde la adecuada implementación del presente trabajo.

1.1 Justificación

En la actualidad tres de cada cinco familias poseen una mascota, según un estudio de la Secretaría de Salud del Distrito Metropolitano de Quito (El Telégrafo, 2015), debido a eso, las personas tienen la necesidad de cuidar de sus mascotas y tratarlas como un miembro más de la familia. En consecuencia, las familias en promedio gastan alrededor de 30 dólares en la compra de artículos y cuidados de su mascota (El Comercio, 2022), por lo que tiendas de mascotas como Zeus Peluquería Canina, tiene como objetivo la venta de productos para mascotas los cuales pueden ser comida, juguetes y accesorios, además de eso brinda varios servicios, entre los cuales son: baño a las mascotas, peluquería y ofrece vitaminas o vacunas en caso de que la mascota lo requiera. Por lo que tener un software que se encargue de gestionar sus procesos, productos, y servicios que son ofrecidos por el establecimiento a los clientes sería de gran ayuda. Por otra parte, según una investigación si una empresa o negocio pequeño no adopta las nuevas tecnologías no logrará operar en su área de manera eficiente, esto también significa que sin un software que satisfaga las demandas locales del negocio no podrá agilizar ciertas tareas dentro del negocio (Madridiario, 2022), por lo tanto ante dicho problema el tener una aplicación web significaría una ayuda muy importante al momento de administrar los diferentes módulos y llevar un control sobre cada uno de ellos, todo esto en beneficio de Zeus Peluquería Canina.

1.2 Planteamiento del problema

La inexistencia de un software adecuado que cumpla con las necesidades locales enfocado en la gestión de tiendas o locales que brindan servicios a las mascotas ha generado que en pequeños negocios la administración de sus servicios sea ineficiente ya que estos se realizan en pequeñas hojas en donde se registra al cliente, la mascota y el servicio ofrecido y también existe el riesgo de perder alguna de estas fichas (Asto, 2020). Además de eso, cada vez que va un cliente al negocio se debe llenar los registros del cliente y de la mascota de nuevo, esto sin poder llevar un control automatizado por parte del administrador del negocio (Jiménez, 2020). Es por eso que para solventar dicho problema en este proyecto se enfocará en Zeus Peluquería Canina como caso de estudio que solviente las necesidades locales (ISOTools, 2017), además de tener una

interfaz amigable para que el proceso de la administración sea sencillo y eficiente. Si bien existe software que ya se encargan de gestionar estos procesos como “GESPET” el cual es un producto extranjero, no hay una aplicación web que trabaje de forma local y se dé a conocer en los pequeños y medianos locales en el país (Gespel, s.f.).

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar una aplicación web para la gestión de productos, clientes, mascotas y servicios que se ofrecen en Zeus Peluquería Canina.

1.3.2 Objetivos específicos

- Analizar los procesos de la tienda Zeus Peluquería Canina.
- Diseñar la arquitectura, base de datos y prototipos de la aplicación web.
- Implementar una aplicación web para la gestión de los procesos de la tienda Zeus Peluquería Canina.
- Realizar pruebas de funcionamiento del sistema.

1.4 Alcance

El objetivo de la aplicación web para la tienda Zeus Peluquería Canina tiene como propósito solventar las necesidades de la tienda, entre las cuales se tiene la administración de varios procesos de la tienda, en la cual se podrá agregar o editar un cliente, mascota, servicio o cita.

Dentro de cada módulo mencionado se deberán registrar los datos necesarios del cliente como pueden ser nombre, apellido, cédula, teléfono, correo electrónico, etc. Por parte de la mascota se deberá registrar el nombre, fecha de nacimiento, tipo de mascota, raza, etc.

Para el manejo del módulo de servicios en el local se tendrán listas de los diferentes servicios que brinda el local con el precio correspondiente, además de eso se tiene que gestionar las citas con el cliente y mascota correspondiente y también verificar si la mascota tiene alguna enfermedad o algo previo a la cita.

1.5 Contenido

En el capítulo 2 se va a presentar los fundamentos teóricos, lo cual está compuesto por el marco teórico, marco referencial y marco conceptual. Además de presentar la metodología que se va a utilizar para el desarrollo del proyecto y las herramientas que serán utilizadas para la implementación correspondiente.

En el capítulo 3 se enseñará el caso de estudio, aquí se realizará el análisis y levantamiento de los requisitos que tendrá la aplicación, el diseño de la base de datos, la arquitectura que tendrá la misma y se comenzará con el primer ciclo de desarrollo, el cual está compuesto por la elaboración de diseños para las diferentes interfaces, la configuración para realizar el desarrollo del proyecto, la creación de la base de datos y la respectiva conexión con el proyecto, además de realizar el sistema de autenticación.

En el capítulo 4 se va a desarrollar la fase de implementación de la aplicación web, se comenzará con el desarrollo de los diferentes módulos que tendrá el sistema y lograr completar toda la funcionalidad que se estableció gracias a los requerimientos, es decir realizar las interfaces y los formularios de las entidades clientes, mascotas, servicios, productos, citas y facturación, además de realizar las pruebas correspondientes del sistema.

En el capítulo 5 se va a presentar las conclusiones y recomendaciones que se obtuvo luego de haber terminado todo el proceso de desarrollo del aplicativo web y haber realizado las pruebas correspondientes para comprobar que está funcionando de manera correcta.

2 CAPÍTULO II: FUNDAMENTOS TEÓRICOS

En el presente capítulo se da una introducción sobre lo que es una aplicación web, los conceptos básicos para entender lo que se está realizando. Además, se describe la metodología que se va a usar en el desarrollo, para lo cual se presentan los roles que se involucran y las herramientas necesarias dentro de la metodología usada. También se presentan los diferentes programas, plataformas o librerías que van a ser usadas para el desarrollo de la aplicación web.

2.1 Situación Actual

2.1.1 Antecedentes o marco referencial

El uso de las aplicaciones para la gestión de negocios dentro del país ha ido aumentando y esto se ha visto reflejado luego del impacto que tuvo la pandemia (Primicias, 2022), debido a esto es que pequeños locales como tiendas de mascotas o veterinarias que están enfocadas en las mascotas, requieren de un sistema para poder gestionar los diferentes módulos y llevar un control mucho más rápido y ágil, por lo que es necesario el uso de aplicaciones enfocadas en cubrir esta demanda local en los establecimientos enfocados en esta área.

2.1.2 Marco teórico

A diferencia de una página web estática, una aplicación web ofrece páginas dinámicas que le permite al usuario interactuar y realizar diferentes actividades según las necesidades del negocio. De esta forma la aplicación web tendrá interfaces más intuitivas con un manejo más rápido y sencillo para el usuario.

En base a esto el tener una aplicación web permite satisfacer las necesidades de manejar cualquier información y poder trabajar de manera mucho más eficiente, además de que esto puede generar mayor productividad, ya que da una respuesta inmediata al momento de solventar procesos dentro de un negocio (Soft, 2021).

2.1.3 Marco conceptual

Para la creación de la aplicación web es necesario contar una metodología ágil para que se pueda gestionar de manera correcta el proyecto y se pueda adaptar a los cambios necesarios. Además, es necesario gestionar los diferentes requerimientos y crear una base de datos, la cual va a ayudar a organizar la información que se tendrá dentro del sistema. También, es necesario seleccionar un lenguaje de programación adecuado para la construcción del aplicativo, en base al lenguaje seleccionado es importante seleccionar un *framework*, el cual facilita el desarrollo e implementación del proyecto. En base a esto vamos a describir cada uno de los elementos más importantes que se tendrá a lo largo del desarrollo.

Metodología Ágil: Una forma de diseño de una aplicación web o móvil para poder asegurar su desarrollo en el menor tiempo posible y poder cumplir con los requerimientos establecidos antes de comenzar con la fase de desarrollo y lograr la satisfacción por parte del cliente (Redhat, s.f.).

Lenguaje de Programación: Forma en que la computadora recibe las órdenes y las ejecuta, de forma secuencial para poder generar un proceso, cada lenguaje de programación está enfocado en cumplir con varias áreas específicas en el desarrollo (SUAYED-FCA, s.f.).

Framework: Un *framework* se podría entender como el conjunto de herramientas establecidas en un lenguaje de programación para poder utilizarlas, mediante esto todos los elementos que puede tener el *framework* es para poder desarrollar una aplicación de manera mucho más fácil y ágil.

Base de Datos: La base de datos no es más que la organización de datos, con la cual se puede gestionar cualquier tipo de sistema, una base de datos está controlada por un sistema gestor de base de datos, con el cual vamos a poder implementarlo con el lenguaje de programación correspondiente y usarlo para satisfacer el propósito de una aplicación web (Oracle, s.f.).

2.2 Metodología de desarrollo

A continuación, se menciona la metodología sobre la cual está realizada el presente proyecto, la cual ha sido seleccionada debido a que se adapta en el enfoque de la resolución del problema en el tiempo que se tiene para la implementación y desarrollo del mismo, además de solventar el problema de forma rápida.

Este proyecto está desarrollado bajo la metodología ágil *Scrum*, la cual, al ser enfocada en entrega de trabajos en un periodo de tiempo corto, es la mejor solución debido a su flexibilidad para con las necesidades que requiera el desarrollo porque permite realizar cambios de forma continua y esto ayuda a que se satisfaga los requerimientos del proyecto (Sordo, 2023).

Scrum trabaja bajo *sprints*, los cuales son intervalos de tiempo para realizar determinadas actividades con respecto al proyecto, cada *sprint* se tendrá que cumplir con las tareas previstas sin realizar cambios de última hora, al final de cada *sprint* se debe realizar su respectiva revisión para analizar y evaluar los cambios o correcciones pertinentes sobre lo trabajado y que esto sea en favor de los siguientes *sprints*, en ciertas ocasiones cuando se trabaja en grupos numerosos, es de gran ayuda contar con retroalimentación de los demás miembros del equipo. Esto permite que se puedan aportar ideas y sugerencias para la corrección y mejora del trabajo de cada integrante del equipo (Conexión Esan, 2018).

2.2.1 Roles

Dentro de la metodología ágil que se utiliza en el proyecto existen varios roles, los cuales ayudan a que cada persona involucrada tenga tareas asignadas durante cada *sprint*.

Product Owner: Es el responsable de maximizar el valor del trabajo, además de ser el único que interactúa directamente con el cliente el cual le pone en conocimiento sobre lo que requiere el proyecto (de Dios, 2022).

Scrum Master: Es la persona que se encarga del equipo de desarrollo y solventa los inconvenientes que pueden aparecer en el transcurso del desarrollo de cada *sprint*, además de eso debe ser una persona que comprenda los valores y principios como el trabajo en equipo, respeto, compromiso y sinceridad (de Dios, 2022).

Development Team: Son todos los encargados de desarrollar tareas en función del *sprint* correspondiente, las tareas son priorizadas en base a lo que proporcione el *Product Owner* (de Dios, 2022).

Para el presente proyecto se tendrá la asignación de roles siguiente como se presenta en la **Tabla 1**.

Tabla 1: Asignación de roles

Rol	Responsable
<i>Product Owner</i>	Andrés Alvear
<i>Scrum Master</i>	Fabian de la Cruz
<i>Development Team</i>	Andrés Alvear

2.2.2 Herramientas de Scrum

Para el desarrollo del proyecto se deben tener un orden para ir registrando la información correspondiente con respecto a las actividades realizadas del proyecto, por lo cual se debe tener en cuenta la recopilación de requerimiento, el *Product Backlog*, las historias de usuario y el *Sprint Backlog*.

Recopilación de Requerimientos: El principal objetivo es analizar las diferentes actividades que tendrá el sistema, es parte fundamental conocer las funciones que se tendrán para poder ir trabajando en los respectivos avances.

Product Backlog: Es un listado de todas las actividades que se establecieron en base a los requerimientos del cliente para desarrollar el producto final (Programacionymas, s.f.).

Sprint Backlog: Cada una de las actividades del *Product Backlog* se pueden agrupar para que correspondan a un conjunto de tareas o sprint, en el cual se deberán cumplir las actividades para que el desarrollo se cumpla en el tiempo establecido en cada *sprint* (EALDE, s.f.).

Historias de Usuario: Son descripciones cortas y sencillas de entender de una actividad correspondiente del *Product Backlog* (SCRUM MEXICO, 2018). Un ejemplo puede ser la que corresponde a la **Tabla 2**.

Tabla 2: Ejemplo de Historia de Usuario

Historia de Usuario	
Identificador: HU01	Usuario: Administrador
Nombre historia: Iniciar y Cerrar Sesión	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Sprint asignado: 1
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema con un correo y contraseña que se encuentre registrado en la base de datos.	
Observaciones: El usuario cierra sesión luego de que haya seleccionado el botón que se encuentra en la pantalla de inicio.	

2.3 Herramientas de desarrollo

Para el desarrollo del sistema se componen de las siguientes herramientas, para la gestión de base de datos, almacenamiento de imágenes y sistema de autenticación de usuarios se realiza con *Firebase*, mientras que la implementación del sistema se lo trabaja en el *Framework React* utilizando el lenguaje de programación *JavaScript*, apoyado de librerías de NPM o herramientas de creación de elementos de interfaz gráfica como MUI. Todo esto se verá reflejado en el correspondiente navegador de preferencia presentando al cliente una página web con los requerimientos correspondientes. En la

Tabla 3 se realiza la respectiva especificación de las herramientas que se presentaron.

Tabla 3: Herramientas de desarrollo

Herramienta	Justificación
Firestore	Plataforma de desarrollo de aplicaciones para el almacenamiento de datos en la nube, acelera el desarrollo de aplicaciones gracias a las funciones de que posee con una infraestructura <i>backend</i> (Firestore, s.f.). Se seleccionó como gestor de bases de datos debido a su fácil manejo e implementación en un sistema web.
React	Una librería de <i>Javascript</i> enfocada en el desarrollo de interfaces de usuario lo cual hace que las aplicaciones de página simple o SPA sean mucho más veloces porque todo funciona en una sola página web, además de que se trabaja bajo componentes, lo que hace que se puedan reutilizar y armar de forma sencilla (React, s.f.). Se utiliza esta librería o Framework debido a la facilidad que aporta al momento de construir interfaces gráficas y su uso de componentes.
JavaScript	Es un lenguaje de programación enfocado en el desarrollo web el cual ayuda a que las páginas web sean interactivas con el usuario (AWS, s.f.). Al momento de utilizar un Framework como React se tiene la posibilidad de usar JavaScript o <i>Typescript</i> , y se seleccionó la primera opción debido al conocimiento y facilidad de manejo del lenguaje de programación.
Material-UI	Es una biblioteca de componentes para React que aportan en el desarrollo de las interfaces gráficas (Gonzalez, s.f.). Se utiliza esta biblioteca de componentes principalmente por la facilidad al momento de usar botones, formularios, íconos, entre otros elementos que pueden estar dentro del sistema web.
NPM	Manejador de paquetes de <i>Node</i> es el entorno de ejecución en el lado del servidor, brinda el uso de paquetes de software para la elaboración de proyectos que han sido desarrollados por programadores alrededor del mundo (Hernandez, 2021). Se utiliza ya que sin npm no es posible

trabajar con paquetes o librerías como React, por lo que es necesario instalarlo para hacer uso de los paquetes o librerías que sean requeridas en el desarrollo del proyecto.

Visual Studio Code Es un editor de código desarrollado por *Microsoft* de software libre, lo cual permite que se le puedan agregar extensiones para que el desarrollo sea mucho más amigable con el desarrollador (Flores, 2022). El editor fue seleccionado porque es ligero en cuanto a su tamaño dentro del computador y permite utilizar muchos lenguajes de programación, por eso y su manejo con anterioridad fue seleccionado.

3 CAPÍTULO III: CASO DE ESTUDIO

En el presente capítulo se va a describir los diferentes requerimientos que serán utilizados en la aplicación, los cuales serán realizados en diferentes ciclos de desarrollo. Luego se realizará la base de datos en base a los requerimientos que se tengan. También se presentará la arquitectura del proyecto y se comenzará con el primer ciclo de desarrollo del trabajo, el cual tendrá la creación del diseño de las interfaces, la creación de la base de datos en el respectivo gestor, la configuración del proyecto y la conexión necesaria. Además de comenzar con el sistema de autenticación para el usuario.

3.1 Análisis y levantamiento de requerimientos

Para empezar con el desarrollo del proyecto primero es necesario conocer acerca de lo que se va a realizar, en este caso para el desarrollo de una aplicación web para la gestión de tiendas de mascotas, centrándose en Zeus Peluquería Canina, se debe entender el funcionamiento que tiene la tienda, para eso se debe tener una comunicación adecuada con la persona a cargo del negocio. En base a eso se conoce que en la tienda la parte fundamental es el ofrecer servicios a las mascotas, por lo que se procede a comprender como funciona el proceso desde que llega un cliente para solicitar una cita de un determinado servicio para su mascota. El proceso que se tiene es en cada cita llenar de forma manual la ficha que se muestra en la **Figura 1**, y esto hace que cada ficha se almacene en cualquier sitio de la tienda sin guardar la información en caso de que sea necesaria en una cita futura.

Figura 1: Ficha Tienda

El formulario 'Ficha Tienda' de Zeus Peluquería Canina está dividido en varias secciones:

- ENCABEZADO:** Logo de Zeus Peluquería Canina, 'EXPEDIENTE ESTETICA ANIMAL' y 'FECHA'.
- DATOS GENERALES:** 'HORA DE TURNO'.
- MASCOTA:** Campos para Nombre, Especie, Raza, Edad, Vacunas, Alergias y Peso.
- PROPIETARIO:** Campos para Nombre, Teléfonos, Domicilio y Correo.
- EXAMEN FISICO:** Sección con tres íconos de animales (gato, perro, perro) para registrar el estado de salud.
- SERVICIOS ESTETICOS:** Incluye 'TAMAÑO' con íconos de perros de diferentes tamaños, 'COSTO' con una flecha hacia abajo y 'ESPECIFICACIONES CORTE DE PELO'.
- SERVICIO ADICIONALES:** Lista de servicios con casillas de verificación: Baño cosmético, Limpieza oídos, Pedicura, Limpieza dental, Limpieza glándulas, Baño medicado antipulgas, Desparasitación, Profilaxis dental, Vacunas, Esterilización.
- NOVEDADES GROOMER / VETERINARIO:** Espacio para registrar novedades.
- VALOR DEL TRABAJO:** Campos para 'ABONO' y 'SALDO'.
- FIRMAS:** Sección para 'FIRMA DE RECIBO CONFORME' y 'FIRMA AUTORIZACION DE TRABAJO CON LA MASCOTA'.

Luego de haber tenido la ficha como base para el sistema e ir conociendo cuales son los componentes más esenciales desde que llega un cliente hasta que se va con su mascota luego de haber sido atendido satisfactoriamente. Se conoce que los elementos necesarios para que se gestionen en una aplicación web son clientes, mascotas, servicios, productos y citas. Cada una de estas entidades serán parte fundamental e indispensable para el proyecto. Es por eso por lo que se procede a realizar un listado de las diferentes tareas que tendrá el sistema web en la **Tabla 4.**

Tabla 4: Requerimientos Funcionales

Id	Requerimiento	Descripción
RF1	Iniciar y cerrar sesión	El administrador podrá iniciar sesión mediante la pantalla de <i>login</i> . Dentro del sistema puede cerrar sesión en la pantalla de inicio.
RF2	Visualizar, crear y editar Clientes	El administrador podrá crear un nuevo cliente con los datos correspondientes al formulario. Editar los datos en caso de que haya alguna equivocación. Visualizar la información de todos los clientes en una tabla.
RF3	Visualizar, crear y editar Mascotas	El administrador podrá crear una nueva mascota con los datos correspondientes al formulario, relacionándolo con un cliente. Editar los datos en caso de que haya alguna equivocación. Visualizar la información de todas las mascotas en una tabla.
RF4	Visualizar, crear y editar Servicios	El administrador podrá crear un nuevo servicio con los datos correspondientes al formulario. Editar los datos en caso de que haya alguna equivocación. Visualizar la información de todos los servicios en una tabla.
RF5	Visualizar, crear y editar Productos	El administrador podrá crear un nuevo producto con los datos correspondientes al formulario. Editar los datos en caso de que haya alguna equivocación. Visualizar la información de todos los productos en una tabla.

RF6	Visualizar, crear y editar Citas	El administrador podrá crear una nueva cita con los datos correspondientes al formulario. Editar los datos en caso de que haya alguna equivocación. Visualizar la información de todas las citas en una tabla.
RF7	Visualizar y crear Factura	El administrador podrá visualizar la información de la factura, además de poder crearla.
RF8	Recuperar contraseña	El administrador podrá restablecer la contraseña mediante el envío de un correo electrónico.

En base a cada uno de los requerimientos que se han establecido, se procederá a trabajar en 3 ciclos de desarrollo, los cuales están detallados en la **Tabla 5**.

Tabla 5: Planificación de sprints

Ciclo de desarrollo	Actividades
Primer ciclo	En el primer ciclo de desarrollo se va a comenzar con el diseño de las interfaces, las cuales servirán de guía para poder ir desarrollándolas en el siguiente ciclo, además, se realizará la configuración adecuada del entorno de desarrollo con las herramientas establecidas. También se comenzará con el sistema de autenticación conectado con Firebase.
Segundo ciclo	En el segundo ciclo de desarrollo se comenzará con la creación del CRUD de cada entidad dentro del sistema, es decir se realizarán los formularios de creación, edición y visualización para clientes, mascotas, productos, servicios, citas y facturación.
Tercer ciclo	En el tercer ciclo se realizará el proceso de recuperación de contraseña para el usuario, además se va a mejorar las vistas de todas las pantallas para que se vean de la mejor manera posible, finalmente se va a realizar las pruebas correspondientes del sistema.

Ahora para ir trabajando dentro de cada ciclo se puede ir relacionando cada requerimiento dentro de historias de usuario para luego ir seleccionándolas dentro del *sprint* correspondiente. Como se puede ver a continuación en la **Tabla 6**.

Tabla 6: Historia de Usuario de Registro de Cliente

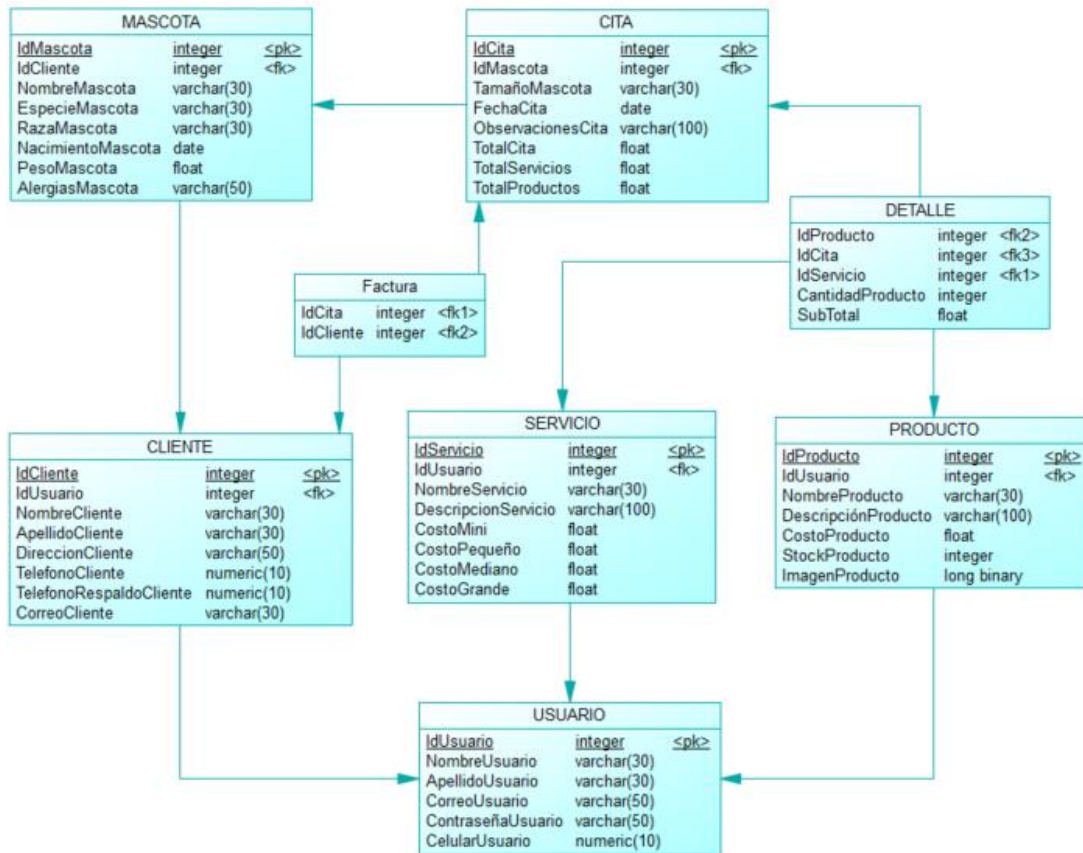
Historia de Usuario	
Identificador: HU02	Usuario: Administrador
Nombre historia: Crear nuevo cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y debe llenar el formulario para el registro con los campos de: nombre, apellido, domicilio, teléfono, teléfono de respaldo y correo electrónico.	
Observaciones: El sistema verificará que no existan clientes repetidos. El campo de teléfono de respaldo es opcional.	

3.2 Diseño de la base de datos y arquitectura del sistema

3.2.1 Base de Datos

En la **Figura 2** se presenta el diagrama entidad-relación de la base de datos, la cual sirve de guía para la elaboración del proyecto, a pesar de que *Firebase* no está enfocado en bases de datos relacionales se debe tener una idea con respecto a las relaciones correspondientes entre las diferentes entidades para lograr que el sistema interactúe de manera adecuada.

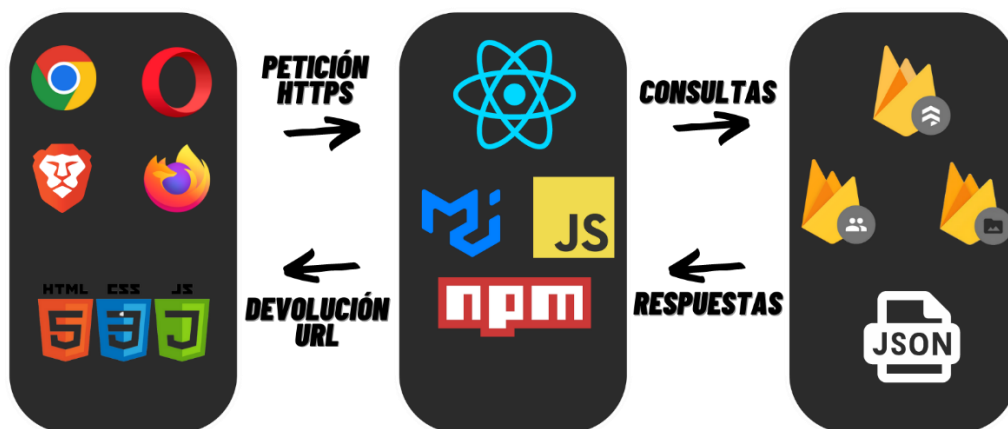
Figura 2: Base de datos



3.2.2 Arquitectura

La arquitectura en el sistema web está compuesto por la parte de base de datos, las restricciones y la comunicación entre el la interfaz gráfica y la base de datos es en *Firebase*, mientras que la parte de la interfaz gráfica se la realiza en *React*, todo esto para que sea consumido por el cliente en el navegador deseado, la arquitectura se puede apreciar de forma más visual en la **Figura 3**.

Figura 3: Arquitectura



3.3 Primer ciclo de desarrollo

En el primer ciclo de desarrollo empezará con los diseños para poder usarlos de guía para desarrollarlos más adelante, además de eso se realizará la configuración del ambiente de desarrollo sobre el cual se va a trabajar. Además, se comenzará con la creación de la base de datos en *Firebase*, luego de eso ya se podrá continuar con la primera parte del desarrollo del proyecto que será el sistema de autenticación, el cual estará conectado con la base de datos y se podrá realizar el inicio y cierre de sesión.

3.3.1 Diseño de interfaces

Antes de empezar con el desarrollo del sistema web, es necesario sentar las bases necesarias para que se pueda realizar el trabajo de manera correcta, para ello ha sido fundamental realizar el modelaje de las pantallas que tendrá la aplicación. Uno de los ejemplos que se tiene respecto a los prototipos se puede ver en **Figura 4** y **Figura 5**.

Figura 4: Prototipo de inicio de sesión

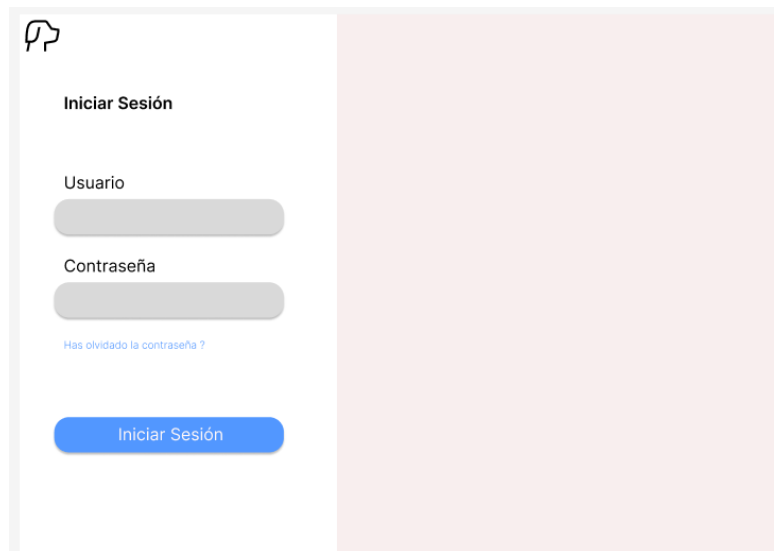
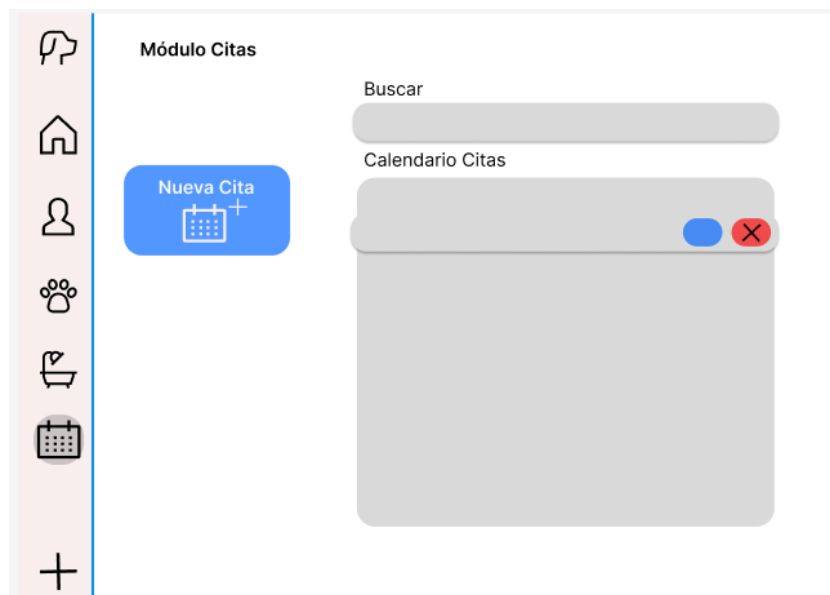


Figura 5: Prototipo de módulo citas



3.3.2 Configuración del ambiente de desarrollo

Ya teniendo la base de datos y los prototipos que se pueden seguir como guía para realizar el proyecto, vamos a utilizar las herramientas necesarias con sus respectivas versiones que se muestra en la **Tabla 7**.

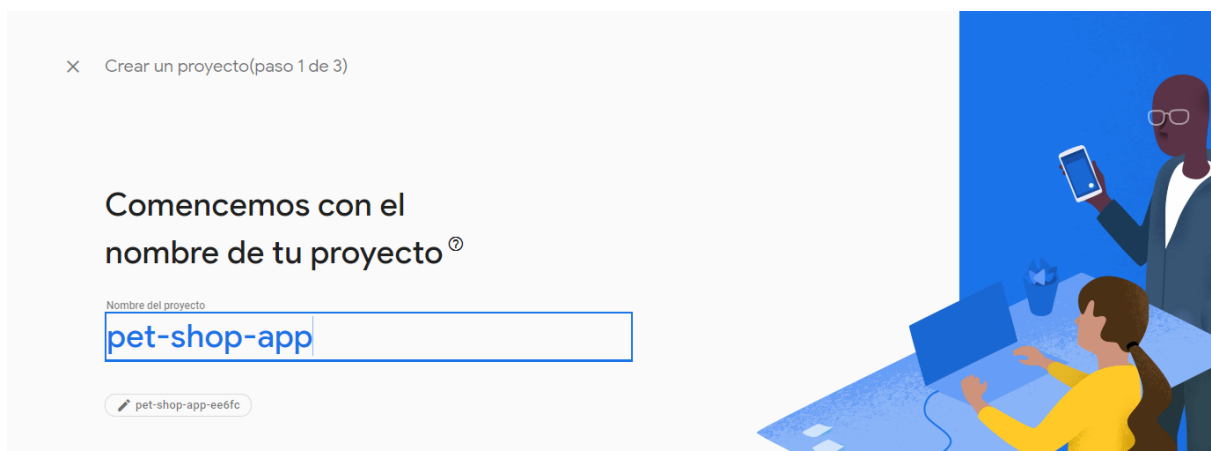
Tabla 7: Versiones de Herramientas

Herramienta	Versión
React	18.2.0
Firebase	9.19.1
MUI	5.11.0
Node JS	18.15.0
NPM	9.6.3
Visual Studio Code	1.77.3

3.3.3 Creación de base de datos en Firebase

Para empezar a utilizar la base de datos debemos crear un proyecto, el cual se vinculará con el proyecto en *React* con las credenciales correspondientes, luego de eso ya podemos hacer lo que nos ofrece Firebase, en este caso usaremos el servicio de autenticación para que sea mucho más sencillo la creación de usuarios para que puedan realizar el inicio de sesión correspondiente, también tenemos el servicio de *Firestore* en el cual se van a crear las colecciones, es decir hace referencia a todas las entidades que tendría en una base de datos relacional. Con todo esto ya se puede empezar con el desarrollo con el *Framework*.

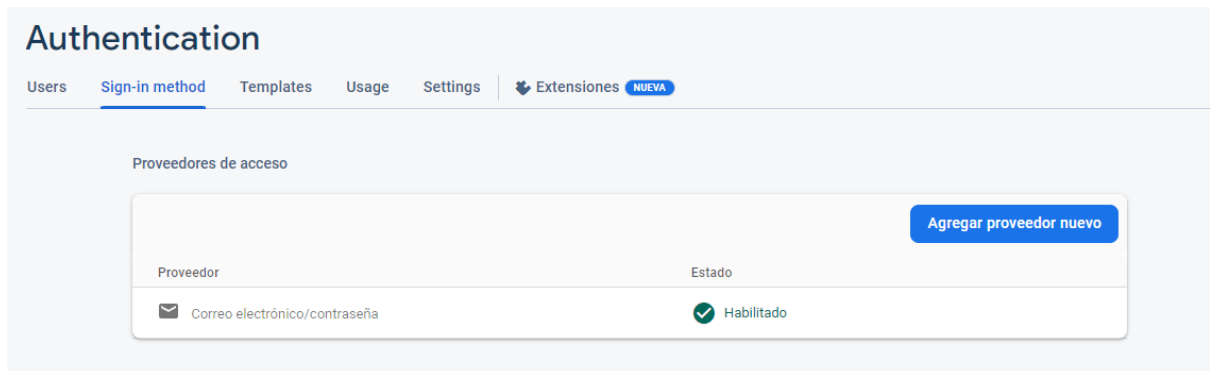
Figura 6: Creación de base de datos en Firebase



3.3.4 Sistema de autenticación

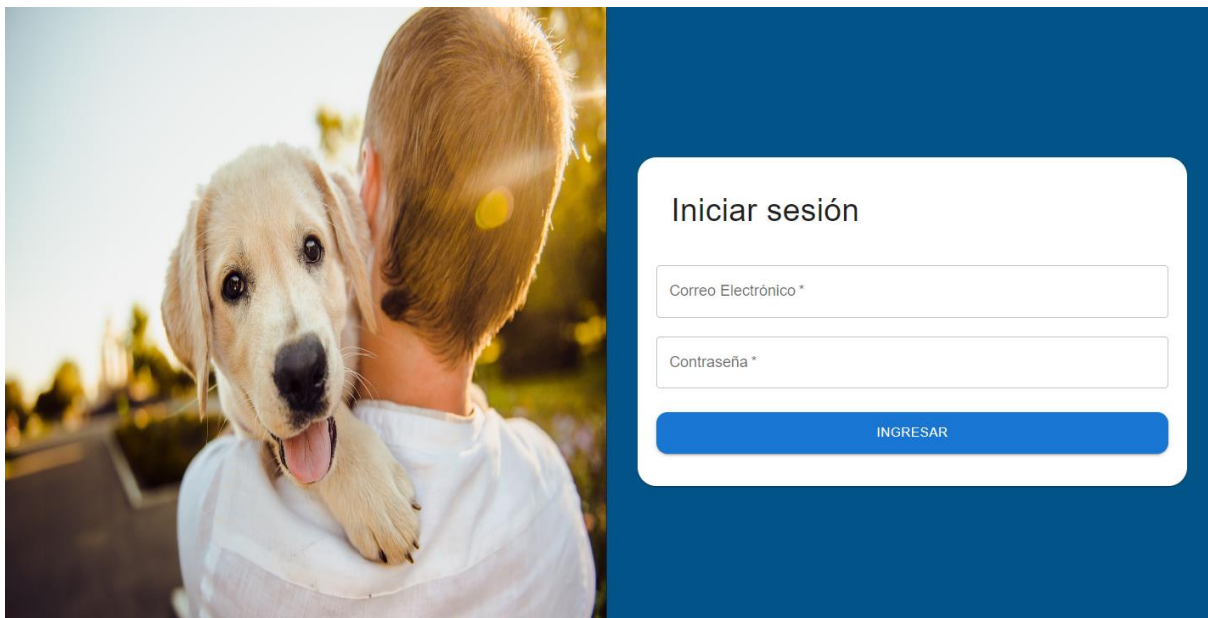
Para hacer uso de las funciones que nos brinda *Firebase* se deben activar desde la consola del proyecto correspondiente, para lo cual en este proyecto vamos a usar un inicio de sesión con el correo electrónico y una contraseña. Esto se lo activa y se utilizan las funciones dentro del código.

Figura 7: Autenticación de Firebase



Con esto ya se desarrolla el funcionamiento del inicio de sesión y la creación correspondiente de las pantallas para que lleven al inicio del sistema web, el mismo que tiene el botón correspondiente para que el administrador pueda cerrar sesión.

Figura 8: Pantalla de iniciar sesión



4 CAPÍTULO IV: IMPLEMENTACIÓN

En el siguiente capítulo se comenzará con la implementación del segundo ciclo de desarrollo, el cual tiene los formularios de cada una de las entidades, para ingresar nuevos datos, editarlos y poder visualizarlos. Además de realizar las pruebas correspondientes para la aplicación para comprobar el correcto funcionamiento

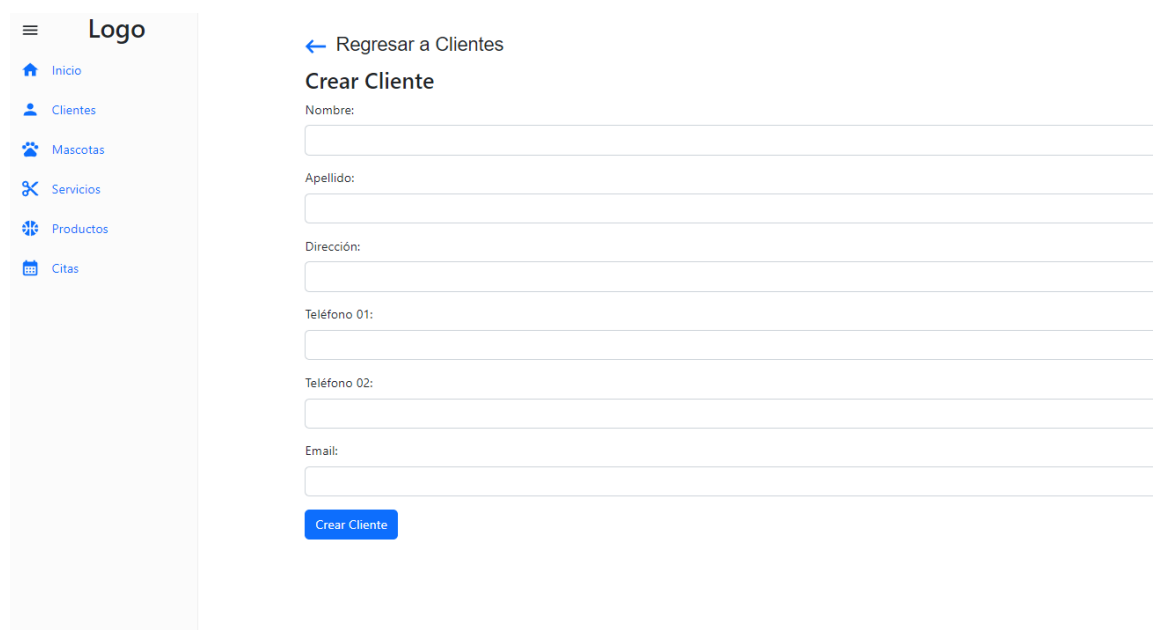
4.1 Segundo ciclo de desarrollo

En este ciclo de desarrollo se contempla realizar la mayor parte del sistema, el cual corresponde a la creación, visualización y edición de las diferentes entidades que tienen, todas estas actividades hacen parte de un concepto el cual es *CRUD*, que significa *Create, Read, Update y Delete*, son las cuatro operaciones más simples y comunes que se realiza en un sistema, por lo que en este caso debemos aplicar un *CRUD* para cada entidad correspondiente, únicamente no se debería utilizar el *Delete* o eliminar debido a que puede generar conflictos dentro de la base de datos y además la integridad de datos se vería afectada. (Durán, 2023)

4.1.1 CRUD de clientes

Empezando con la primera entidad del sistema que será clientes, aquí se deberá ingresar a una ruta en donde se podrá agregar un nuevo cliente, editarlo y revisar el listado de todos los clientes que se encuentren registrados. Además de eso debe al momento de seleccionar el botón correspondiente para crea un nuevo cliente o editarlo deberá aparecer el formulario con los campos establecidos.

Figura 9: Formulario de creación de clientes



The image shows a web application interface. On the left is a sidebar menu with a 'Logo' at the top and a hamburger menu icon. Below the logo are several menu items: 'Inicio' (home icon), 'Clientes' (person icon), 'Mascotas' (paw print icon), 'Servicios' (wrench icon), 'Productos' (gears icon), and 'Citas' (calendar icon). The main content area is titled 'Crear Cliente' and has a back link 'Regresar a Clientes'. The form contains the following fields: 'Nombre:' (text input), 'Apellido:' (text input), 'Dirección:' (text input), 'Teléfono 01:' (text input), 'Teléfono 02:' (text input), and 'Email:' (text input). At the bottom of the form is a blue button labeled 'Crear Cliente'.

Figura 10: Vista de módulo clientes

Módulo Clientes

[NUEVO CLIENTE](#)

Nombre	Apellido	Dirección	Teléfono 01	Teléfono 02	Correo	Acciones
Andrés	Alvear	Quito	0987456123		andres_alvear@email.com	Editar
Juan	Perez	Quito	0987412546		jperez@email.com	Editar

Figura 11: Vista editar cliente

[← Regresar a Clientes](#)

Actualizar Cliente

Nombre:

Apellido:

Dirección:

Teléfono 01:

Teléfono 02:

Email:

[Actualizar Cliente](#)

4.1.2 CRUD de mascota

Para continuar con la segunda entidad del proyecto es necesario haber realizado la respectiva creación del módulo clientes, debido a que la entidad mascotas se relaciona con clientes, como se puede observar en la base de datos, una mascota está relacionada con un cliente, por lo tanto se procede a crear el respectiva módulo de mascotas en *Firebase*, y a crear el respectivo formulario para que se puedan agregar las mascotas correspondientes, para lo cual se envían los datos a la base de datos luego de presionar el botón de crear mascota, como se puede observar en la **Figura 12**.

Figura 12: Pantalla de Mascotas

Logo

- Inicio
- Clientes
- Mascotas
- Servicios
- Productos
- Citas

← Regresar a Mascotas

Crear Mascota

Nombre:

Especie:

Raza:

Año Nacimiento:

Peso:

Dueño:

Alergias:

[Crear Mascota](#)

Figura 13: Vista módulo mascotas

Módulo Mascotas

[NUEVA MASCOTA](#)

Nombre	Especie	Raza	Año Nacimiento	Peso	Dueño	Alergias	Acciones
Zeus	Perro	M	2020	20	Juan Perez	aa	Editar

4.1.3 CRUD de servicios

Para realizar el módulo de servicios se va a realizar el mismo proceso que con el módulo de clientes, debido a que no dependen de alguna entidad, por lo que se realiza la vista para poder observar todos los servicios correspondientes, además de crear la página en donde se va a realizar el ingreso de un nuevo servicio, para lo cual se va a ingresar los campos de forma manual por parte del usuario para luego ser almacenados en una tabla en *Firebase*, para poder realizar los cambios correspondientes en la tabla si así lo requiere el cliente. Para lo cual se tiene la vista de la creación del formulario en la **Figura 14**.

Figura 14: Pantalla de Servicios

Logo

Inicio

Clientes

Mascotas

Servicios

Productos

Citas

← Regresar a Servicios

Crear Servicio

Nombre:

Descripción:

Costo Mini:

Costo Pequeño:

Costo Mediano:

Costo Grande:

Crear Servicio

Figura 15: Vista módulo de servicios

Módulo Servicios

NUEVO SERVICIO

Nombre	Descripción	Costo Mini	Costo Pequeño	Costo Mediano	Costo Grande	Acciones
Servicio 2	Desc. servicio 2	22	12	15	20	Editar
Servicio 3	Descp. 3	20	25	28	30	Editar
Servicio Ejemplo	Descripcion ejemplo	5	20	30	40	Editar

4.1.4 CRUD de productos

Con el módulo productos, al igual que en un servicio se puede crear sin la necesidad de que se tenga una relación con respecto a otra entidad, para lo cual se sigue el mismo proceso en donde se va a crear una vista para poder observar los productos creados, el botón correspondiente para que se pueda crear un nuevo producto. Para que se pueda crear un producto se deben llenar los campos que se pueden observar en la **Figura 16**, a diferencia de las demás entidades este se puede agregar una imagen la cual se almacena en *Firebase*, debido a esto no existe un inconveniente para su uso. Finalmente se puede realizar los cambios si así lo desea el cliente de cada producto que haya ingresado.

Figura 16: Pantalla de Productos

The screenshot shows a web application interface for creating a product. On the left is a sidebar menu with a hamburger icon and the word 'Logo'. The menu items are: Inicio (home icon), Clientes (person icon), Mascotas (paw print icon), Servicios (wrench icon), Productos (gear icon), and Citas (calendar icon). The main content area is titled 'Crear Producto' and has a back arrow labeled 'Regresar a Productos'. Below the title are five input fields: 'Nombre:' (empty), 'Descripción:' (empty), 'Costo:' (containing '0'), 'Stock:' (containing '0'), and 'Imagen:' (a file selection area with 'Seleccionar archivo' and 'Sin archivos seleccionados' text). At the bottom of the form is a blue button labeled 'Crear Producto'.

4.1.5 CRUD de citas

En el módulo de Citas, es necesario haber realizado los diferentes CRUD's de las entidades mencionadas, debido a que, en cada Cita se deben tener varios elementos como el cliente, la mascota que corresponde a dicho cliente, los servicios que se van a ofrecer en la cita y de ser el caso los productos para la cita. A diferencia de los otros formularios que hay que llenar la mayoría de los campos existen algunos que se completaran de forma automática como puede ser el caso del cálculo del valor de los servicios que han sido seleccionados, los productos de igual manera tendrán un campo para verificar el total de estos y se tiene el campo del total de la cita que será la cantidad que el cliente deberá pagar. Esto se lo puede observar en el formulario que tendrá el módulo Cita que se puede observar en **Figura 17** y **Figura 18**.

Figura 17: Formulario Cita Parte 1

[← Regresar a Citas](#)

Crear Cita

Cliente:

Tamaño Mascota:

Fecha Cita:

Servicios:

- Servicio 2
- Servicio 3
- Servicio Ejemplo

Figura 18: Formulario Cita Parte 2

Comentarios:

Valor Servicios:

Valor Productos:

Valor Total:

Status:

Figura 19: Vista módulo de citas

Módulo Citas

Ciente	Mascota	Tamaño Mascota	Fecha Cita	Servicios	Productos	Comentarios Cita	Valor Servicios	Valor Productos	Valor Total	Estado Cita	Acciones
Juan Perez	Zeus	miniCost	2023-05-03	• Servicio 3			20	0	20	Cancelado	<input type="button" value="Editar"/>

4.1.6 CRUD de facturación

Para realizar la automatización de facturación en este proyecto, es necesario tener el resto de las entidades funcionando de manera correcta para que se pueda apreciar por parte del cliente, en la

pantalla de Facturación se pueden observar las diferentes citas que se tendrán a la fecha actual, por lo que se podrá acceder a cada una de ellas, observar los detalles y seleccionar lo que se hará con cada una de ellas, es decir se podrá cancelar la cita o se le puede dar el estado de pagado. Al momento de que la cita es pagada los datos pasarán a la tabla de factura, en donde se va a tener únicamente las citas que se han completado de forma exitosa. Como se puede ver **Figura 20** y **Figura 21**, la cita que se tiene a la fecha actual se podrá observar para realizar el pago correspondiente o cancelarla en caso de que así sea, al momento de que una cita es creada se almacena con el estado de pendiente para que se mantenga así hasta que sea pagada o cancelada por parte del administrador del negocio.

Figura 20: Pantalla de citas de la fecha actual

Citas de Hoy

















 Juan Perez  Zeus	 Servicio 3 	\$ 20 Pendiente	VER DETALLE
 Juan Perez  Zeus	 Servicio 2 Servicio Ejemplo 	\$ 27 Pagado	
 Juan Perez  Zeus	 Servicio 2 	\$ 22 Pagado	
 Juan Perez  Zeus	 Servicio 2 Servicio 3 	\$ 42 Pendiente	VER DETALLE

Figura 21: Pantalla para realizar el pago

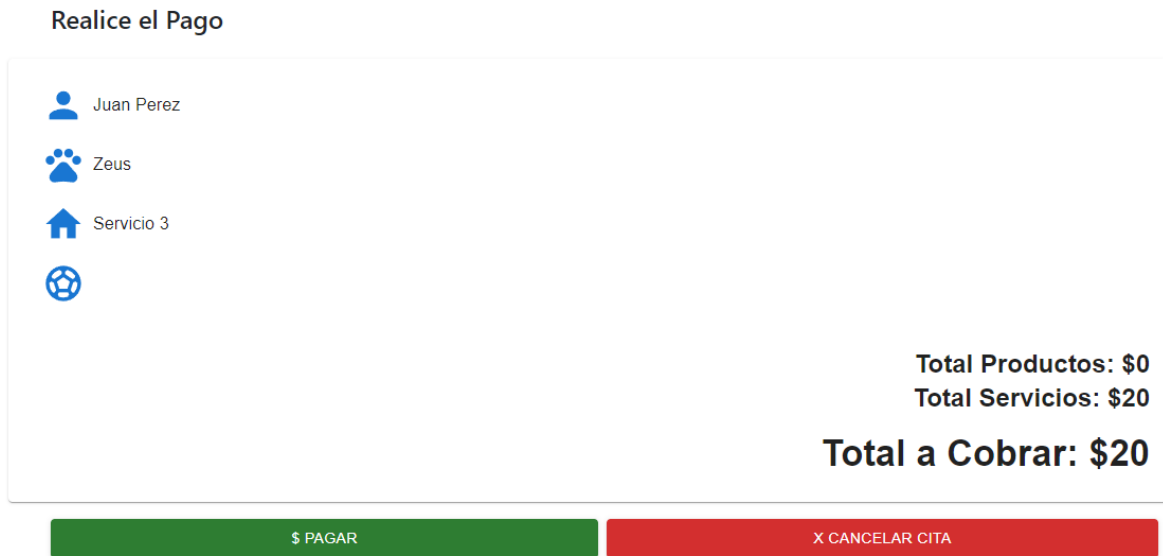
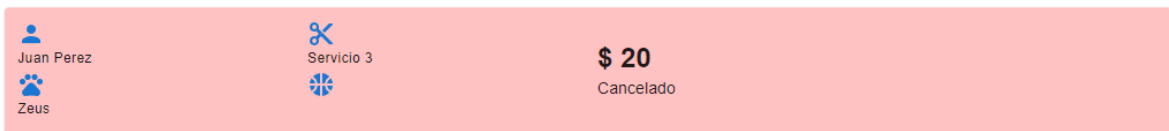


Figura 22: Cita que ha sido cancelada



4.1.7 Tercer ciclo de desarrollo

En el tercer ciclo del desarrollo corresponde a la parte final de la implementación del sistema, por lo cual se realiza la obtención de los datos para que se puedan presentar la pantalla de inicio para que el administrador pueda observar con respecto a las citas que tendrá en la fecha correspondiente y el valor que ha ingresado luego de haber completado una cita, es decir cuánto dinero lleva generado con respecto al día en que está trabajando, esto se puede observar en la **Figura 23.**

Figura 23: Pantalla de inicio



Además de esto se realiza el proceso de recuperación de contraseña por parte del administrador, en caso de que no pueda recuperar la contraseña para poder iniciar sesión, le llegará el respectivo enlace para que pueda ingresar sin problema. Para eso, en el formulario de inicio aparecerá un enlace que llevará a una pantalla para que se pueda realizar el proceso de cambio de la contraseña como se puede ver en la **Figura 24**.

Figura 24: Formulario de recuperación de contraseña



Una vez realizado el proceso de ingresar el correo en el campo solicitado, llegará al correo electrónico un mensaje para que se pueda realizar el cambio de contraseña.

Figura 25: Mensaje de cambio de contraseña en el correo

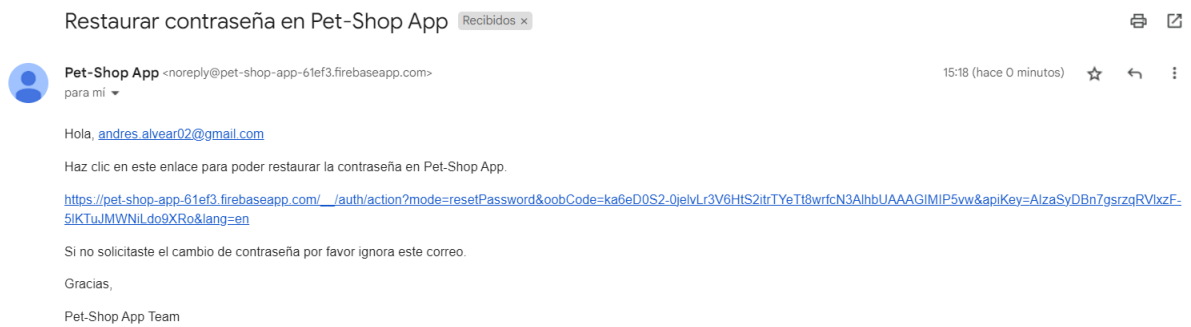


Figura 26: Cambio de contraseña

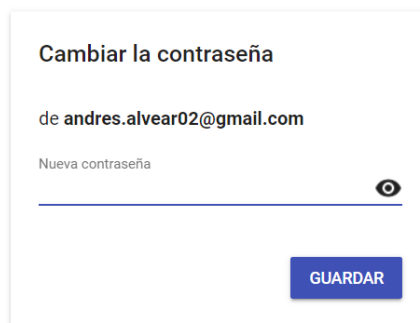
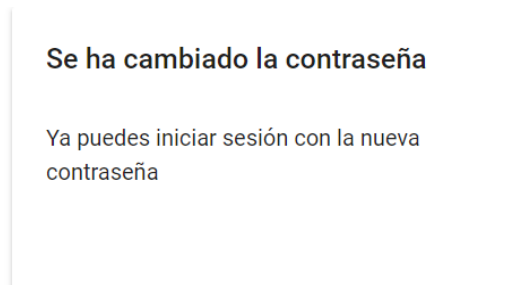


Figura 27: Verificación del cambio de contraseña realizado



Además del proceso realizado anteriormente se procede con la mejora del aspecto visual para que se pueda apreciar de mejor manera las diferentes interfaces gráficas de la aplicación.

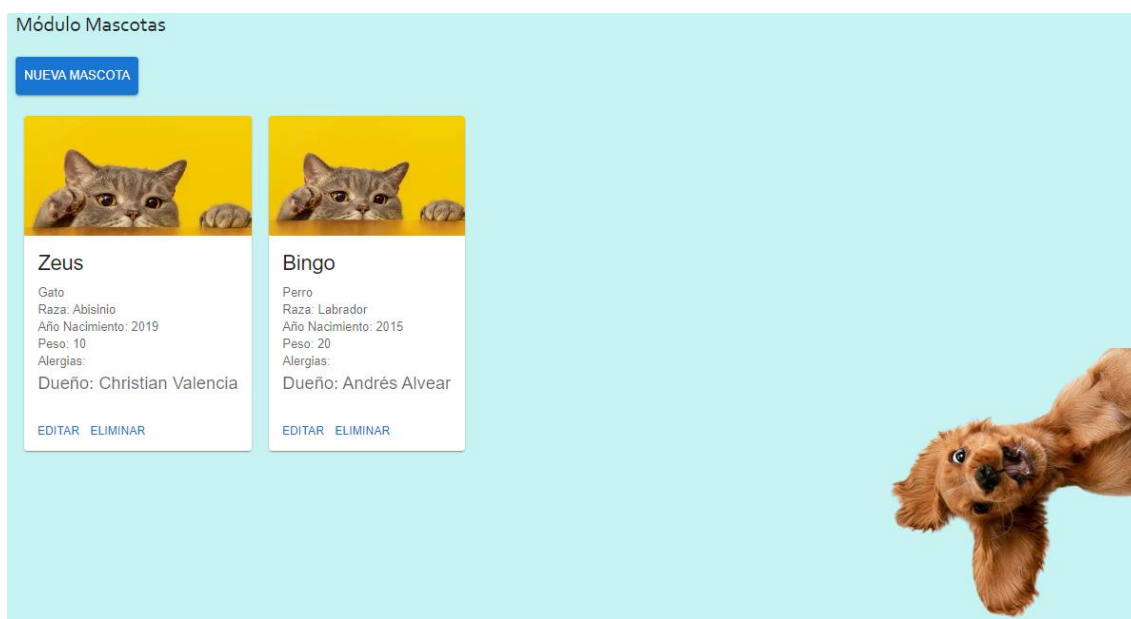
Figura 28: Página de Inicio



Figura 29: Página de productos



Figura 30: Página de mascotas



4.2 Pruebas del sistema

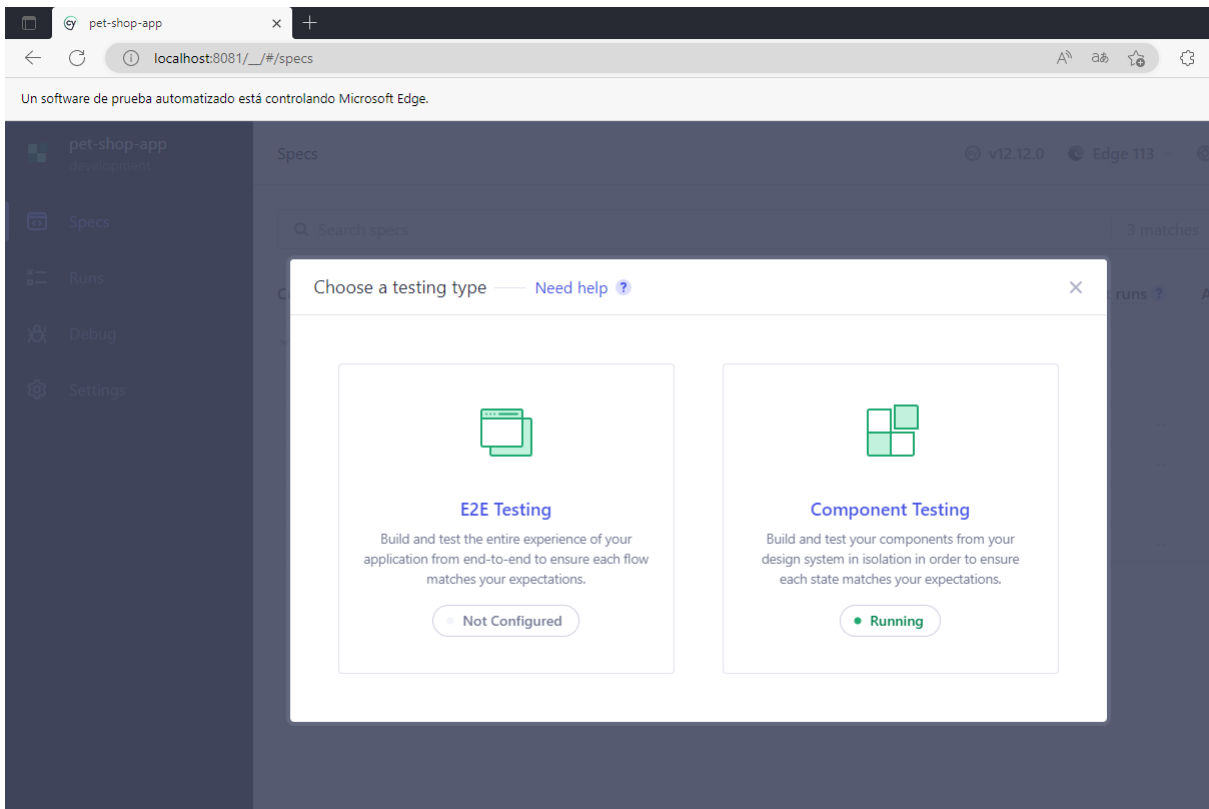
Las pruebas dentro de un sistema son importantes para conocer el correcto funcionamiento del sistema. Su uso permite verificar si la integración de los módulos es adecuada y si cumplen con los requerimientos establecidos (Cillero, 2016).

4.2.1 Pruebas Unitarias

Una de las pruebas más fáciles de realizar y que resultan muy útiles, son las pruebas unitarias, las cuales se enfocan en comprobar el correcto funcionamiento de una parte específica del código, es decir se puede comprobar si un módulo realiza su tarea de manera satisfactoria o si cumple con su propósito (Yeeply, 2022).

Para realizar las pruebas unitarias, se utilizará la herramienta *Cypress* la cual está enfocada en el proceso de *testing*. Esta herramienta se la puede utilizar gracias a la previa implementación de NPM que se está usando para el desarrollo del sistema. Es por lo que únicamente debemos instalar la herramienta para realizar las pruebas sobre el proyecto y ejecutar el comando para comenzar a usar su interfaz como se puede ver en la **Figura 31**.

Figura 31: Interfaz de Cypress



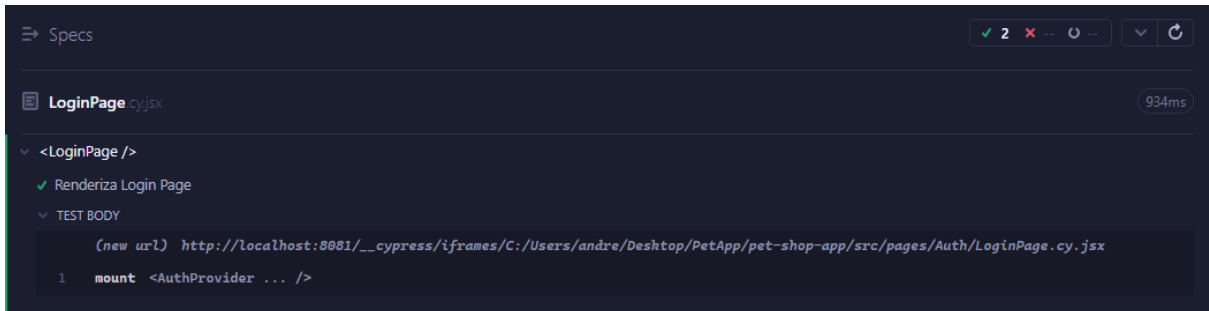
Para hacer uso de esta herramienta se debe seleccionar el proceso de pruebas por componentes, para que se pueda elegir sobre que archivo se desea realizar las pruebas, luego se debe escribir el código correspondiente para lo que se desea probar, como se puede ver en la **Figura 32**, se realizará el renderizado del componente inicial, es decir de la página de inicio de sesión.

Figura 32: Código para realizar la prueba de renderizado de la página de inicio de sesión

```
1 import React from 'react';
2 import LoginPage from './LoginPage';
3 import { BrowserRouter as Router } from 'react-router-dom';
4 import { AuthProvider } from '../contexts/AuthContext';
5
6 describe('<LoginPage />', () => {
7   it('Renderiza Login Page', () => {
8     cy.mount(
9       <AuthProvider>
10         <Router>
11           <LoginPage />
12         </Router>
13       </AuthProvider>
14     );
15   });
16 }
```

Lo que generará que se realice la prueba y verificar si funciona de manera adecuada o existe algún error en la misma, en este caso se realiza la ejecución y se puede verificar que todo funciona de manera correcta como se puede observar en la **Figura 33**.

Figura 33: Prueba de renderización finalizada



En base a esto se realizará diferentes pruebas para verificar el correcto funcionamiento de los componentes o módulos.

Figura 34: Prueba de renderización e inicio de sesión en la página inicial

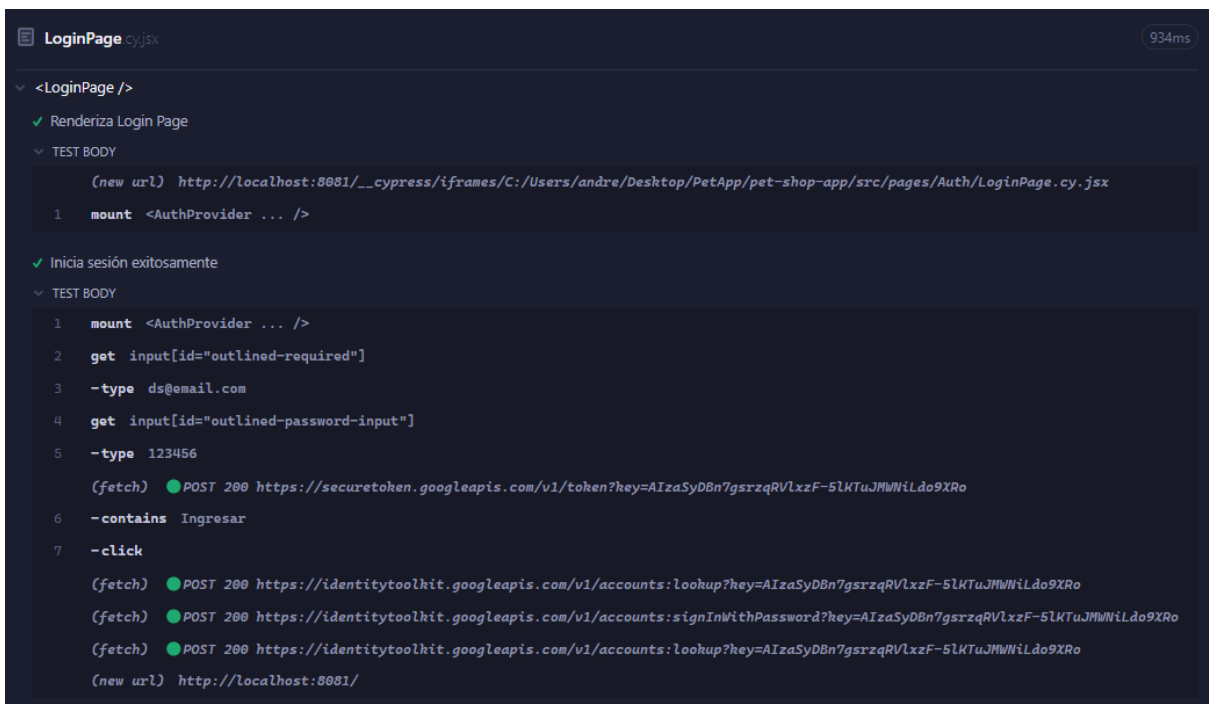


Figura 35: Prueba de ingreso de un nuevo cliente y almacenamiento en la base de datos

```

ClientCreate cy.js
00:02

<ClientShow />
  ✓ renders
  TEST BODY
    (new url) http://localhost:8081/__cypress/iframe/C:/Users/andre/Desktop/PetApp/pet-shop-
    app/src/components/Clientes/ClientCreate.cy.js
  1 mount <AuthProvider ... />
  2 get input[id="name"]
  3 -type John, {force: true}
    (fetch) ● POST 200 https://identitytoolkit.googleapis.com/v1/accounts:lookup?key=AIzaSyDBn7gsrzqRVLxzF-5LkTuJWwHILdo9XRo
  4 get #lastname
  5 -type Doe, {force: true}
  6 get #address
  7 -type 123 Main St, {force: true}
  8 get #phone
  9 -type 123456789, {force: true}
  10 get #phone2
  11 -type 987654321, {force: true}
  12 get #email
  13 -type john.doe@example.com, {force: true}
  14 get form
  15 -submit
    (fetch) ● POST 200 https://firestore.googleapis.com/google.firestore.v1.Firestore/Write/channel?VER=8&database=projects%2Fpet-
    shop-app-61ef3%2Fdatabases%2F(default)&RID=2126&CVER=22&X-HTTP-Session-Id=gsessionid&zx=ftfpisd2r2U&t=1
    (fetch) ● GET 200 https://firestore.googleapis.com/google.firestore.v1.Firestore/Write/channel?gsessionId=KsLsjRrFuMP2BnbyHLUy-
    05z9e_KiOagntP-Hxbz1Zc&VER=8&database=projects%2Fpet-shop-app-
    61ef3%2Fdatabases%2F(default)&RID=rpc&SID=XN_ffjZqI6Smb4YJWSET6A&CI=0&AID=0&TYPE=xmlhttp&zx=9h2h35t0eavb&t=1
    (fetch) ● POST 200 https://firestore.googleapis.com/google.firestore.v1.Firestore/Write/channel?VER=8&database=projects%2Fpet-
    shop-app-61ef3%2Fdatabases%2F(default)&gsessionId=KsLsjRrFuMP2BnbyHLUy-05z9e_KiOagntP-
    Hxbz1Zc&SID=XN_ffjZqI6Smb4YJWSET6A&RID=2127&AID=1&zx=r0zfrnchm81o&t=1
    (new url) http://localhost:8081/clients
  
```

4.2.2 Pruebas de aceptación

Las pruebas de aceptación son realizadas por parte del usuario final, debido a esto son pruebas muy importantes en el proceso del desarrollo debido a que se verifica si cumple o no con los requisitos establecidos en el inicio del proyecto (Ocerin, 2023).

Las pruebas se harán en base a los requerimientos de la **Tabla 4** y un ejemplo de prueba de aceptación se puede observar en la **Tabla 8**

Tabla 8: Prueba de aceptación relacionada con la HU01

PRUEBA DE ACEPTACIÓN			
Id	PA01	HU	HU01
Nombre	Iniciar y cerrar sesión		
Objetivos	Comprobar que el usuario pueda iniciar sesión mediante su correo y contraseña, además de cerrar sesión en el sistema.		
Desarrollo	Abrir el aplicativo web en el navegador, seguido de eso se deberá colocar las		

	credenciales de inicio de sesión del usuario, luego dar clic en ingresar para acceder al sistema. Una vez se deberá hacer clic en el botón de cerrar sesión en la parte superior de la pantalla y se podrá observar que regresó a la página donde se colocan las credenciales para el inicio de sesión.
Resultado	Se puede realizar el inicio de sesión con el correo y contraseña de forma satisfactoria mediante el formulario, al hacer clic en el botón de cierre de sesión en la página de inicio, lleva automáticamente al formulario de inicio de sesión.
Conclusión	La prueba se realizó con éxito

5 CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentará las conclusiones y recomendaciones que se consiguió obtener al final del proceso de implementación del aplicativo web y pruebas correspondientes para comprobar que funcione de manera correcta y cumpla con los requisitos establecidos en un principio.

5.1 Conclusiones

- Las herramientas seleccionadas para el desarrollo en la parte del *front-end* de este proyecto han sido las adecuadas debido a que el uso de un *framework* como *React* que se centra en la construcción de interfaces de usuario, han facilitado la creación y reutilización de componentes dentro de la aplicación, además de la fácil integración de librerías que apoyan en la elaboración de las páginas, con el uso de componentes prediseñados para únicamente modificarlos y usarlos según sea conveniente para el proyecto.
- El manejo de *Firebase* dentro de la aplicación ha sido de una gran ayuda debido a los servicios que se utilizó, como por ejemplo el uso de la base de datos en tiempo real que brinda una sincronía en tiempo real con los datos entre el sistema y la base de datos. Además de eso *Firebase* ha servido como un servicio de almacenamiento en la nube con el servicio de *Storage*, en el cual al momento de subir fotografías se pueden almacenar sin ningún inconveniente y usarlas en el caso que sea necesario.
- El sistema de autenticación dentro del sistema es una parte fundamental y es por eso que gracias al uso de del sistema de *Authentication* que brinda el propio *Firebase*, resulta fácil la gestión de usuarios, ya que nos permite hacer llamado a las funciones ya desarrolladas y que la creación de una interfaz de inicio de sesión sea mucho más sencilla, además de brindar la seguridad que posee al pertenecer a la empresa de *Google*.
- La metodología *Scrum* utilizada en este proyecto ha sido muy útil debido a que se puede establecer las tareas varios ciclos de desarrollo, lo que permite que sea mucho más fácil la organización y distribución de tareas al momento de desarrollar la aplicación. Además de que al hacer uso de esta metodología se logra agilizar el desarrollo y que el tiempo de entrega sea mucho más breve y en este caso es el más adecuada para un proyecto en un tiempo corto.
- La aplicación de pruebas dentro de un proyecto de software es muy importante para conocer si la aplicación funciona de manera correcta, el uso de pruebas unitarias ayuda a conocer si un determinado componente funciona como debería y eso hace que las pruebas sean mucho más fáciles de realizar. Además de las pruebas de aceptación que se verifica que cumpla con el requerimiento que se tenía en un principio y sea satisfactoria para el usuario.

5.2 Recomendaciones

- Al realizar un proyecto en el que se trate de automatizar algún proceso dentro de un negocio o empresa es necesario entender completamente y de forma minuciosa cada aspecto del proceso, para poder realizar de forma óptima el desarrollo y que no existe ningún fallo. Es por eso que definir un proceso para automatizar ayuda mucho en el desarrollo del sistema.
- El *framework* y el resto de las librerías utilizadas en el desarrollo del sistema web suelen actualizarse de forma constante es por eso que es recomendable ir actualizando el sistema con las más nuevas versiones para evitar en ciertos casos que algunas librerías dejen de funcionar de manera correcta.
- Al momento de crear interfaces es recomendable que sean intuitivas y sean vistosas para el usuario debido a que esto hace que sea el sistema mucho más fácil de utilizarlo, esto puede ayudar a que no existan muchos errores y eso puede generar satisfacción para el usuario final.

6 REFERENCIAS BIBLIOGRÁFICAS

- Asto, L. U. (29 de Octubre de 2020). Obtenido de Implementación de un software de historial médico de mascotas para mejorar los procesos de registro y búsqueda en la veterinaria Bandy Pet's distrito de San Martín de Porres.:
<https://repositorio.uch.edu.pe/handle/20.500.12872/509>
- AWS. (s.f.). Obtenido de ¿Qué es JavaScript?: <https://aws.amazon.com/es/what-is/javascript/>
- Cillero, M. (17 de Octubre de 2016). Obtenido de Pruebas:
<https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/pruebas/>
- Conexión Esan. (10 de Octubre de 2018). Obtenido de <https://www.esan.edu.pe/conexion-esan/las-etapas-del-scrum-como-aplicar-este-metodo>:
<https://www.esan.edu.pe/conexion-esan/las-etapas-del-scrum-como-aplicar-este-metodo>
- de Dios, M. Á. (09 de Mayo de 2022). Obtenido de Scrum: qué es y cómo funciona este marco de trabajo: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>
- Durán, M. (30 de Marzo de 2023). Obtenido de Qué es CRUD, cómo funciona y dónde se aplica: <https://blog.hubspot.es/website/que-es-crud>
- EALDE. (s.f.). Obtenido de En qué consiste el Product Backlog y el Sprint Backlog:
<https://www.ealde.es/product-backlog-sprint-backlog/>
- El Comercio. (28 de Agosto de 2022). *El Comercio*. Obtenido de Ecuatorianos gastan cada vez más en sus mascotas: <https://www.elcomercio.com/narices-frias/ecuatorianos-gastan-mas-mascotas-perros-gatos.html>
- El Telégrafo. (29 de Agosto de 2015). *El Telégrafo*. Obtenido de 3 de cada 5 familias tienen una mascota.: <https://www.eltelegrafo.com.ec/noticias/quito/1/3-de-cada-5-familias-tienen-una-mascota>
- Firebase. (s.f.). Obtenido de <https://firebase.google.com/?hl=es>
- Flores, F. (22 de Julio de 2022). Obtenido de Qué es Visual Studio Code y qué ventajas ofrece:
<https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- Gespet. (s.f.). *Gespet*. Obtenido de <https://www.gespet.com/es/index.php>
- Gonzalez, J. (s.f.). Obtenido de Styled Components y Material UI. ¿Qué son y cómo utilizarlas en tu proyecto React?: <https://www.aluracursos.com/blog/styled-components-material-ui-que-son-como-utilizarlas#:~:text=Material%20UI%20es%20una%20biblioteca,de%20dise%C3%B1o%20creado%20por%20Google.>
- Hernandez, M. (8 de Febrero de 2021). *¿Qué es NPM?* Obtenido de <https://www.freecodecamp.org/espanol/news/que-es-npm/>
- ISOTools. (2017). *ISOTools*. Obtenido de Importancia de implementar y automatizar la gestión por procesos: <https://www.isotools.org/2017/06/16/importancia-de-implementar-y-automatizar-la-gestion-por-procesos/>
- Jiménez, J. A. (2020). Obtenido de LA IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO REDUCE LOS TIEMPOS DE ATENCIÓN EN LOS PROCESOS

APLICABLES A LA VENTANILLA ÚNICA DE TURISMO EN LA
MUNICIPALIDAD PROVINCIAL DEL CALLAO.:

<https://www.redalyc.org/journal/816/81665362003/html/>

- Madridiario. (22 de Noviembre de 2022). *Madridiario*. Obtenido de La importancia del software de gestión de negocios: digitalizarse o morir.: <https://www.madridiario.es/la-importancia-del-software-de-gestion-de-negocios--digitalizarse-o-morir>
- Ocerin, A. (24 de Febrero de 2023). Obtenido de Pruebas de aceptación de usuario: qué son y por qué es importante hacerlas: <https://www.wearetesters.com/investigacion-de-mercados/pruebas-aceptacion-usuario-uat-que-son>
- Oracle. (s.f.). *Oracle*. Obtenido de ¿Qué es una base de datos?: <https://www.oracle.com/mx/database/what-is-database/>
- Primicias. (04 de Febrero de 2022). *Primicias*. Obtenido de Nueve de cada 10 pymes en Ecuador invierte en tecnología: <https://www.primicias.ec/noticias/economia/pymes-ecuador-inversion-tecnologia-pandemia/>
- Programacionymas. (s.f.). Obtenido de Scrum: ¿Qué es el Product Backlog?: <https://programacionymas.com/blog/scrum-product-backlog>
- React. (s.f.). Obtenido de <https://es.reactjs.org/>
- Redhat. (s.f.). *Redhat*. Obtenido de ¿Qué es la metodología ágil?: <https://www.redhat.com/es/devops/what-is-agile-methodology>
- SCRUM MEXICO. (2 de Agosto de 2018). Obtenido de Escribiendo Historias de Usuario: <https://scrum.mx/informate/historias-de-usuario>
- Soft, A. (26 de Abril de 2021). Obtenido de Ventajas de las aplicaciones web. Alboka Soft.: <https://www.albokasoft.com/index.php/blog/103-aplicaciones-web>
- Sordo, A. I. (19 de Enero de 2023). Obtenido de Metodología Scrum: qué es, cuáles son sus fases y cómo implementarla: <https://blog.hubspot.es/marketing/metodologia-scrum>
- SUAYED-FCA. (s.f.). *SUAYED-FCA*. Obtenido de http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/4/informatica_4.pdf
- Yeeply. (06 de Septiembre de 2022). Obtenido de ¿Qué son las pruebas unitarias y cómo llevar una a cabo?: <https://www.yeeply.com/blog/que-son-pruebas-unitarias/>

7 ANEXOS

7.1 Historias de Usuario

Tabla 9: HU01 Iniciar y Cerrar sesión

Historia de Usuario	
Identificador: HU01	Usuario: Administrador
Nombre historia: Iniciar y Cerrar Sesión	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Sprint asignado: 1
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema con un correo y contraseña que se encuentre registrado en la base de datos.	
Observaciones: El usuario cierra sesión luego de que haya seleccionado el botón que se encuentra en la pantalla de inicio.	

Tabla 10: HU02 Crear nuevo cliente

Historia de Usuario	
Identificador: HU02	Usuario: Administrador
Nombre historia: Crear nuevo cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y debe llenar el formulario para el registro con los campos de: nombre, apellido, domicilio, teléfono, teléfono de respaldo y correo electrónico.	
Observaciones: El sistema verificará que no existan clientes repetidos. El campo de teléfono de respaldo es opcional.	

Tabla 11: HU03 Editar cliente

Historia de Usuario	
Identificador: HU03	Usuario: Administrador

Nombre historia: Editar cliente	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 1	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y edita los campos en el formulario en caso de ser necesario.	
Observaciones: El sistema verificará que no existan campos en blanco a excepción del teléfono de respaldo el cual es opcional.	

Tabla 12: HU04 Visualizar listado de clientes

Historia de Usuario	
Identificador: HU04	Usuario: Administrador
Nombre historia: Visualizar listado de clientes	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 1	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y puede visualizar el listado de todos los clientes que han sido agregados.	
Observaciones:	

Tabla 13: HU05 Crear nueva mascota

Historia de Usuario	
Identificador: HU05	Usuario: Administrador
Nombre historia: Crear nueva mascota	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y debe llenar el formulario para el registro de la mascota: nombre, especie, raza, año de nacimiento, peso, dueño y alergias.	
Observaciones: Para crear una nueva mascota es necesario tener al menos un cliente para vincularlo.	

Tabla 14: HU06 Editar mascota

Historia de Usuario	
Identificador: HU06	Usuario: Administrador
Nombre historia: Editar mascota	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 1	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y edita los campos que sean requeridos.	
Observaciones: El sistema verificará que no existan campos en blanco.	

Tabla 15: HU07 Visualizar listado de mascotas

Historia de Usuario	
Identificador: HU07	Usuario: Administrador
Nombre historia: Visualizar listado de mascotas	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 1	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema puede observar el listado de todas las mascotas que han sido registradas.	
Observaciones:	

Tabla 16: HU08 Crear nuevo servicio

Historia de Usuario	
Identificador: HU08	Usuario: Administrador
Nombre historia: Crear nuevo servicio	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y llena el formulario correspondiente para el nuevo servicio: nombre	

y descripción del servicio, costo para los diferentes tamaños de mascotas.
Observaciones: El sistema verificará que no existan campos en blanco.

Tabla 17: HU09 Editar servicio

Historia de Usuario	
Identificador: HU09	Usuario: Administrador
Nombre historia: Editar servicio	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 1	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y edita los campos que sean necesarios en caso de que así lo requiera el servicio.	
Observaciones: El sistema verificará que no existan campos en blanco.	

Tabla 18: HU10 Visualizar listado de servicios

Historia de Usuario	
Identificador: HU10	Usuario: Administrador
Nombre historia: Visualizar listado de servicios	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 1	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y puede observar el listado de todos los servicios que han sido creados.	
Observaciones:	

Tabla 19: HU11 Crear nuevo producto

Historia de Usuario	
Identificador: HU11	Usuario: Administrador
Nombre historia: Crear nuevo producto	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta

Días estimados: 2	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y llena los campos del formulario para agregar un nuevo producto: nombre, descripción, costo, imagen y stock.	
Observaciones:	

Tabla 20: HU12 Editar producto

Historia de Usuario	
Identificador: HU12	Usuario: Administrador
Nombre historia: Editar producto	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 1	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y edita los campos que se consideren ser necesarios de modificación.	
Observaciones: El sistema verificará que no existan campos en blanco.	

Tabla 21: HU13 Visualizar listado de productos

Historia de Usuario	
Identificador: HU13	Usuario: Administrador
Nombre historia: Visualizar listado de productos	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 1	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y puede observar el listado de todos los productos que han sido creados.	
Observaciones:	

Tabla 22: HU14 Crear nueva cita

Historia de Usuario	
Identificador: HU14	Usuario: Administrador

Nombre historia: Crear nueva cita	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 3	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y llena los campos correspondientes para agendar una cita: cliente, mascota, tamaño mascota, fecha de cita, servicios, comentarios, productos, valor de servicios, productos y valor total.	
Observaciones: Se debe seleccionar el cliente primero para poder seleccionar la mascota y ambos deben ser creados con anterioridad. Los servicios y productos deben ser creados para poder ser seleccionados, los valores son asignados automáticamente en base a los datos de cada servicio o producto.	

Tabla 23: HU15 Editar cita

Historia de Usuario	
Identificador: HU15	Usuario: Administrador
Nombre historia: Editar cita	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 2	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y edita los campos que sean necesarios en caso de que así lo requiera.	
Observaciones:	

Tabla 24: HU16 Visualizar listado de citas

Historia de Usuario	
Identificador: HU16	Usuario: Administrador
Nombre historia: Visualizar listado de citas	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 1	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción:	

El usuario ingresa al sistema y podrá visualizar el listado de las citas que han sido creadas.
Observaciones:

Tabla 25: HU17 Crear nueva factura

Historia de Usuario	
Identificador: HU17	Usuario: Administrador
Nombre historia: Crear nueva factura	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y luego de haber completado una cita se va a generar una factura con los datos que se creó en la cita.	
Observaciones:	

Tabla 26: HU18 Visualizar factura

Historia de Usuario	
Identificador: HU18	Usuario: Administrador
Nombre historia: Visualizar factura	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 1	Sprint asignado: 2
Programador responsable: Alvear Andrés	
Descripción: El usuario ingresa al sistema y va a poder observar las diferentes facturas que han sido completadas.	
Observaciones:	

Tabla 27: HU19 Recuperar contraseña

Historia de Usuario	
Identificador: HU19	Usuario: Administrador
Nombre historia: Recuperar contraseña	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Días estimados: 1	Sprint asignado: 3
Programador responsable: Alvear Andrés	

Descripción:

El usuario podrá recuperar la contraseña en caso de que la haya olvidado, para esto en el formulario inicial habrá un enlace para que pueda solicitar cambiar la contraseña.

Observaciones:

Para solicitar la nueva contraseña le llegará un enlace a su correo que iniciar sesión.

7.2 Product Backlog

Tabla 28: Product Backlog

PRODUCT BACKLOG				
ID	HISTORIA DE USUARIO	ESTADO	PRIORIDAD	ITERACIÓN
HU01	Iniciar y Cerrar Sesión	Finalizado	Alta	1
HU02	Crear nuevo cliente	Finalizado	Alta	2
HU03	Editar cliente	Finalizado	Media	2
HU04	Visualizar listado de clientes	Finalizado	Media	2
HU05	Crear nueva mascota	Finalizado	Alta	2
HU06	Editar mascota	Finalizado	Media	2
HU07	Visualizar listado de mascotas	Finalizado	Media	2
HU08	Crear nuevo servicio	Finalizado	Alta	2
HU09	Editar servicio	Finalizado	Media	2
HU10	Visualizar listado de servicios	Finalizado	Media	2
HU11	Crear nuevo producto	Finalizado	Alta	2
HU12	Editar producto	Finalizado	Media	2
HU13	Visualizar listado de productos	Finalizado	Media	2
HU14	Crear nueva cita	Finalizado	Alta	2
HU15	Editar cita	Finalizado	Media	2
HU16	Visualizar listado de citas	Finalizado	Media	2
HU17	Crear nueva factura	Finalizado	Alta	2
HU18	Visualizar factura	Finalizado	Media	2
HU19	Recuperar contraseña	Finalizado	Media	3

7.3 Sprint Backlog

Tabla 29: Sprint Backlog

SPRINT BACKLOG				
SB	NOMBRE	FASE	HU	ACTIVIDADES
SB01	Diseño de Interfaces	Diseño	N/A	<ul style="list-style-type: none"> • <i>Mockups</i> de la aplicación web
	Configuración del ambiente de desarrollo	Desarrollo	N/A	<ul style="list-style-type: none"> • Crear proyecto en <i>Firestore</i>, activación de autenticación y creación de la base de datos en <i>Firestore</i>. • Crear proyecto en <i>React JS</i>. • Vincular base de datos con el proyecto.
	Sistema de autenticación	Desarrollo	HU01	<ul style="list-style-type: none"> • Crear formulario de inicio de sesión.

				<ul style="list-style-type: none"> • Redirigir a la página de inicio. • Crear botón de cerrar sesión en la pantalla de inicio.
SB02	Desarrollo de CRUD's	Desarrollo	HU02	<ul style="list-style-type: none"> • Crear formulario para crear un nuevo cliente. • Guardar el registro en la colección Clientes de <i>Firebase</i>. • Redirigir a la pantalla Clientes.
			HU03	<ul style="list-style-type: none"> • Consultar la información de un cliente por su ID desde <i>Firebase</i>. • Presentar la información en un formulario, para poder editarla. • Editar la información del cliente y actualizarla en la colección en <i>Firebase</i>. • Redirigir a la pantalla Clientes.
			HU04	<ul style="list-style-type: none"> • Consultar todos los registros de la colección Clientes. • Presentarlos en una tabla.
			HU05	<ul style="list-style-type: none"> • Crear formulario para crear una nueva mascota. • Guardar el registro en la colección Mascotas de <i>Firebase</i>. • Redirigir a la pantalla Mascotas.
			HU06	<ul style="list-style-type: none"> • Consultar la información de una mascota por su ID desde <i>Firebase</i>. • Presentar la información en un formulario, para poder editarla. • Editar la información de la mascota y actualizarla en la colección en <i>Firebase</i>. • Redirigir a la pantalla Mascotas.
			HU07	<ul style="list-style-type: none"> • Consultar todos los registros de la colección Mascotas. • Presentarlos en una tabla.
			HU08	<ul style="list-style-type: none"> • Crear formulario para crear un nuevo servicio. • Guardar el registro en la colección Servicios de <i>Firebase</i>. • Redirigir a la pantalla Servicios.
			HU09	<ul style="list-style-type: none"> • Consultar la información de un servicio por su ID desde <i>Firebase</i>. • Presentar la información en un formulario, para poder editarla. • Editar la información del servicio y actualizarla en la

				colección en <i>Firebase</i> . <ul style="list-style-type: none"> • Redirigir a la pantalla Servicios.
			HU10	<ul style="list-style-type: none"> • Consultar todos los registros de la colección Servicios. • Presentarlos en una tabla.
			HU11	<ul style="list-style-type: none"> • Crear formulario para crear un nuevo producto. • Guardar el registro en la colección Productos de <i>Firebase</i>. • Redirigir a la pantalla Productos.
			HU12	<ul style="list-style-type: none"> • Consultar la información de un producto por su ID desde <i>Firebase</i>. • Presentar la información en un formulario, para poder editarla. • Editar la información del producto y actualizarla en la colección en <i>Firebase</i>. • Redirigir a la pantalla Productos.
			HU13	<ul style="list-style-type: none"> • Consultar todos los registros de la colección Productos. • Presentarlos en una tabla.
			HU14	<ul style="list-style-type: none"> • Crear formulario para crear una nueva cita. • Guardar el registro en la colección Citas de <i>Firebase</i>. • Redirigir a la pantalla Citas.
			HU15	<ul style="list-style-type: none"> • Consultar la información de una cita por su ID desde <i>Firebase</i>. • Presentar la información en un formulario, para poder editarla. • Editar la información de la cita y actualizarla en la colección en <i>Firebase</i>. • Redirigir a la pantalla Citas.
			HU16	<ul style="list-style-type: none"> • Consultar todos los registros de la colección Citas. • Presentarlos en una tabla.
			HU17	<ul style="list-style-type: none"> • Completar una cita para almacenar un nuevo registro en la colección Facturación de <i>Firebase</i>. • Redirigir a la pantalla Facturación.
			HU18	<ul style="list-style-type: none"> • Consultar todos los registros de la colección Facturación. • Presentarlos en una tabla.
SB03	Recuperación contraseña	Desarrollo	HU19	<ul style="list-style-type: none"> • Crear un campo para solicitar un cambio de clave, ingresando el correo.

				<ul style="list-style-type: none"> • Recibir el correo electrónico con el enlace para poder cambiar la contraseña. • Cambiar la contraseña en el formulario mostrado y actualizarla en <i>Firestore</i>.
	Pruebas	Pruebas	N/A	<ul style="list-style-type: none"> • Realizar pruebas unitarias mediante el uso de <i>Cypress</i>. • Comprobar el funcionamiento de los módulos del sistema. • Realizar pruebas de aceptación. • Comprobar de forma manual el uso de todas las tareas que se establecieron en los requerimientos.

7.4 Pruebas de Aceptación

Tabla 30: PA01 Iniciar y cerrar sesión

PRUEBA DE ACEPTACIÓN			
Id	PA01	HU	HU01
Nombre	Iniciar y cerrar sesión		
Objetivos	Comprobar que el usuario pueda iniciar sesión mediante su correo y contraseña, además de cerrar sesión en el sistema.		
Desarrollo	Abrir el aplicativo web en el navegador, seguido de eso se deberá colocar las credenciales de inicio de sesión del usuario, luego dar clic en ingresar para acceder al sistema. Una vez se deberá hacer clic en el botón de cerrar sesión en la parte superior de la pantalla y se podrá observar que regresó a la página donde se colocan las credenciales para el inicio de sesión.		
Resultado	Se puede realizar el inicio de sesión con el correo y contraseña de forma satisfactoria mediante el formulario, al hacer clic en el botón de cierre de sesión en la página de inicio, lleva automáticamente al formulario de inicio de sesión.		
Conclusión	La prueba se realizó con éxito		

Tabla 31: PA02 Crear nuevo cliente

PRUEBA DE ACEPTACIÓN			
Id	PA02	HU	HU02
Nombre	Crear nuevo cliente		
Objetivos	Comprobar que el usuario pueda agregar un nuevo cliente.		

Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo clientes, luego hacer clic sobre el botón de nuevo cliente. Una vez aparezca el formulario de ingresar un nuevo cliente, llenar los campos correspondientes y hacer clic en crear cliente.
Resultado	El sistema permite crear un nuevo cliente, luego de haber ingresado los datos correspondientes en el formulario, aparecen en el listado de todos los clientes.
Conclusión	La prueba se realizó con éxito

Tabla 32: PA03 Editar cliente

PRUEBA DE ACEPTACIÓN			
Id	PA03	HU	HU03
Nombre	Editar cliente		
Objetivos	Comprobar que el usuario pueda editar un cliente ya creado.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo clientes, luego buscar el cliente a editar, dar clic en el botón editar en la parte derecha del cliente seleccionado. Finalmente editar los campos deseados y dar clic en guardar.		
Resultado	El sistema permite editar un cliente ya creado con anterioridad, luego de que se editaron los campos correspondientes, aparecen en el listado de todos los clientes con los cambios realizados.		
Conclusión	La prueba se realizó con éxito		

Tabla 33: PA04 Visualizar listado de clientes

PRUEBA DE ACEPTACIÓN			
Id	PA04	HU	HU04
Nombre	Visualizar listados de clientes		
Objetivos	Comprobar que el usuario pueda visualizar el listado de clientes.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo clientes, aparecerán enlistados todos los clientes creados.		
Resultado	El sistema permite visualizar todos los clientes creados.		
Conclusión	La prueba se realizó con éxito		

Tabla 34: PA05 Crear nueva mascota

PRUEBA DE ACEPTACIÓN			
-----------------------------	--	--	--

Id	PA05	HU	HU05
Nombre	Crear nueva mascota		
Objetivos	Comprobar que el usuario pueda agregar una nueva mascota.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo mascotas, luego hacer clic sobre el botón de nueva mascota. Una vez aparezca el formulario de ingresar una nueva mascota, llenar los campos correspondientes y hacer clic en crear mascota.		
Resultado	El sistema permite crear una nueva mascota, luego de haber ingresado los datos correspondientes en el formulario, aparecen en el listado de todas las mascotas.		
Conclusión	La prueba se realizó con éxito		

Tabla 35: PA06 Editar mascota

PRUEBA DE ACEPTACIÓN			
Id	PA06	HU	HU06
Nombre	Editar mascota		
Objetivos	Comprobar que el usuario pueda editar una mascota ya creada.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo mascotas, luego buscar la mascota a editar, dar clic en el botón editar en la parte derecha de la mascota seleccionada. Finalmente editar los campos deseados y dar clic en guardar.		
Resultado	El sistema permite editar una mascota ya creada con anterioridad, luego de que se editaron los campos correspondientes, aparecen en el listado de todas las mascotas con los cambios realizados.		
Conclusión	La prueba se realizó con éxito		

Tabla 36: PA07 Visualizar listado de mascotas

PRUEBA DE ACEPTACIÓN			
Id	PA07	HU	HU07
Nombre	Visualizar listado de mascotas		
Objetivos	Comprobar que el usuario pueda visualizar el listado de mascotas.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo mascotas, aparecerán enlistados todas las mascotas creadas.		
Resultado	El sistema permite visualizar todas las mascotas creadas.		

Conclusión	La prueba se realizó con éxito
-------------------	--------------------------------

Tabla 37: PA08 Crear nuevo servicio

PRUEBA DE ACEPTACIÓN			
Id	PA08	HU	HU08
Nombre	Crear nuevo servicio		
Objetivos	Comprobar que el usuario pueda agregar un nuevo servicio.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo servicios, luego hacer clic sobre el botón de nuevo servicio. Una vez aparezca el formulario de ingresar un nuevo servicio, llenar los campos correspondientes y hacer clic en crear servicio.		
Resultado	El sistema permite crear un nuevo servicio, luego de haber ingresado los datos correspondientes en el formulario, aparecen en el listado de todos los servicios.		
Conclusión	La prueba se realizó con éxito		

Tabla 38: PA09 Editar servicio

PRUEBA DE ACEPTACIÓN			
Id	PA09	HU	HU09
Nombre	Editar servicio		
Objetivos	Comprobar que el usuario pueda editar un servicio ya creado.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo servicios, luego buscar el servicio a editar, dar clic en el botón editar en la parte derecha del servicio seleccionado. Finalmente editar los campos deseados y dar clic en guardar.		
Resultado	El sistema permite editar un servicio ya creado con anterioridad, luego de que se editaron los campos correspondientes, aparecen en el listado de todos los servicios con los cambios realizados.		
Conclusión	La prueba se realizó con éxito		

Tabla 39: PA10 Visualizar listado de servicios

PRUEBA DE ACEPTACIÓN			
Id	PA10	HU	HU10
Nombre	Visualizar listado de servicios		

Objetivos	Comprobar que el usuario pueda visualizar el listado de servicios.
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo servicios, aparecerán enlistados todos los servicios creados.
Resultado	El sistema permite visualizar todos los servicios creados.
Conclusión	La prueba se realizó con éxito

Tabla 40: PA11 Crear nuevo producto

PRUEBA DE ACEPTACIÓN			
Id	PA11	HU	HU11
Nombre	Crear nuevo producto		
Objetivos	Comprobar que el usuario pueda agregar un nuevo producto.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo productos, luego hacer clic sobre el botón de nuevo producto. Una vez aparezca el formulario de ingresar un nuevo producto, llenar los campos correspondientes y hacer clic en crear producto.		
Resultado	El sistema permite crear un nuevo producto, luego de haber ingresado los datos correspondientes en el formulario, aparecen en el listado de todos los productos.		
Conclusión	La prueba se realizó con éxito		

Tabla 41: PA12 Editar producto

PRUEBA DE ACEPTACIÓN			
Id	PA12	HU	HU12
Nombre	Editar producto		
Objetivos	Comprobar que el usuario pueda editar un producto ya creado.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo productos, luego buscar el producto a editar, dar clic en el botón editar en la parte derecha del producto seleccionado. Finalmente editar los campos deseados y dar clic en guardar.		
Resultado	El sistema permite editar un producto ya creado con anterioridad, luego de que se editaron los campos correspondientes, aparecen en el listado de todos los productos con los cambios realizados.		
Conclusión	La prueba se realizó con éxito		

Tabla 42: PA13 Visualizar listado de productos

PRUEBA DE ACEPTACIÓN			
Id	PA13	HU	HU13
Nombre	Visualizar listado de productos		
Objetivos	Comprobar que el usuario pueda visualizar el listado de productos.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo productos, aparecerán enlistados todos los productos creados.		
Resultado	El sistema permite visualizar todos los productos creados.		
Conclusión	La prueba se realizó con éxito		

Tabla 43: PA14 Crear nueva cita

PRUEBA DE ACEPTACIÓN			
Id	PA14	HU	HU14
Nombre	Crear nueva cita		
Objetivos	Comprobar que el usuario pueda agregar una nueva cita.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo citas, luego hacer clic sobre el botón de nueva cita. Una vez aparezca el formulario de ingresar una nueva cita, llenar los campos correspondientes y hacer clic en crear cita.		
Resultado	El sistema permite crear una nueva cita, luego de haber ingresado los datos correspondientes en el formulario, aparecen en el listado de todas las citas.		
Conclusión	La prueba se realizó con éxito		

Tabla 44: PA15 Editar cita

PRUEBA DE ACEPTACIÓN			
Id	PA15	HU	HU15
Nombre	Editar cita		
Objetivos	Comprobar que el usuario pueda editar una cita ya creado.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo citas, luego buscar la cita a editar, dar clic en el botón editar en la parte derecha de la cita seleccionada. Finalmente editar los campos deseados y dar clic en guardar.		
Resultado	El sistema permite editar una cita ya creada con anterioridad, luego de que se editaron los campos correspondientes, aparecen en el listado de todas las citas con los cambios realizados.		

Conclusión	La prueba se realizó con éxito
-------------------	--------------------------------

Tabla 45: PA16 Visualizar listado de citas

PRUEBA DE ACEPTACIÓN			
Id	PA16	HU	HU16
Nombre	Visualizar listado de citas		
Objetivos	Comprobar que el usuario pueda visualizar el listado de citas.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo citas, aparecerán enlistados todas las citas creadas.		
Resultado	El sistema permite visualizar todas las citas creadas.		
Conclusión	La prueba se realizó con éxito		

Tabla 46: PA17 Crear nueva factura

PRUEBA DE ACEPTACIÓN			
Id	PA17	HU	HU17
Nombre	Crear nueva factura		
Objetivos	Comprobar que el usuario pueda crear una nueva factura.		
Desarrollo	Se debe ir al menú lateral de la aplicación y seleccionar el módulo de facturas, se deberá escoger la cita correspondiente y al momento de completar la cita se va a generar la factura correspondiente.		
Resultado	El sistema permite crear una nueva factura luego de haber completado una cita de forma satisfactoria.		
Conclusión	La prueba se realizó con éxito		

Tabla 47: PA18 Visualizar factura

PRUEBA DE ACEPTACIÓN			
Id	PA18	HU	HU18
Nombre	Visualizar factura		
Objetivos	Comprobar que el usuario pueda visualizar las facturas.		
Desarrollo	Ir al menú lateral de la aplicación y seleccionar el módulo facturas, luego seleccionar el botón de ver facturas y se tendrá el listado.		
Resultado	El sistema permite visualizar el listado de las facturas.		
Conclusión	La prueba se realizó con éxito		

Tabla 48: PA19 Recuperar contraseña

PRUEBA DE ACEPTACIÓN			
Id	PA19	HU	HU19
Nombre	Recuperar contraseña		
Objetivos	Comprobar que el usuario pueda actualizar su contraseña.		
Desarrollo	Ingresar al sistema, en el formulario inicial aparece una opción de contraseña olvidada, seleccionar la opción para que aparezca un campo para ingresar el correo. Luego llegará el correo con el enlace correspondiente y se accederá para cambiar la contraseña.		
Resultado	El sistema envía el correo para realizar el cambio de contraseña y se pueda acceder al sistema.		
Conclusión	La prueba se realizó con éxito		