

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

SISTEMAS DE INFORMACIÓN



**DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS DE INFORMACIÓN**

**DESARROLLO DE UN APLICATIVO WEB PARA LA VENTA DE MASCOTAS DE UN
CRIADERO DE ANIMALES**

Autor: David Alejandro Rueda Vallejo

Director: Ing. Oswaldo Espinosa

QUITO DM, DICIEMBRE 2022

DEDICATORIA

Este trabajo de titulación está dedicado a mis hermanos como una muestra de que podemos lograr los objetivos que nos propongamos por más complicados y largos que puedan ser; y aunque a momentos pensemos que tal vez tomamos el camino equivocado, hay que recordar que no existe una sola forma de llegar a la cima, somos lo suficientemente inteligentes para encontrar el modo de lograrlo.

AGRADECIMIENTOS

Agradezco a mi tutor de titulación por ayudarme en todo el proceso de planificación y desarrollo de este trabajo; y a mis amigos de carrera, que siempre estuvieron para ayudarme con temas que no manejaba con tanta facilidad.

RESUMEN

El presente trabajo de titulación pretende brindar una herramienta digital a los negocios relacionados con la cría y venta de animales para mejorar sus servicios a la comunidad en épocas de pandemia y desolación. Se describirán las razones del desarrollo de este proyecto y se hablará sobre la metodología y herramientas escogidas para desarrollar el aplicativo de la mejor manera posible y se describirá todo el proceso realizado, desde la planificación de funcionalidades, casos de uso y modelado de la base de datos, hasta el desarrollo como tal del aplicativo con la preparación del entorno de trabajo, el diseño de la interfaz gráfica, la implementación de la base de datos y la generación de las funcionalidades. Finalmente se brindarán conclusiones sobre el trabajo realizado y sus herramientas, y se describirán recomendaciones a implementar en el futuro del aplicativo.

ÍNDICE

Tabla de Contenidos

DEDICATORIA.....	II
AGRADECIMIENTOS.....	III
RESUMEN.....	IV
ÍNDICE.....	V
Tabla de Contenidos.....	V
Índice de Ilustraciones.....	VII
Índice de Tablas.....	10
1. CAPÍTULO I: INTRODUCCIÓN.....	11
1.1. Marco Teórico.....	12
1.2. Marco Conceptual.....	13
2. CAPÍTULO II: MODELADO DEL APLICATIVO WEB.....	14
2.1. Identificación de Funcionalidades.....	14
2.2. Casos de Uso.....	15
2.2.1. Nivel General.....	15
2.2.2. Nivel de Gestión.....	15
2.2.3. Nivel a Detalle.....	17
2.3. Modelado de la Base de Datos.....	25
2.3.1. Colecciones (Objetos) de la Base de Datos.....	26
3. CAPÍTULO III: DISEÑO Y DESARROLLO DEL APLICATIVO WEB... 27	
3.1. Preparación del Entorno de Trabajo.....	27
3.1.1. Instalación de Visual Studio Code.....	27
3.1.2. Creación del Proyecto Angular.....	27
3.2. Diseño de la Interfaz Gráfica.....	29
3.2.1. Interfaces de los Componentes.....	29
3.3. Implementación de la Base de Datos.....	54
3.3.1. Requisitos para la conexión con Firebase.....	54
3.3.2. Incorporación de los servicios de Firebase al aplicativo.....	59
3.4. Implementación de las Funcionalidades.....	64
4. CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES.....	66

4.1.	Conclusiones.....	66
4.2.	Recomendaciones	67
	BIBLIOGRAFÍA	69

Índice de Ilustraciones

Ilustración 1 – Casos de Uso Generales	15
Ilustración 2 – Casos de Uso: Gestión de Usuarios.....	15
Ilustración 3 – Casos de Uso: Gestión de Mascotas.....	15
Ilustración 4 – Casos de Uso: Gestión de Publicaciones.....	16
Ilustración 5 – Casos de Uso: Gestión de Comentarios.....	16
Ilustración 6 – Casos de Uso: Gestión del Carrito de compras	16
Ilustración 7 – Casos de Uso: Ingreso de Usuario.....	17
Ilustración 8 – Casos de Uso: Consulta de Usuario	18
Ilustración 9 – Casos de Uso: Ingreso de Mascota.....	19
Ilustración 10 – Casos de Uso: Eliminación de Mascota	20
Ilustración 11 – Casos de Uso: Consulta de Mascotas	20
Ilustración 12 – Casos de Uso: Ingreso de Publicación.....	21
Ilustración 13 – Casos de Uso: Eliminación de Publicación.....	21
Ilustración 14 – Casos de Uso: Consulta de Publicaciones	22
Ilustración 15 – Casos de Uso: Ingreso de Comentario.....	22
Ilustración 16 – Casos de Uso: Consulta de Comentarios.....	23
Ilustración 17 – Casos de Uso: Ingreso de Mascota al Carrito de Compras	24
Ilustración 18 – Casos de Uso: Eliminación de Mascota del Carrito de Compras ..	24
Ilustración 19 – Casos de Uso: Consulta del Carrito de Compras	25
Ilustración 20 Ilustración - Creación Proyecto Angular: Seleccionar Carpeta de Trabajo.....	27
Ilustración 21 - Creación Proyecto Angular: Abrir Nueva Terminal	28
Ilustración 22 - Creación Proyecto Angular: Terminal	28
Ilustración 23 - Creación Proyecto Angular: Generación del Proyecto Angular	28
Ilustración 24 - Interfaz Gráfica: NavBar Vista de Computador.....	30
Ilustración 25 - Interfaz Gráfica: NavBar Vista de Computador con Scroll.....	30
Ilustración 26 - Interfaz Gráfica: NavBar Vista de Smartphone.....	30
Ilustración 27 - Interfaz Gráfica: NavBar Vista de Smartphone Menú Desplegable	30
Ilustración 28 - Interfaz Gráfica: Footer Vista de Computador.....	31
Ilustración 29 - Interfaz Gráfica: Footer Vista de Smartphone	32
Ilustración 30 - Interfaz Gráfica: Registro Vista de Computador	33
Ilustración 31 - Interfaz Gráfica: Registro Vista de Smartphone	34

Ilustración 32 - Interfaz Gráfica: Login Vista de Computador	35
Ilustración 33 - Interfaz Gráfica: Login Vista de Smartphone	36
Ilustración 34 - Interfaz Gráfica: Home Vista de Computador	37
Ilustración 35 - Interfaz Gráfica: Home Vista de Smartphone	37
Ilustración 36 - Interfaz Gráfica: Home: Carrusel Hipervínculos Vista de Computador	38
Ilustración 37 - Interfaz Gráfica: Home: Carrusel Hipervínculos Vista de Smartphone	38
Ilustración 38 - Interfaz Gráfica: Home: Carrusel Publicaciones Vista de Computador	39
Ilustración 39 - Interfaz Gráfica: Home: Carrusel Publicaciones Vista de Smartphone	39
Ilustración 40 - Interfaz Gráfica: Pets Vista de Computador	40
Ilustración 41 - Interfaz Gráfica: Pets Vista de Smartphone	41
Ilustración 42 - Interfaz Gráfica: About us Vista de Computador	42
Ilustración 43 - Interfaz Gráfica: About us Vista de Smartphone	43
Ilustración 44 - Interfaz Gráfica: About us: Our team Vista de Computador	44
Ilustración 45 - Interfaz Gráfica: About us: Our Team Vista de Smartphone.....	45
Ilustración 46 - Interfaz Gráfica: Reviews Vista de Computador	46
Ilustración 47 - Interfaz Gráfica: About us Vista de Smartphone	46
Ilustración 48 - Interfaz Gráfica: Cart Vista de Computador	47
Ilustración 49 - Interfaz Gráfica: Cart Vista de Smartphone.....	48
Ilustración 50 - Interfaz Gráfica: Profile Vista de Computador	49
Ilustración 51 - Interfaz Gráfica: Profile Vista de Smartphone.....	50
Ilustración 52 - Interfaz Gráfica: Manage Pets: New Pet vista de Computador.....	51
Ilustración 53 - Interfaz Gráfica: Manage Pets: New Pet vista de Smartphone	51
Ilustración 54 - Interfaz Gráfica: Manage Pets: Pet List vista de Computador.....	52
Ilustración 55 - Manage Pets: Pet List vista de Smartphone	52
Ilustración 56 - Interfaz Gráfica: Manage Posts: New Post vista de Computador ..	53
Ilustración 57 - Interfaz Gráfica: Manage Posts: New Post vista de Smartphone...	53
Ilustración 58 - Interfaz Gráfica: Manage Posts: Post List vista de Computador ...	53
Ilustración 59 - Interfaz Gráfica: Manage Posts: Post List vista de Smartphone	54
Ilustración 60 - Nuevo Repositorio en Github 1.....	54

Ilustración 61 - Nuevo Repositorio en Github 2.....	55
Ilustración 62 - Exportar Avances en Github	56
Ilustración 63 - Página Principal de Firebase	57
Ilustración 64 - Agregar Nuevo Proyecto Firebase	57
Ilustración 65 - Archivo environment.ts.....	59
Ilustración 66 – Imports Firebase en el Archivo app.module.ts.....	59
Ilustración 67 - Estructura de la Interfaz User.....	60
Ilustración 68 - Estructura del Servicio User	61
Ilustración 69 - Mostrar Usuarios de la Base de Datos: Archivo Html.....	62
Ilustración 70 - Mostrar Usuarios de la Base de Datos: Llamada al Servicio User	62
Ilustración 71 - Mostrar Usuarios de la Base de Datos: Función Get del Servicio User	63
Ilustración 72 - Mostrar Usuarios de la Base de Datos: Interfaz User	63
Ilustración 73 - Mostrar Usuarios de la Base de Datos: Vista Final	64

Índice de Tablas

Tabla 1 - Tabla de Funcionalidades.....	64
---	----

1. CAPÍTULO I: INTRODUCCIÓN

Un mundo que todo el tiempo se encuentra en estado de evolución tecnológica obliga a sus habitantes a acostumbrarse a sus cambios para poder sobrevivir. Con los últimos avances en el internet y las redes sociales, todos los negocios, incluido el caso de los criaderos de animales, han tenido que adaptarse al uso de estos para poder seguir existiendo en el mercado.

A diferencia de años pasados, si alguien requiere de algún producto o servicio, lo puede buscar en internet o mediante redes sociales; esto, con el fin de facilitar la comunicación con el negocio y comprobar la calidad de sus productos, servicios y fiabilidad. Es por este motivo que surge la necesidad de desarrollar un aplicativo web que ayude a los criaderos de animales a generar presencia en internet y redes sociales para poder mejorar sus ventas y garantizar que sus servicios sean de calidad.

Desde su mera existencia, el humano ha sido un ser social que requiere de compañía para vivir a plenitud; si bien es cierto que las sociedades están compuestas por personas que conviven entre ellas para cumplir con esta necesidad, también existen otros seres dignos de compañía: las mascotas. A pesar de tener un intelecto considerado inferior al del humano, muchos animales, especialmente los domésticos, han demostrado ser capaces de generar emociones en las personas y así llegar a convertirse en seres importantes para estas.

En los últimos años, con la aparición del Covid-19, la gente tuvo que aislarse de la sociedad; en los casos de las familias no hubo mayor inconveniente, pero para aquellas personas que viven solas, esto representó una limitación total de su contacto con otros seres. Debido a esto, muchas personas solitarias optaron por la opción de adoptar mascotas que las ayudaran a sobrellevar la soledad.

Para adquirir algún producto o servicio, los procesos de compra y venta se convirtieron en virtuales; el problema de esto fue que, al ser negocios con productos vivientes, los clientes debían acudir presencialmente para interactuar con los animales, cosa que dejó de ser posible con la pandemia. Es esto por lo que nace la idea de desarrollar un aplicativo web con presencia en redes sociales que permita a sus usuarios conocer a las macotas mediante fotos, videos y descripciones, y poder concluir con el proceso de compra de estos.

El alcance del proyecto es el desarrollo de un aplicativo web responsive para la venta de mascotas de un criadero de animales, haciendo uso de la metodología SCRUM. El aplicativo está pensado en tener dos tipos de acceso, uno para clientes y otro para administradores; la única diferencia será que este último contará con la función de gestionar productos y realizar publicaciones.

Se tiene pensado que el aplicativo sea de acceso general, sin necesidad de una cuenta para poder visualizar su contenido; únicamente si se desea realizar una compra, los clientes deberán crear una cuenta con sus datos personales, por lo tanto, el aplicativo contará con una función de registro e ingreso al sistema.

1.1. Marco Teórico

La (Unidad de Bienestar Animal - Ministerio de Agricultura, 2022) de Guatemala define a un criadero como cualquier lugar que tenga animales de compañía para reproducción y con fines de lucro.

Según (Orraiz, 2020), un editor de código es una poderosa herramienta diseñada para desempeñar la tarea específica de escribir y modificar un código web.

(Edix, 2022) indica que un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software.

Front-End hace referencia a todo lo que ven las personas, esto incluye el diseño y ciertos lenguajes (Souza, 2021).

Back-End es la capa de entrada o código de programa al que los usuarios no tienen acceso (Souza, 2021).

(Equipo Editorial Etecé, 2021) define a una base de datos como un conjunto de información perteneciente a un mismo contexto, ordenada de modo sistemático para su posterior recuperación, análisis y/o transmisión.

Los aplicativos o aplicaciones web son programas que funcionan en internet. Es decir, que los datos o los archivos en los que trabajas son procesados y almacenados dentro

de la web. Estas aplicaciones, por lo general, no necesitan ser instaladas en tu computador (GCF Global, s.f.).

(Gonzales, 2019) se refiere a un diseño web responsive como un diseño capaz de adaptarse a pantallas de diferentes tamaños con un solo sitio web. El sistema detecta automáticamente el ancho de la pantalla y a partir de ahí adapta todos los elementos de la página, desde el tamaño de letra hasta las imágenes y los menús, para ofrecer al usuario la mejor experiencia posible.

Una página web es un documento digital multimedia que funciona bajo los estándares de la World Wide Web y a la que se puede acceder mediante un navegador web con conexión activa al internet (Equipo Editorial Etecé, s.f.).

1.2. Marco Conceptual

El sitio oficial de (Visual Studio, 2019) describe a Visual Studio Code como un poderoso editor de código disponible para Windows, macOS y Linux. Contiene soporte y documentación integrada para JavaScript, TypeScript y Node.js y dispone de un ecosistema de gran variedad de lenguajes y extensiones como C++, C#, Java, Python, PHP, y .NET).

Angular es una plataforma de desarrollo generada en TypeScript que contiene un framework basado en componentes para el desarrollo de aplicaciones web escalables. También posee una amplia colección de librerías integradas para diversos usos como son routing, herramientas de administración, comunicación cliente-servidor y más. Por último, tiene una gran variedad de herramientas de desarrollo, compilación, testeo y actualización de código (Angular, 2022).

Bootstrap es un conjunto de herramientas para facilitar el desarrollo FrontEnd (Bootstrap, s.f.). Bootstrap nos permite trabajar con plantillas responsive para no empezar el desarrollo desde cero.

Firebase es una plataforma respaldada por Google para el desarrollo de aplicaciones (Google, s.f.). Esta plataforma servirá para generar la base de datos del aplicativo y lanzarlo a la web.

La metodología Scrum permite abordar proyectos complejos desarrollados en entornos dinámicos y cambiantes de un modo flexible. Está basada en entregas parciales y regulares del producto final en base al valor que ofrecen los clientes (Hurtado, 2021).

2. CAPÍTULO II: MODELADO DEL APLICATIVO WEB

2.1. Identificación de Funcionalidades

A continuación, se describen las funcionalidades principales con las que contará el aplicativo:

F1: El aplicativo tendrá dos tipos de usuarios: clientes y administradores.

F2: El aplicativo debe permitir a los usuarios registrarse e iniciar sesión en el aplicativo.

F3: El aplicativo debe permitir a los usuarios ver la información de los productos sin necesidad de iniciar sesión, pero para poder ocupar las funciones de la página, si será necesario iniciar sesión.

F4: El aplicativo debe permitir a los usuarios administradores la gestión de productos y publicaciones.

F5: El aplicativo debe mostrar a los usuarios todos los productos disponibles y publicaciones en su interfaz.

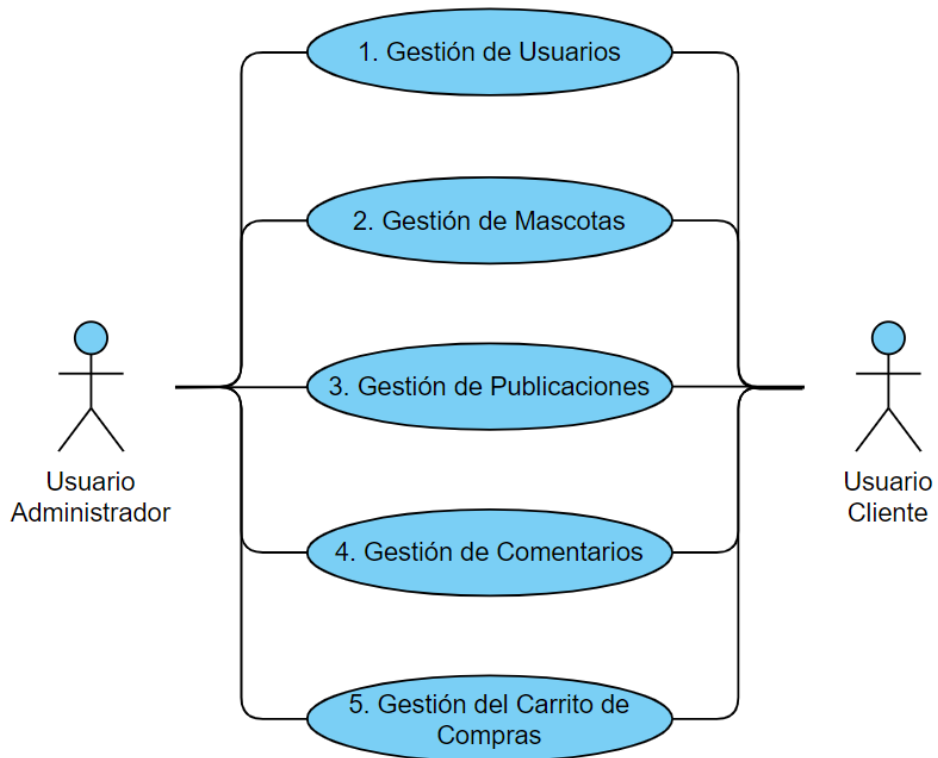
F6: El aplicativo debe permitir a los usuarios clientes añadir y eliminar productos de su carrito de compras.

F7: El aplicativo debe permitir a los usuarios realizar comentarios sobre el servicio brindado a través de la página.

2.2. Casos de Uso

2.2.1. Nivel General

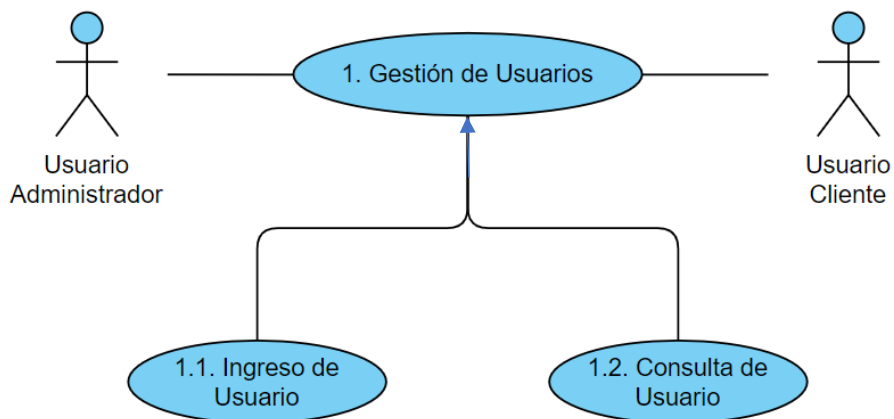
Ilustración 1 – Casos de Uso Generales



2.2.2. Nivel de Gestión

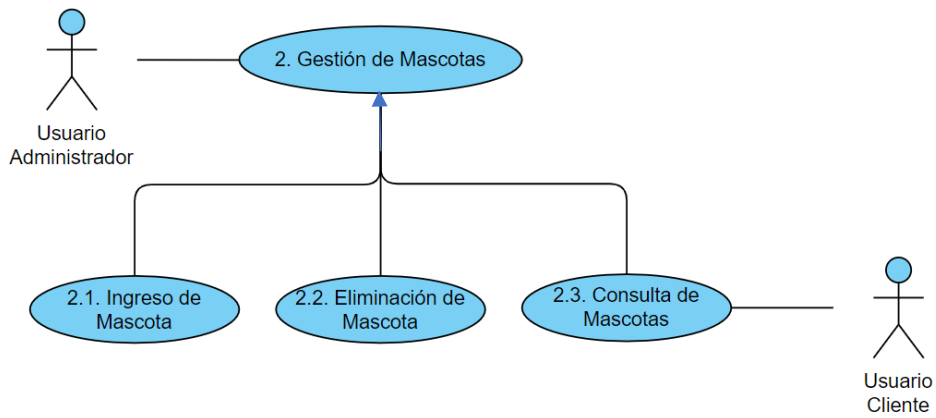
2.2.2.1. Gestión de Usuarios

Ilustración 2 – Casos de Uso: Gestión de Usuarios



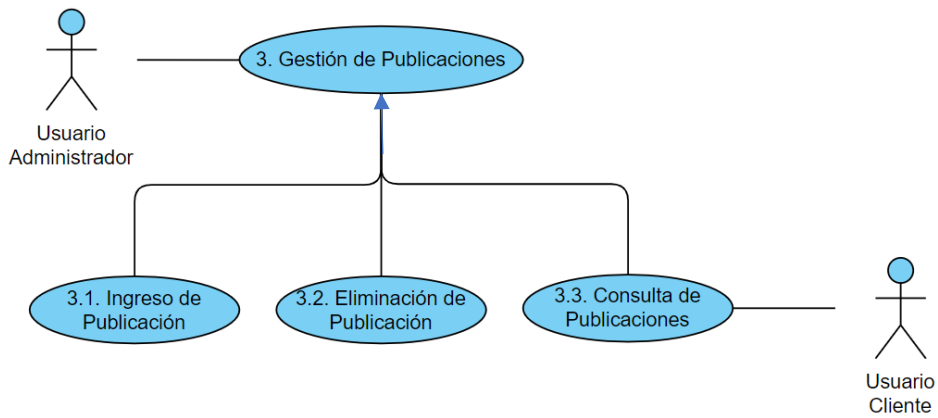
2.2.2.2. Gestión de Mascotas

Ilustración 3 – Casos de Uso: Gestión de Mascotas



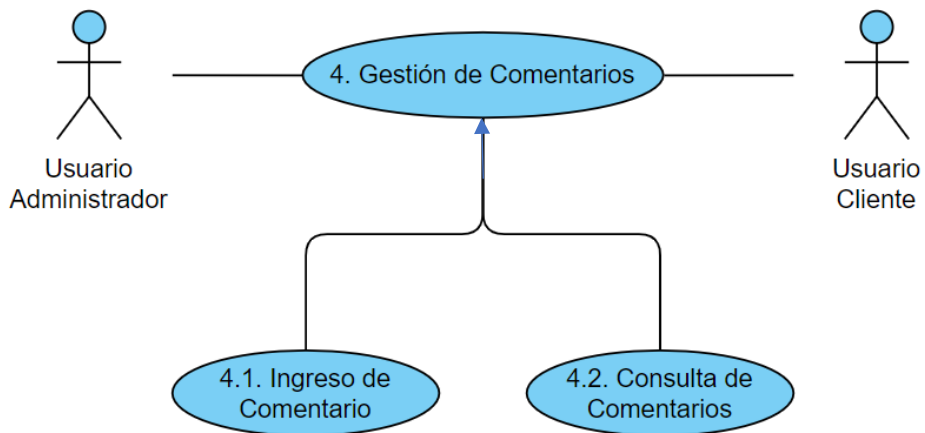
2.2.2.3. Gestión de Publicaciones

Ilustración 4 – Casos de Uso: Gestión de Publicaciones



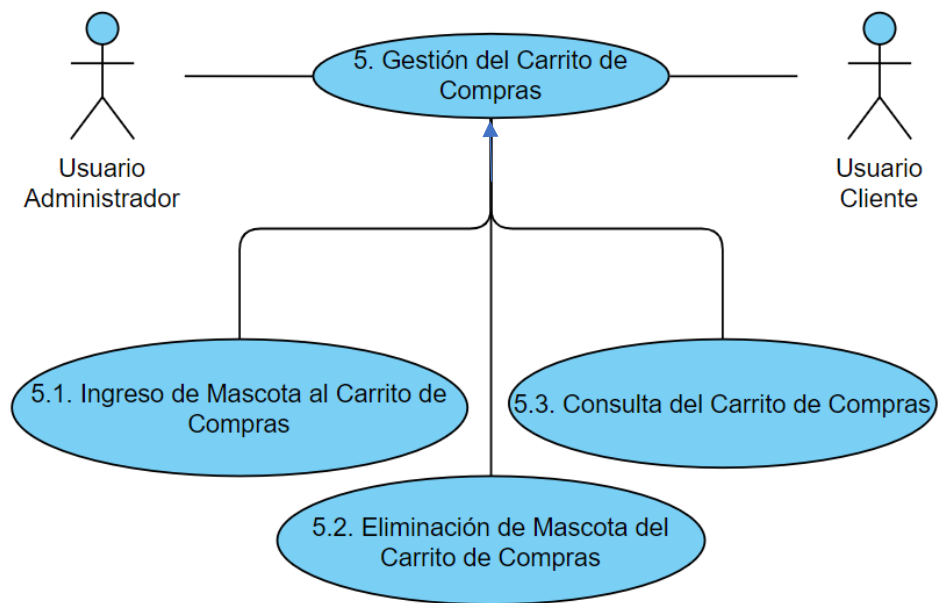
2.2.2.4. Gestión de Comentarios

Ilustración 5 – Casos de Uso: Gestión de Comentarios



2.2.2.5. Gestión del Carrito de Compras

Ilustración 6 – Casos de Uso: Gestión del Carrito de compras

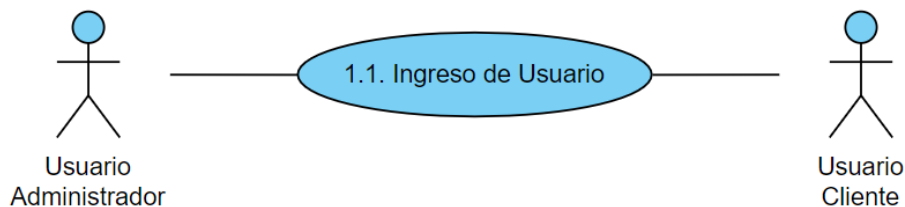


2.2.3. Nivel a Detalle

2.2.3.1. Gestión de Usuarios

2.2.3.1.1. Ingreso de Usuario

Ilustración 7 – Casos de Uso: Ingreso de Usuario



Actores: Usuario Administrador, Usuario Cliente.

Flujo Principal:

1. El actor ingresa al aplicativo y selecciona la pestaña “Profile/Login”. (FA1)
2. El sistema despliega una ventana de ingreso al sistema.
3. El actor selecciona la opción “Register” ubicada en la parte inferior de la ventana.
4. El sistema despliega una ventana de registro de usuario donde solicita la información personal del actor.

5. El actor ingresa la información solicitada y presiona el botón “Register”. (Para registrarse como usuario administrador, el actor debe ingresar con un email con una terminación específica.)

6. El sistema ingresa el usuario nuevo a la base de datos. (E1) (E2) (E3) (E4)

Flujo Alternativo:

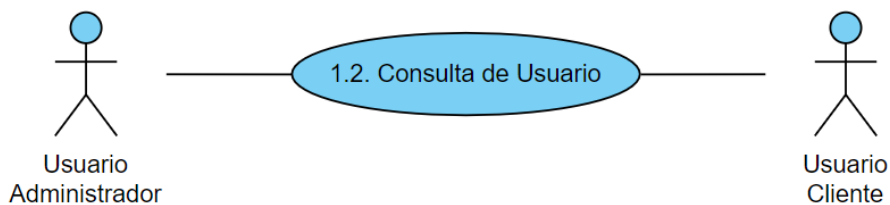
- FA1: Si el actor intenta acceder al carrito de compras, o realizar un comentario, o añadir una mascota al carrito de compras, el sistema redirecciona el aplicativo a la ventana de “Profile/Login”. Continuar con Flujo Principal: Literal 2.

Posibles errores:

- E1: Existen campos sin llenar del formulario.
- E2: El formato del email ingresado es incorrecto.
- E3: La contraseña ingresada tiene menos de 6 caracteres.
- E4: El email ingresado ya está vinculado a otra cuenta.

2.2.3.1.2. Consulta de Usuario

Ilustración 8 – Casos de Uso: Consulta de Usuario



Actores: Usuario Administrador, Usuario Cliente.

Flujo Principal:

1. El actor ingresa al aplicativo y selecciona la pestaña “Profile/Login”. (FA1)
2. El sistema despliega una ventana de ingreso al sistema donde solicita el email y contraseña para ingresar.
3. El actor ingresa la información solicitada y presiona el botón “Login”. (E1) (E2)
4. El sistema inicia sesión en la cuenta del usuario y lo redirecciona a la pestaña de “Profile/Login” donde puede visualizar su información personal.

Flujo Alternativo:

- FA1: Si el actor intenta acceder al carrito de compras, o realizar un comentario, o añadir una mascota al carrito de compras, el sistema redirecciona el aplicativo a la ventana de “Profile/Login”. Continuar con Flujo Principal: Literal 2.

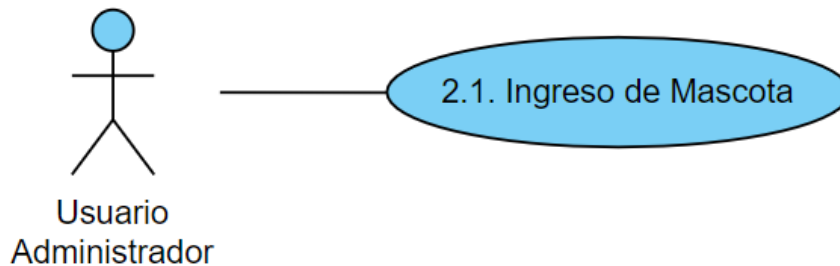
Posibles errores:

- E1: Existen campos sin llenar del formulario.
- E2: Email o contraseña incorrectos.

2.2.3.2. Gestión de Mascotas

2.2.3.2.1. Ingreso de Mascota

Ilustración 9 – Casos de Uso: Ingreso de Mascota



Actores: Usuario Administrador.

Flujo Principal:

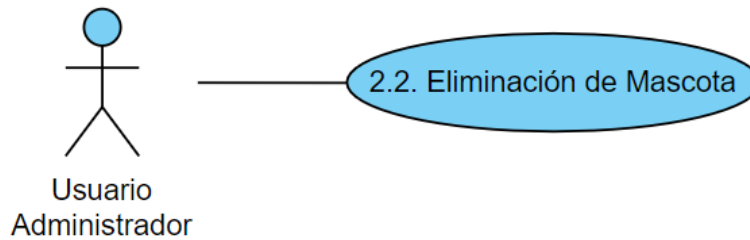
1. El actor ingresa al aplicativo e inicia sesión.
2. El actor selecciona la opción “Manage Pets” ubicada en el footer de la página.
3. El sistema despliega una ventana de gestión de mascotas con una sección para ingresar mascotas al inicio.
4. El actor ingresa la información solicitada de la mascota a ingresar y presiona el botón “Submit”.
5. El sistema ingresa la mascota en la base de datos. (E1)

Posibles errores:

- E1: Existen campos sin llenar del formulario.

2.2.3.2.2. Eliminación de Mascota

Ilustración 10 – Casos de Uso: Eliminación de Mascota



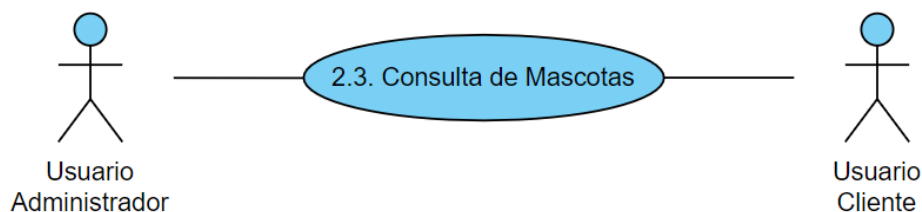
Actores: Usuario Administrador.

Flujo Principal:

1. El actor ingresa al aplicativo e inicia sesión.
2. El actor selecciona la opción “Manage Pets” ubicada en el footer de la página.
3. El sistema despliega una ventana de gestión de mascotas con una sección para ingresar mascotas al inicio y una sección donde se enlistan las mascotas del sistema.
4. El actor se dirige a la mascota que desea eliminar y presiona el botón “Delete”.
5. El sistema elimina la mascota seleccionada de la base de datos.

2.2.3.2.3. Consulta de Mascotas

Ilustración 11 – Casos de Uso: Consulta de Mascotas



Actores: Usuario Administrador, Usuario Cliente

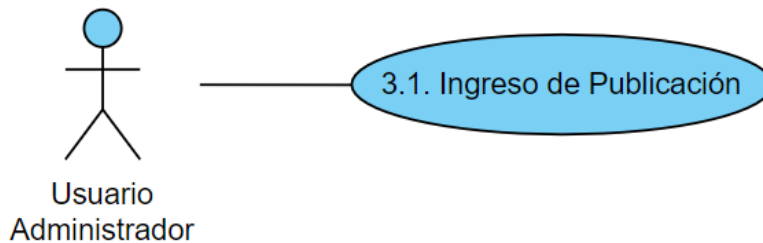
Flujo Principal:

1. El actor ingresa al aplicativo, no es necesario iniciar sesión.
2. El actor selecciona la pestaña “Pets” en la página.
3. El sistema despliega una ventana mostrando todas las mascotas disponibles para comprar.

2.2.3.3. Gestión de Publicaciones

2.2.3.3.1. Ingreso de Publicación

Ilustración 12 – Casos de Uso: Ingreso de Publicación



Actores: Usuario Administrador.

Flujo Principal:

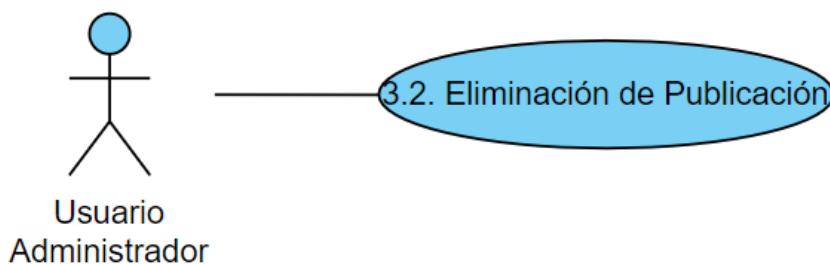
1. El actor ingresa al aplicativo e inicia sesión.
2. El actor selecciona la opción “Manage Posts” ubicada en el footer de la página.
3. El sistema despliega una ventana de gestión de publicaciones con una sección para ingresar publicaciones nuevas al inicio.
4. El actor ingresa la información solicitada de la publicación a ingresar y presiona el botón “Submit”.
5. El sistema ingresa la publicación en la base de datos. (E1)

Posibles errores:

- E1: Existen campos sin llenar del formulario.

2.2.3.3.2. Eliminación de Publicación

Ilustración 13 – Casos de Uso: Eliminación de Publicación



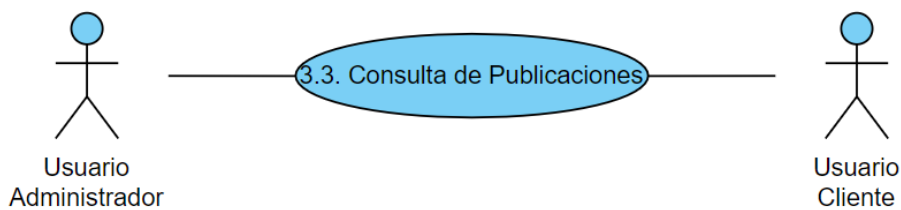
Actores: Usuario Administrador.

Flujo Principal:

1. El actor ingresa al aplicativo e inicia sesión.
2. El actor selecciona la opción “Manage Posts” ubicada en el footer de la página.
3. El sistema despliega una ventana de gestión de publicaciones con una sección para ingresar publicaciones nuevas al inicio y una sección donde se enlistan las publicaciones del sistema.
4. El actor se dirige a la publicación que desea eliminar y presiona el botón “Delete”.
5. El sistema elimina la publicación seleccionada de la base de datos.

2.2.3.3.3. Consulta de Publicaciones

Ilustración 14 – Casos de Uso: Consulta de Publicaciones



Actores: Usuario Administrador, Usuario Cliente

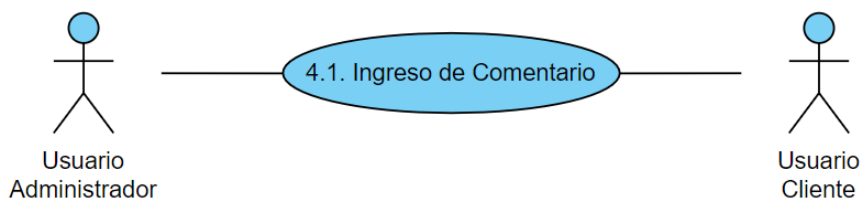
Flujo Principal:

1. El actor ingresa al aplicativo, no es necesario iniciar sesión.
2. Por defecto, el sistema inicia en el home, que es donde se encuentran las publicaciones para ser visualizadas.

2.2.3.4. Gestión de Comentarios

2.2.3.4.1. Ingreso de Comentario

Ilustración 15 – Casos de Uso: Ingreso de Comentario



Actores: Usuario Administrador, Usuario Cliente

Flujo Principal:

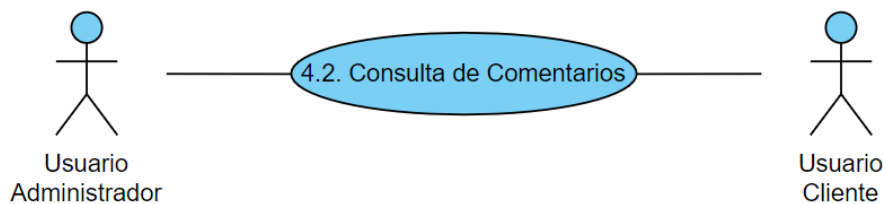
1. El actor ingresa al aplicativo e inicia sesión.
2. El actor selecciona la pestaña “Reviews” ubicada en la barra de navegación de la página.
3. El sistema despliega una ventana de comentarios o reviews del sitio.
4. El actor ingresa el comentario deseado y pulsa el botón “Add Note”.
5. El sistema ingresa el comentario en la base de datos. (E1)

Posibles errores:

- E1: El comentario ingresado está vacío.

2.2.3.4.2. Consulta de Comentarios

Ilustración 16 – Casos de Uso: Consulta de Comentarios



Actores: Usuario Administrador, Usuario Cliente

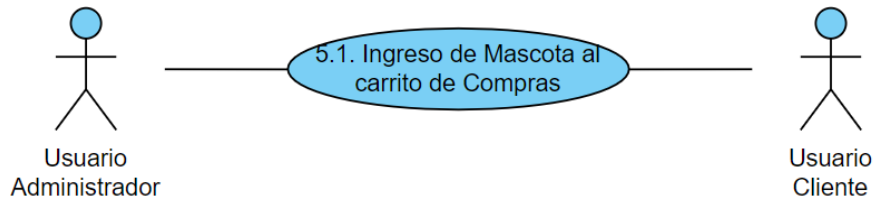
Flujo Principal:

1. El actor ingresa al aplicativo e inicia sesión.
2. El actor selecciona la pestaña “Reviews” ubicada en la barra de navegación de la página.
3. El sistema despliega una ventana de comentarios o reviews del sitio.
4. El actor puede visualizar todos los comentarios realizados con su respectivo nombre de usuario.

2.2.3.5. Gestión del Carrito de Compras

2.2.3.5.1. Ingreso de Mascota al Carrito de Compras

Ilustración 17 – Casos de Uso: Ingreso de Mascota al Carrito de Compras



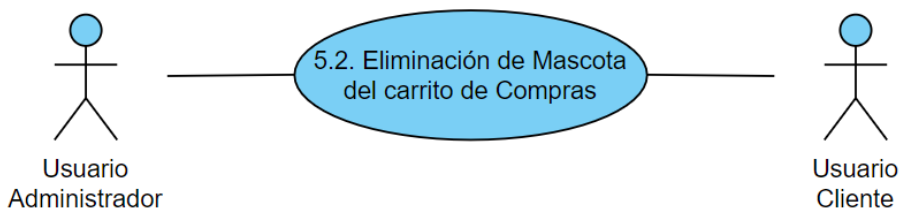
Actores: Usuario Administrador, Usuario Cliente

Flujo Principal:

1. El actor ingresa al aplicativo e inicia sesión.
2. El actor selecciona la pestaña “Pets” en la página.
3. El sistema despliega una ventana mostrando todas las mascotas disponibles para comprar.
4. El actor selecciona la mascota que desea agregar al carrito de compras.
5. El sistema añade la mascota al carrito de compras del actor en la base de datos.

2.2.3.5.2. Eliminación de Mascota del Carrito de Compras

Ilustración 18 – Casos de Uso: Eliminación de Mascota del Carrito de Compras



Actores: Usuario Administrador, Usuario Cliente

Flujo Principal:

1. El actor ingresa al aplicativo e inicia sesión.
2. El actor selecciona la pestaña “Shopping Cart” en la página.

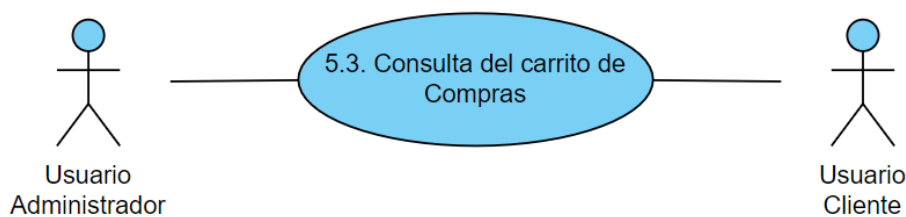
3. El sistema despliega una ventana mostrando el carrito de compras del actor con sus respectivas mascotas añadidas.

5. El actor selecciona la mascota que desea eliminar del carrito de compras y pulsa el botón “Delete”.

6. El sistema elimina la mascota del carrito de compras del autor.

2.2.3.5.3. Consulta del Carrito de Compras

Ilustración 19 – Casos de Uso: Consulta del Carrito de Compras



Actores: Usuario Administrador, Usuario Cliente

Flujo Principal:

1. El actor ingresa al aplicativo e inicia sesión.
2. El actor selecciona la pestaña “Shopping Cart” en la página.
3. El sistema despliega una ventana donde el actor puede ver su carrito de compras con sus respectivas mascotas añadidas.

2.3. Modelado de la Base de Datos

Es importante aclarar que el modelo de base de datos escogido para el aplicativo es no relacional de firebase por lo que no se realizarán los respectivos niveles de estructura de base de datos como en un modelo estructural sino solamente los atributos de los objetos a utilizar. Firebase trabaja con colecciones que representan los objetos, dentro de estas están los documentos que representan a cada registro, y dentro de estos registros se encuentran los campos, que representan a los atributos del registro.

2.3.1. Colecciones (Objetos) de la Base de Datos

2.3.1.1. Pets

- id: string (identificador)
- name: string (nombre)
- race: string (raza de animal)
- price: number (precio)
- image: string (imagen)

2.3.1.2. Users

- id: string (identificador)
- name: string (nombre)
- email: string (correo electrónico)
- password: string (contraseña)
- country: string (país)
- image: string (imagen)
- phone: number (número de teléfono)
- Cart: Colección de Mascotas

2.3.1.3. Posts

- id: string (identificador)
- description: string (contenido)
- title: string (título)

2.3.1.4. Reviews

- id: string (identificador)
- content: string (comentario/review)
- userName: string (nombre de usuario que ingresó el comentario)

3. CAPÍTULO III: DISEÑO Y DESARROLLO DEL APLICATIVO WEB

3.1. Preparación del Entorno de Trabajo

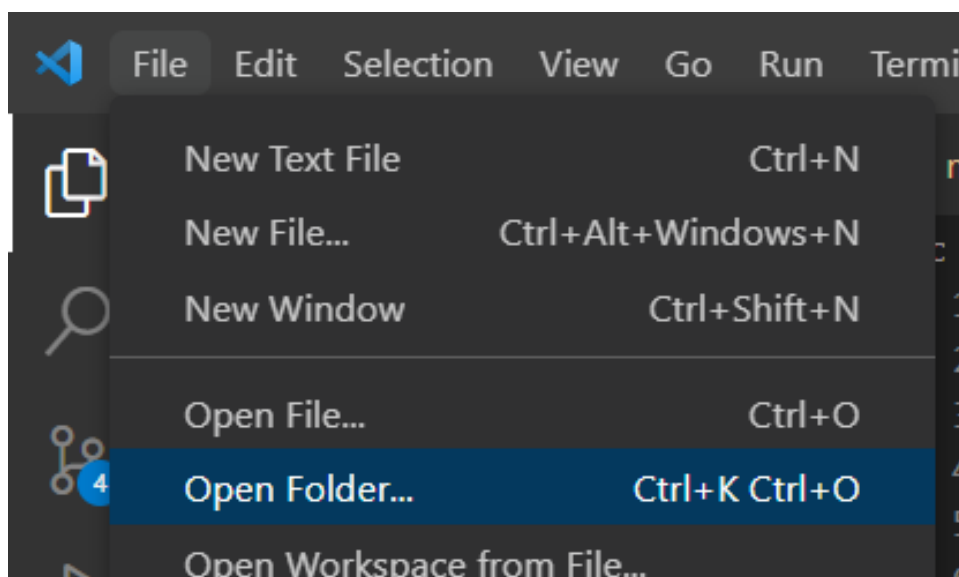
3.1.1. Instalación de Visual Studio Code

El editor de código fuente que se utilizó para realizar el aplicativo propuesto es Visual Studio Code. Basta con descargar la herramienta del sitio web oficial en <https://code.visualstudio.com/> y ejecutarla para poderla usar.

3.1.2. Creación del Proyecto Angular

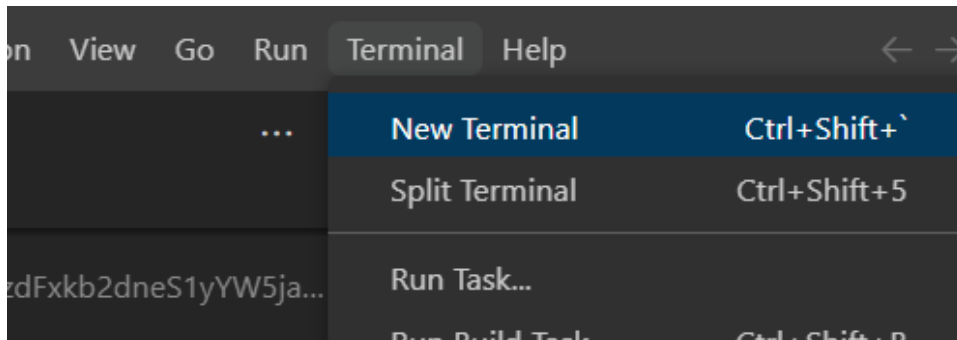
Una vez descargada la herramienta, la abrimos y nos dirigimos a la parte superior izquierda, en la pestaña “File” seleccionamos la opción “Open Folder”. Esta será la carpeta donde se creará el proyecto.

Ilustración 20 Ilustración - Creación Proyecto Angular: Seleccionar Carpeta de Trabajo



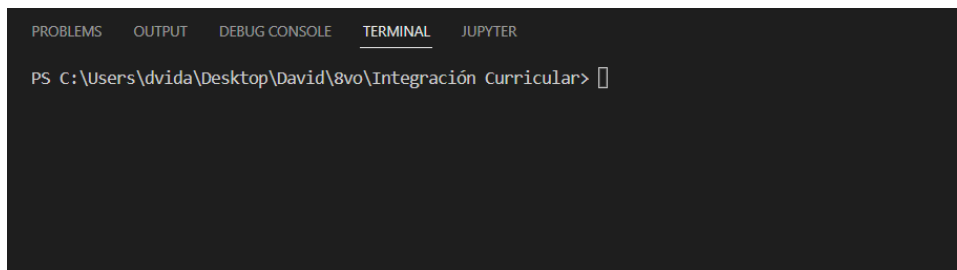
Una vez abierta la carpeta donde se va a trabajar, nos dirigimos a la pestaña “Terminal” y seleccionamos la opción “New Terminal”.

Ilustración 21 - Creación Proyecto Angular: Abrir Nueva Terminal



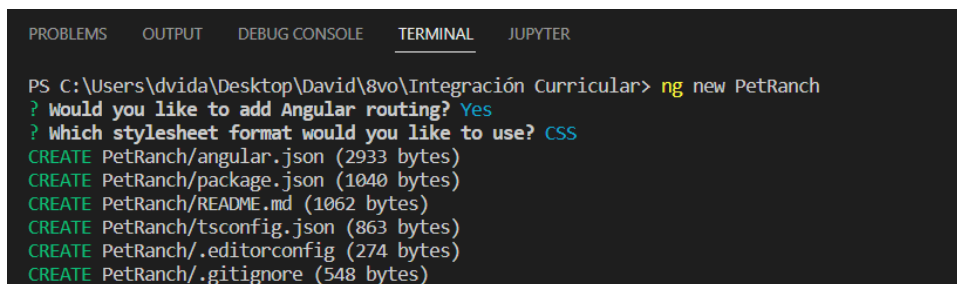
Esto abrirá una terminal en la parte inferior de la herramienta.

Ilustración 22 - Creación Proyecto Angular: Terminal



Para crear un proyecto angular nuevo, basta con escribir el comando “ng nombreProyecto” y dar enter. El terminal nos preguntará si deseamos agregar Routing al proyecto, seleccionamos YES, y luego nos preguntará el formato de la hoja de estilos que usaremos, seleccionamos la opción CSS. Luego esperamos unos minutos a que el proyecto termine de generarse.

Ilustración 23 - Creación Proyecto Angular: Generación del Proyecto Angular



Luego, dentro del proyecto creado, instalamos las siguientes herramientas de angular y bootstrap mediante comandos:

```
npm install -g @angular/cli
```

```
ng add @angular/material
```

```
npm install Bootstrap
```

```
npm install Font awesome
```

```
npm install @ng-bootstrap
```

```
npm install @ngx-bootstrap
```

```
npm install bootstrap-icons
```

```
npm install sweetalert2
```

3.2. Diseño de la Interfaz Gráfica

Antes de pasar a la implementación del diseño de la interfaz gráfica, es importante mencionar que, para poder visualizar el proyecto en un navegador, se debe ejecutar el comando `ng serve` en la terminal. Este comando ejecutará el proyecto en el localhost del navegador y se podrán visualizar cambios en tiempo real con los cambios realizados en el proyecto en Visual Studio Code.

Para las interfaces y funcionamiento del proyecto, angular trabaja con componentes. Cada uno de los componentes generados está compuesto por 4 archivos: un archivo `.css` que sirve para las stylesheets, un archivo `.html` que sirve para el diseño de la interfaz gráfica, un archivo `.specs.ts` con ciertos comandos necesarios para que funcione el componente (este archivo nunca se lo edita), y un archivo `.ts`, que es escrito en lenguaje typescript y sirve para el funcionamiento de las distintas partes del componente.

El aplicativo debe ser responsive, así que los archivos `.html` y `.css` de los componentes funcionan con Bootstrap.

Finalmente, hay que mencionar que existen imágenes sacadas de internet en varias partes del aplicativo. Estas imágenes son solo temporales hasta adaptar el aplicativo a algún cliente.

3.2.1. Interfaces de los Componentes

A continuación se presentan las interfaces del aplicativo tanto en computador como para smartphones.

3.2.1.1. NavBar

El NavBar es una barra con pestañas que aparece en la parte superior de todas las demás interfaces del aplicativo. Sirve para moverse por la página y esta vinculada a un

pequeño código en javascript que escucha un evento de scroll y cambia su fondo y detecta cuando el ancho de la página es muy pequeño y esconde los elementos en un menú desplegable.

Ilustración 24 - Interfaz Gráfica: NavBar Vista de Computador



Ilustración 25 - Interfaz Gráfica: NavBar Vista de Computador con Scroll

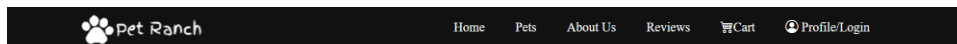


Ilustración 26 - Interfaz Gráfica: NavBar Vista de Smartphone

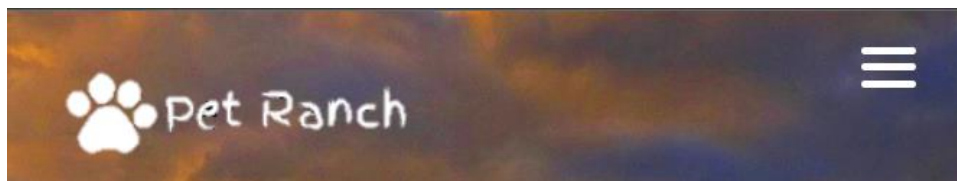
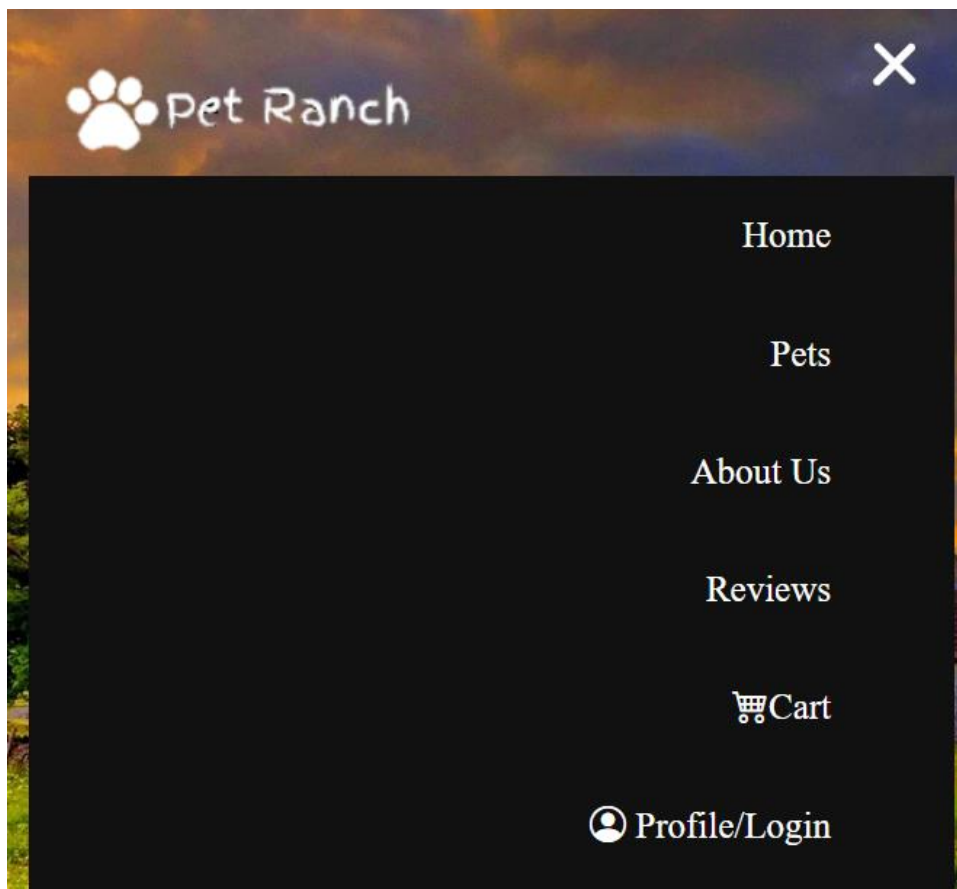


Ilustración 27 - Interfaz Gráfica: NavBar Vista de Smartphone Menú Desplegable



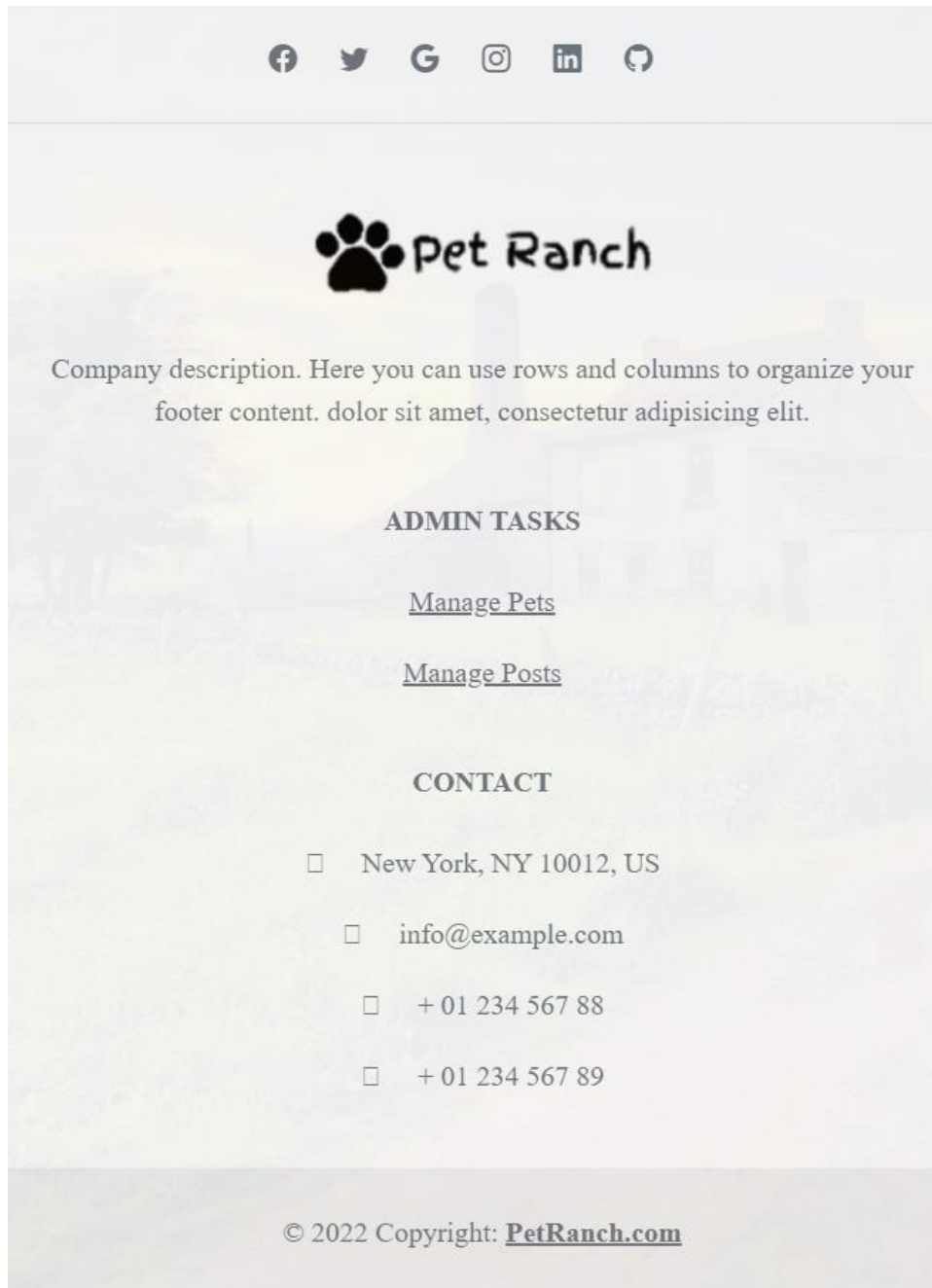
3.2.1.2. Footer

El footer se encuentra en la parte inferior de todas las interfaces del aplicativo. Contiene información acerca del negocio, y también acceso a las funcionalidades administrativas.

Ilustración 28 - Interfaz Gráfica: Footer Vista de Computador



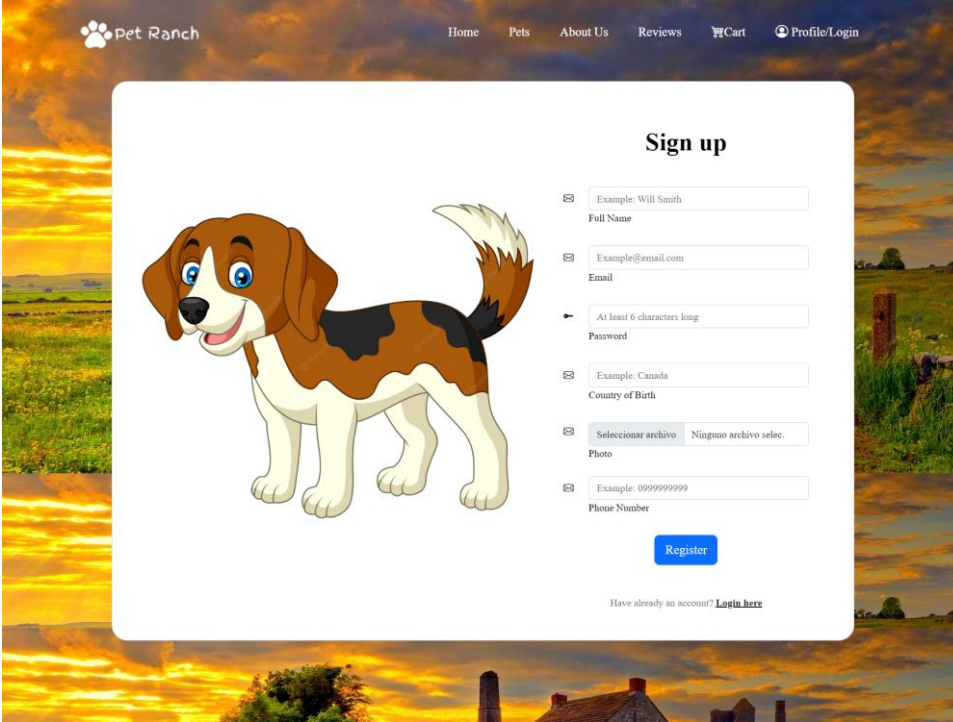
Ilustración 29 - Interfaz Gráfica: Footer Vista de Smartphone



3.2.1.3. Register

La ventana de registro sirve para generar un perfil nuevo en el aplicativo.

Ilustración 30 - Interfaz Gráfica: Registro Vista de Computador


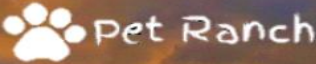


The image shows a web registration form for 'Pet Ranch' set against a background of a sunset over a field. The form is titled 'Sign up' and includes a cartoon illustration of a brown and white dog on the left. The registration fields on the right are:

- Full Name:** Input field with placeholder 'Example: Will Smith'.
- Email:** Input field with placeholder 'Example@email.com'.
- Password:** Input field with a note 'At least 6 characters long'.
- Country of Birth:** Input field with placeholder 'Example: Canada'.
- Photo:** A file selection area with the text 'Seleccionar archivo' and 'Ninguno archivo selec.'.
- Phone Number:** Input field with placeholder 'Example: 0999999999'.

At the bottom of the form is a blue 'Register' button and a link: 'Have already an account? [Login here](#)'.

Ilustración 31 - Interfaz Gráfica: Registro Vista de Smartphone



Sign up

✖

Full Name

✖

Email

🔑

Password

✖

Country of Birth

✖


Photo

✖

Phone Number

Register

Have already an account? [Login here](#)



3.2.1.4. Login

El login es la ventana para iniciar sesión en el aplicativo.

Ilustración 32 - Interfaz Gráfica: Login Vista de Computador

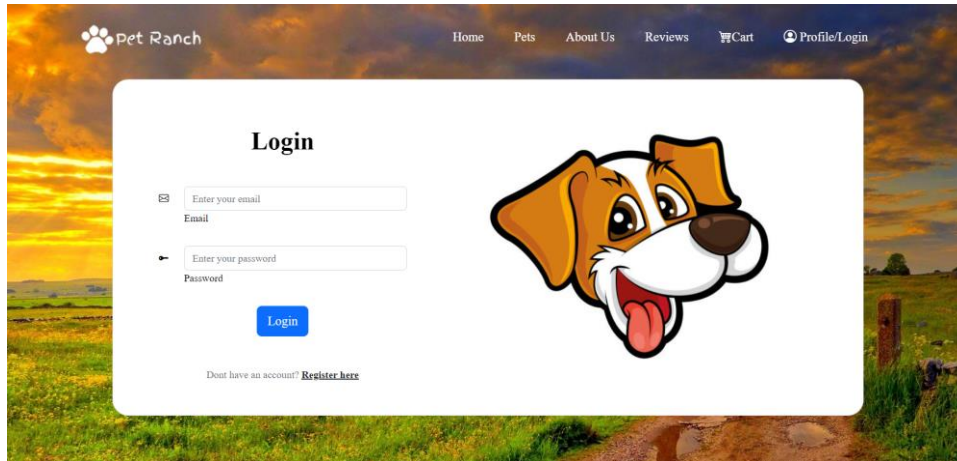
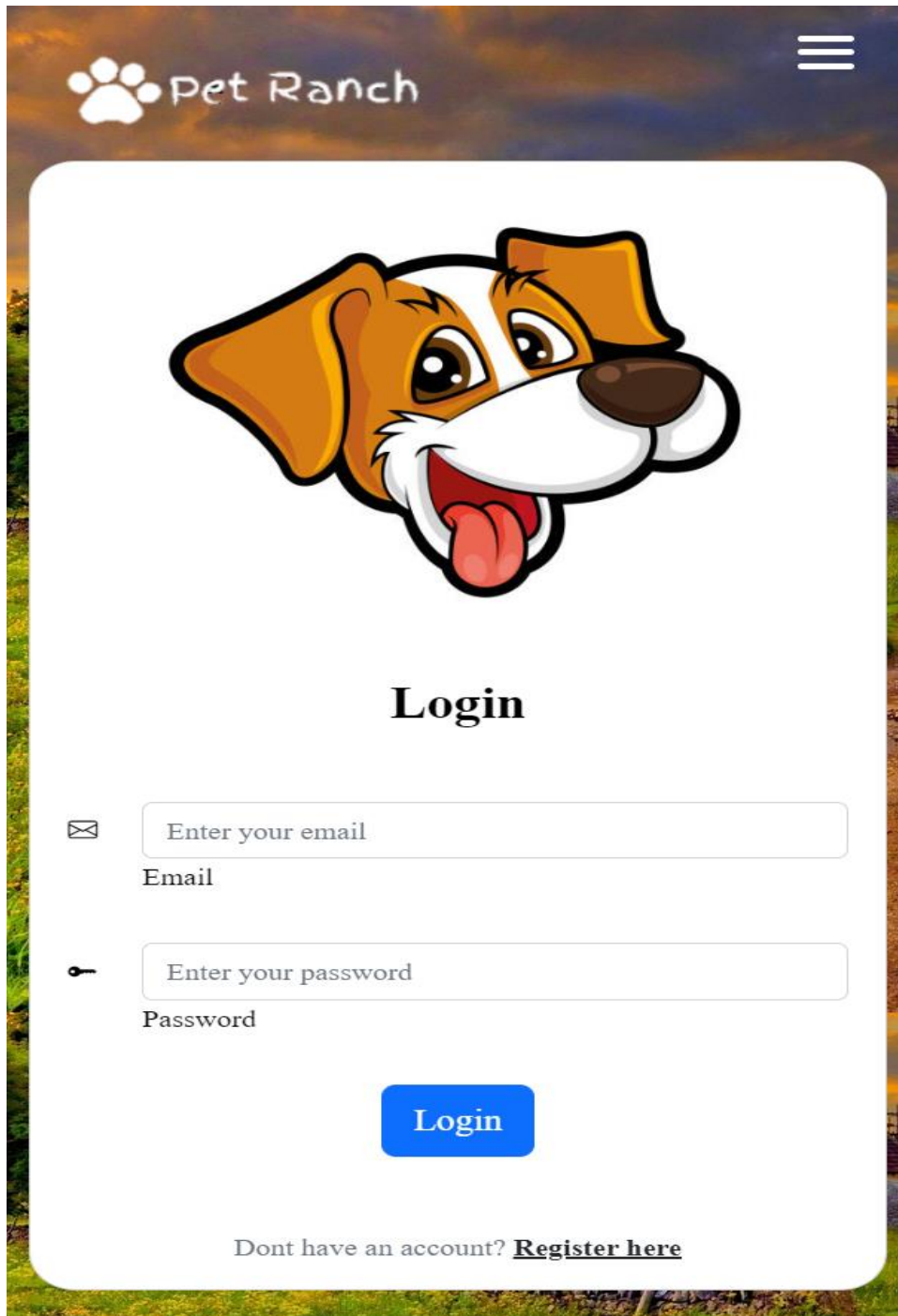


Ilustración 33 - Interfaz Gráfica: Login Vista de Smartphone



3.2.1.5. Home

El Home es la página de inicio del aplicativo, es adonde se redirecciona cuando se ingresa a la página. Cuenta con una imagen que se mueve a la par con el scroll, y está compuesta por dos componentes adicionales: un carrusel con links hacia la interfaz de Pets, About Us y Reviews, y otro carrusel destinado a imprimir publicaciones y noticias del aplicativo.

Ilustración 34 - Interfaz Gráfica: Home Vista de Computador



Ilustración 35 - Interfaz Gráfica: Home Vista de Smartphone



Ilustración 36 - Interfaz Gráfica: Home: Carrusel Hipervínculos Vista de Computador

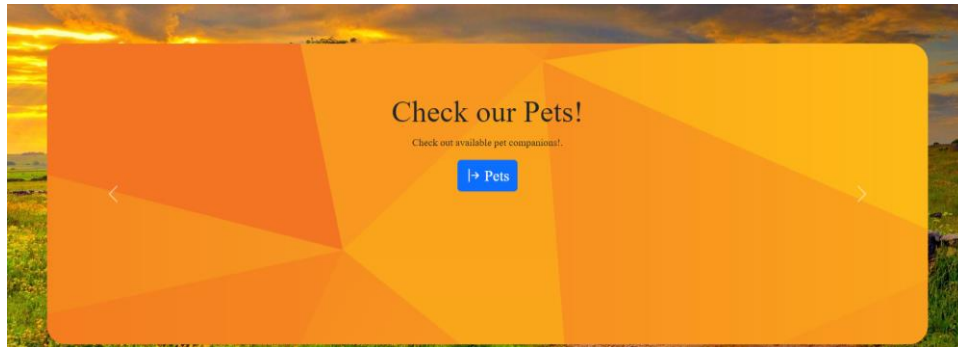


Ilustración 37 - Interfaz Gráfica: Home: Carrusel Hipervínculos Vista de Smartphone

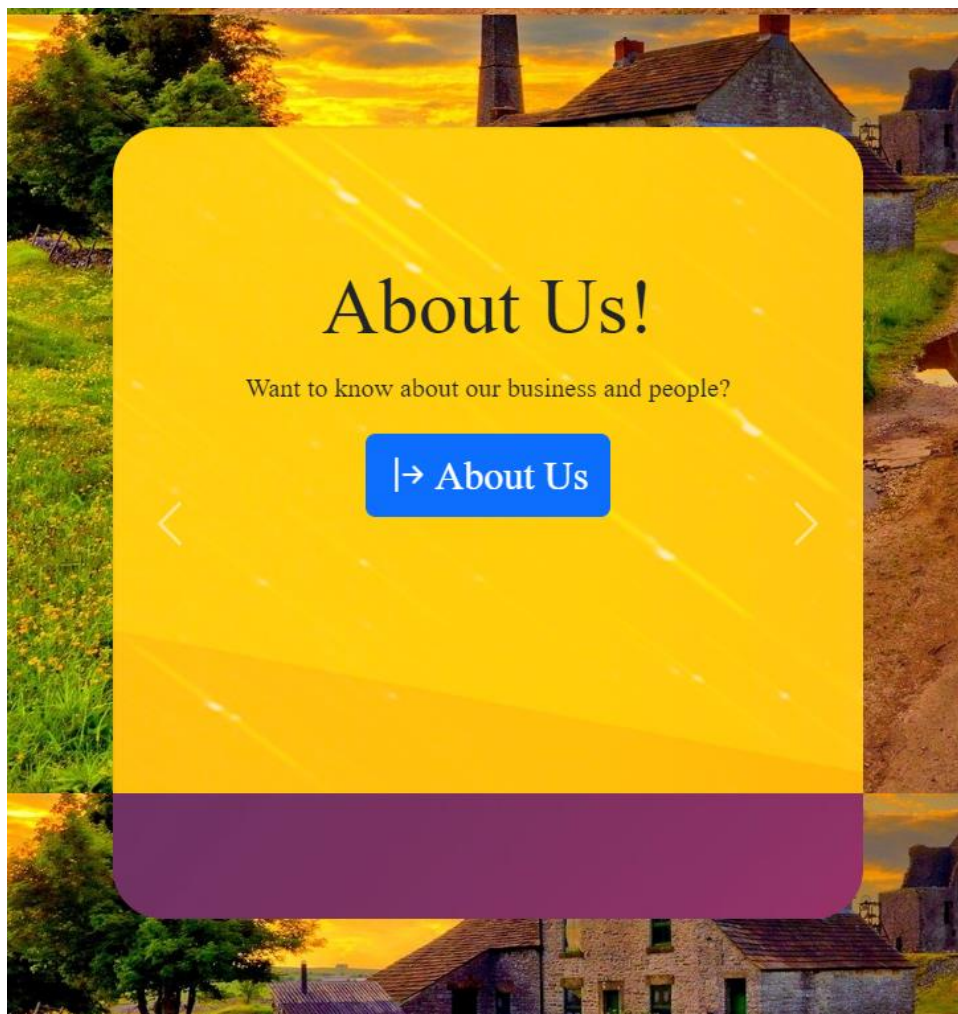
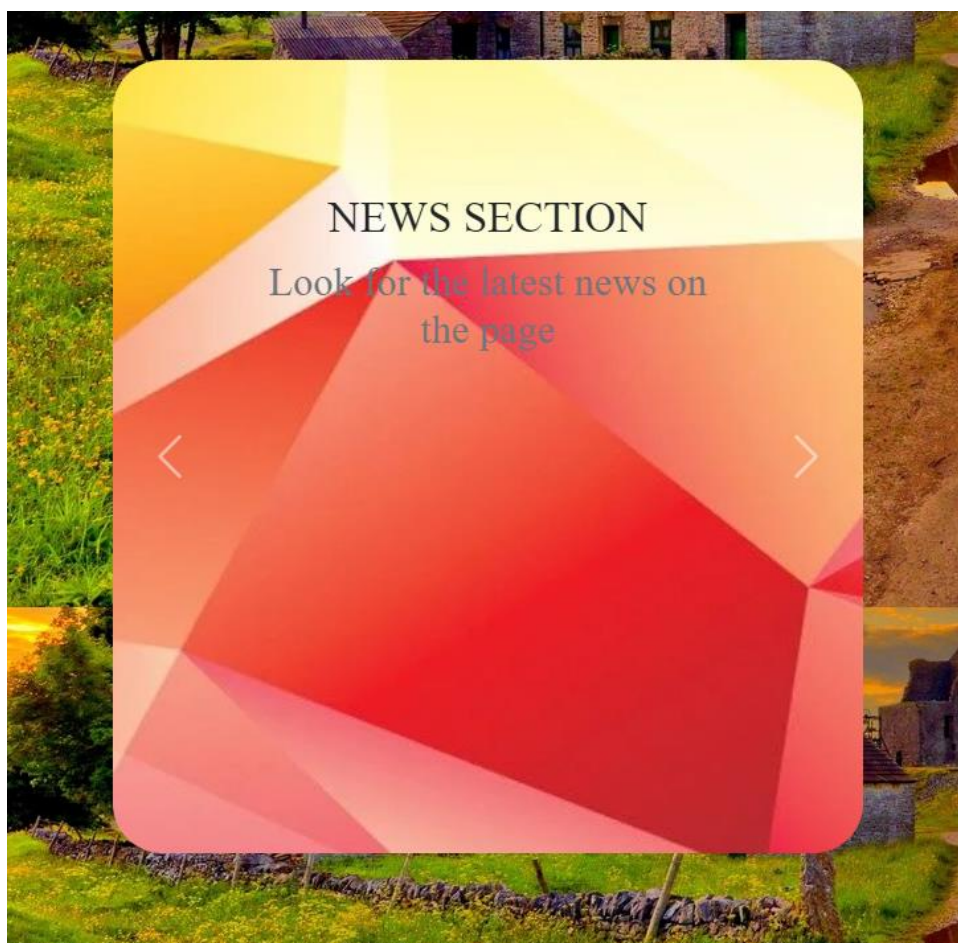


Ilustración 38 - Interfaz Gráfica: Home: Carrusel Publicaciones Vista de Computador



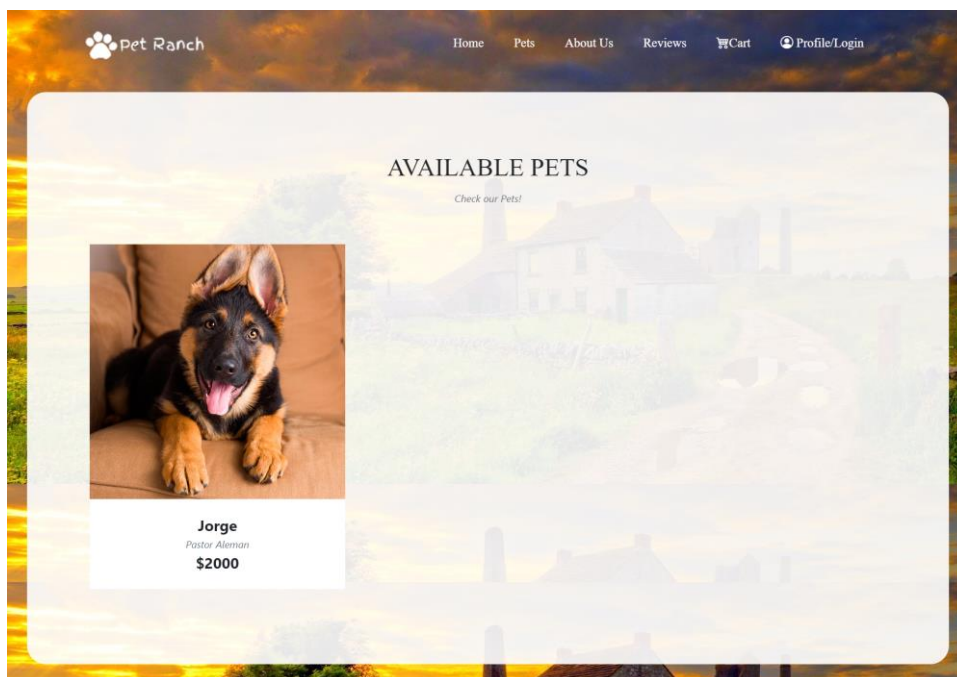
Ilustración 39 - Interfaz Gráfica: Home: Carrusel Publicaciones Vista de Smartphone

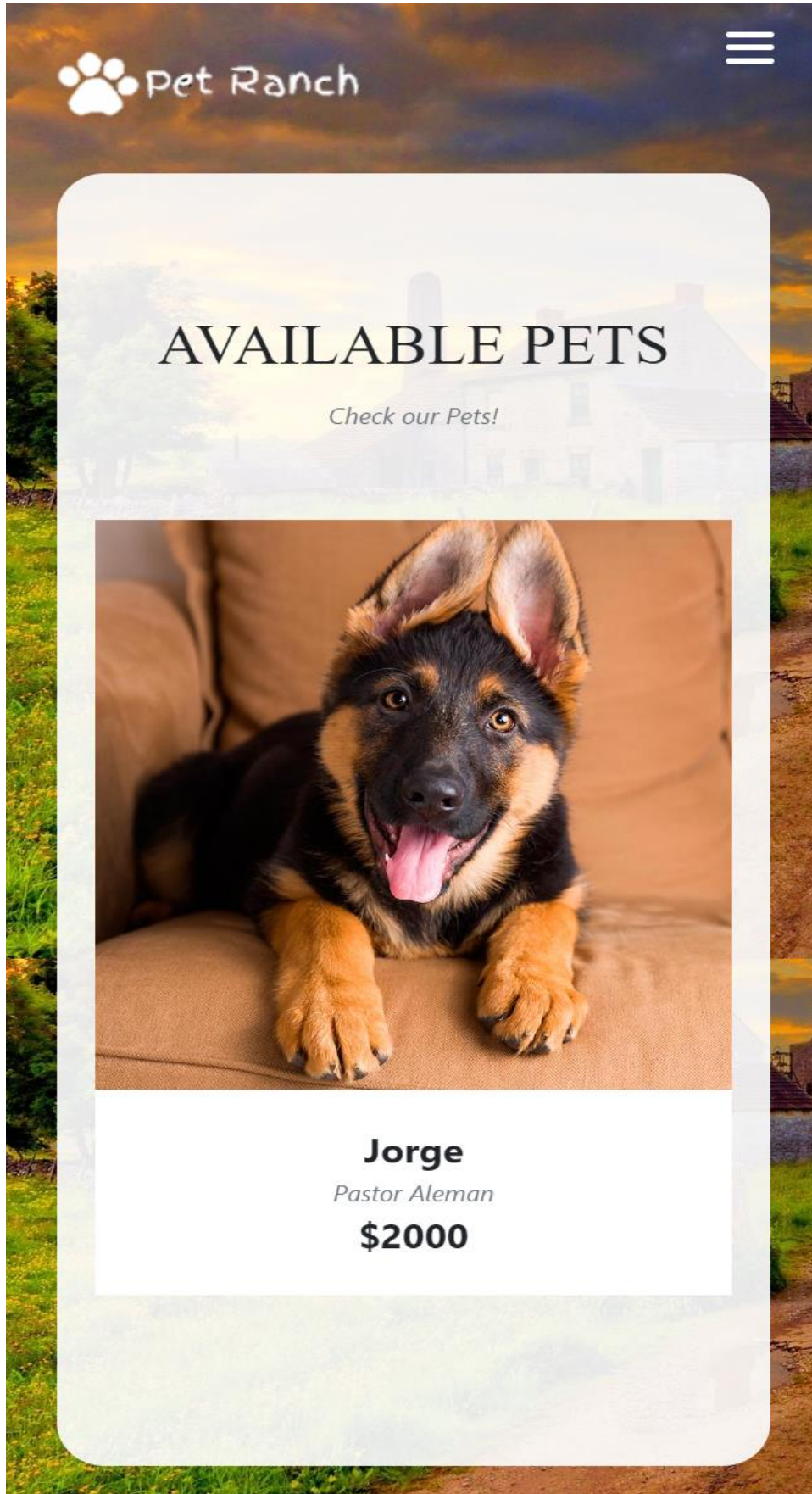


3.2.1.6. Pets

La ventana Pets muestra las mascotas disponibles.

Ilustración 40 - Interfaz Gráfica: Pets Vista de Computador





3.2.1.7. About Us

La ventana About Us contiene un diseño para que el negocio cliente pueda mostrar información sobre ellos. Lo que se encuentran en esta ventana por ahora es un mapa para poner información histórica del negocio, y una sección para dar a conocer las principales personas que lo conforman.

Ilustración 42 - Interfaz Gráfica: About us Vista de Computador

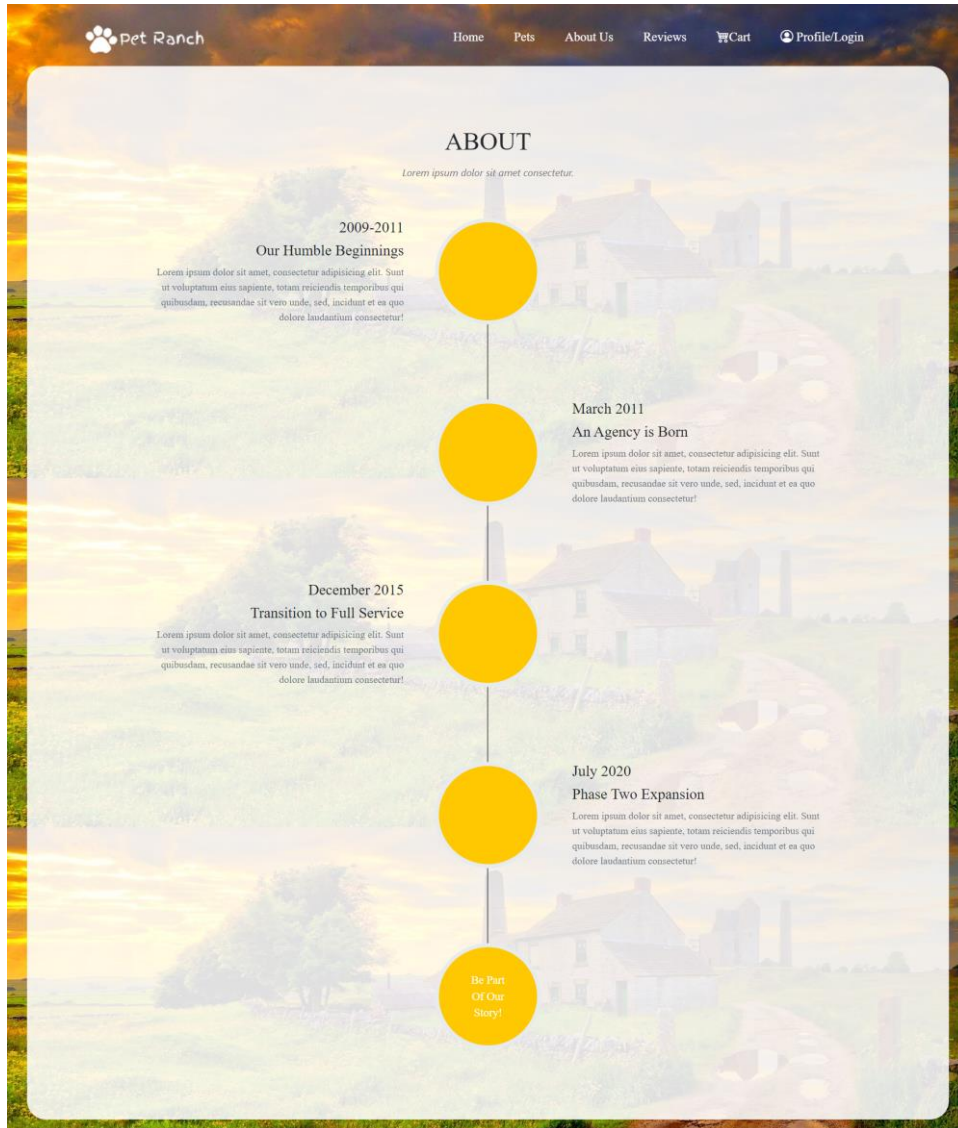


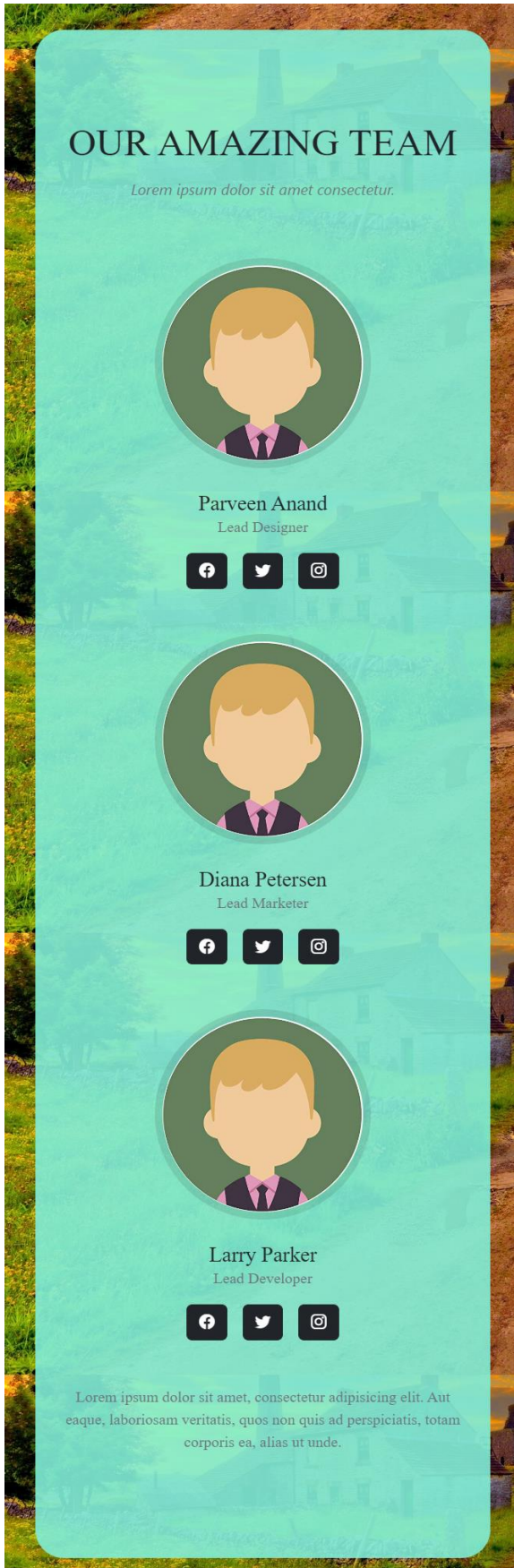
Ilustración 43 - Interfaz Gráfica: About us Vista de Smartphone



Ilustración 44 - Interfaz Gráfica: About us: Our team Vista de Computador



Ilustración 45 - Interfaz Gráfica: About us: Our Team Vista de Smartphone



3.2.1.8. Reviews

La ventana Reviews es un espacio donde se pueden ver y agregar comentarios sobre el negocio.

Ilustración 46 - Interfaz Gráfica: Reviews Vista de Computador

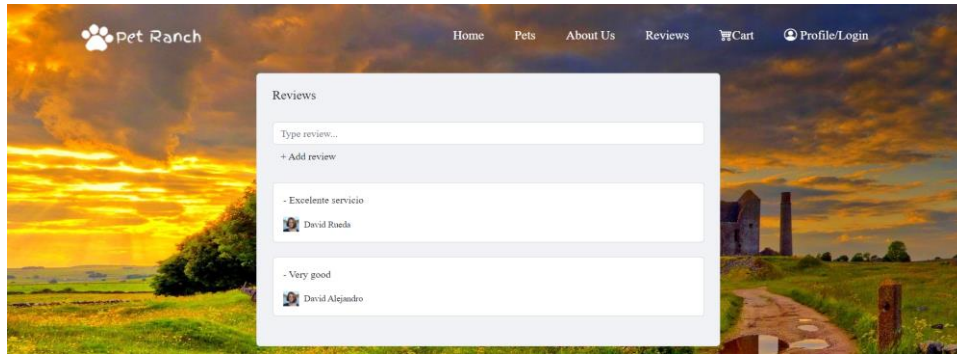
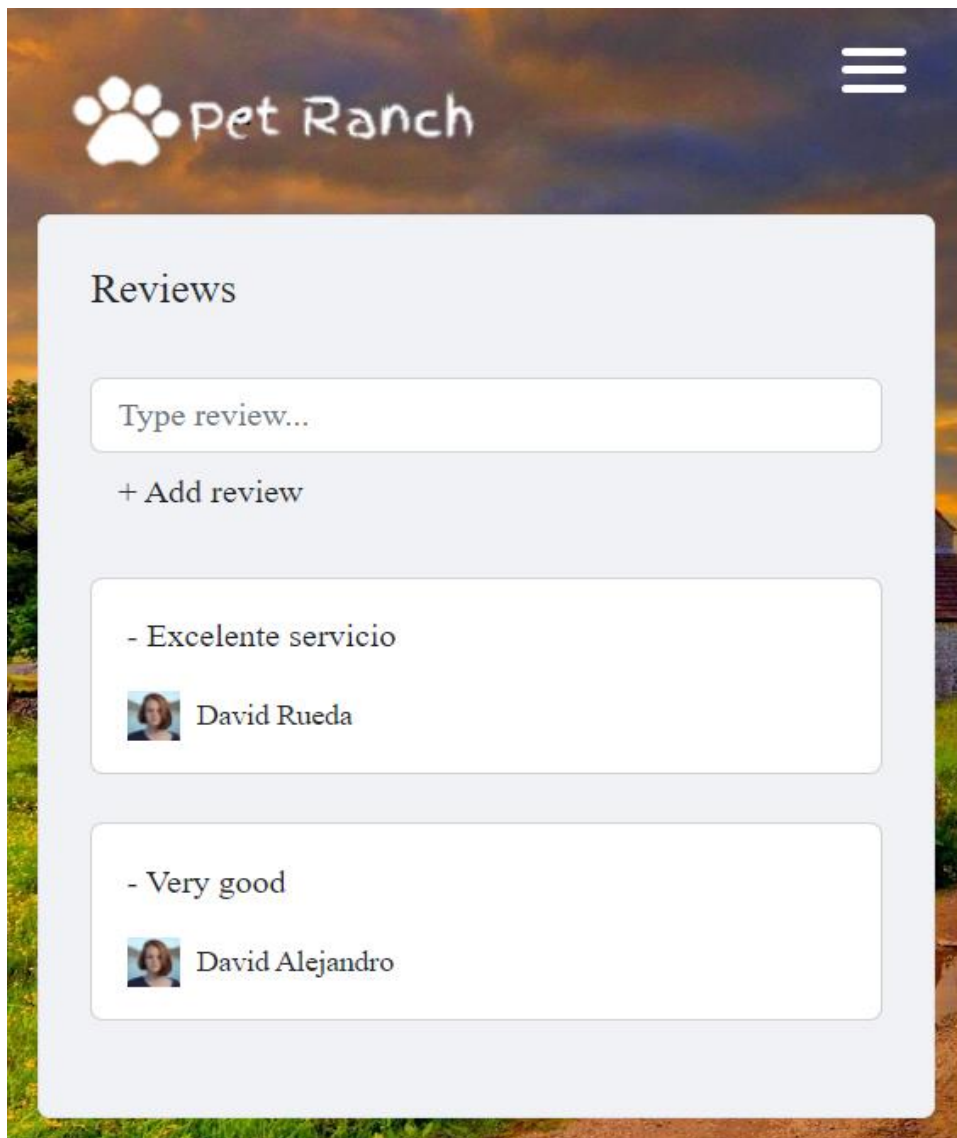


Ilustración 47 - Interfaz Gráfica: About us Vista de Smartphone



3.2.1.9. Cart

La Ventana Cart muestra el carrito de compras del usuario ingresado. Cuenta con una sección para mostrar las mascotas añadidas al carrito y otra sección dedicada a ingresar datos de pago (esta última sección es un diseño ya que la implementación de un sistema de cobro no está incluida en el desarrollo).

Ilustración 48 - Interfaz Gráfica: Cart Vista de Computador

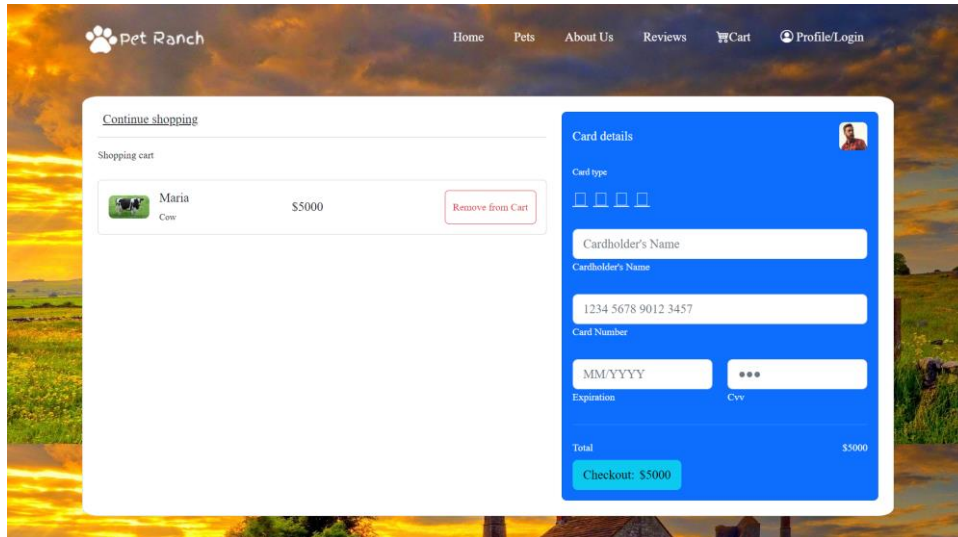
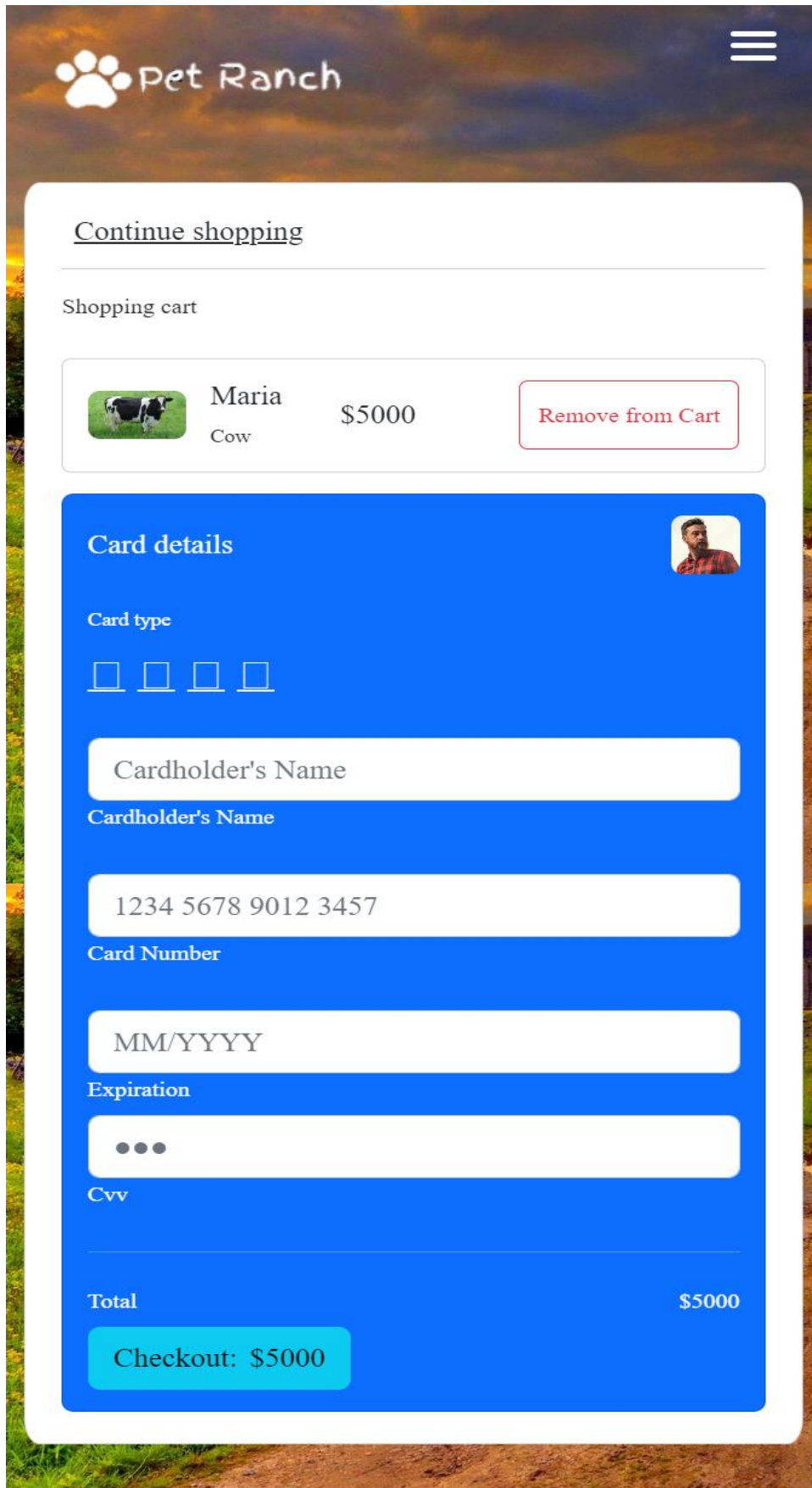


Ilustración 49 - Interfaz Gráfica: Cart Vista de Smartphone



3.2.1.10. Profile

La ventana Profile muestra la información del usuario una vez que haya iniciado sesión.

Ilustración 50 - Interfaz Gráfica: Profile Vista de Computador

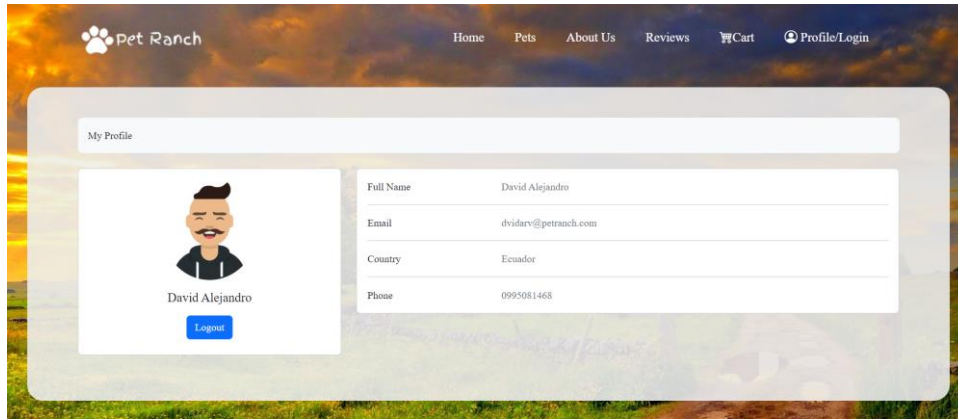
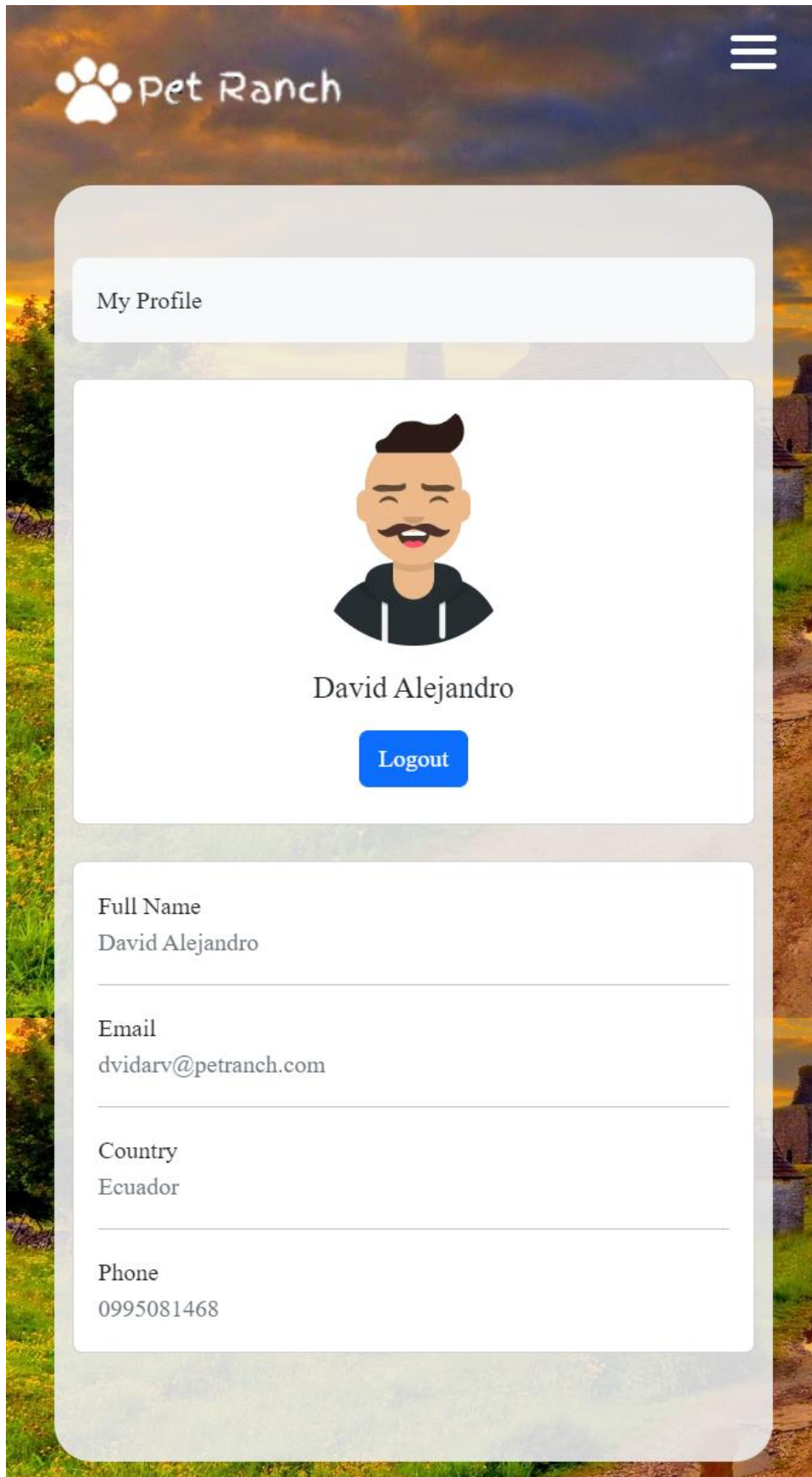


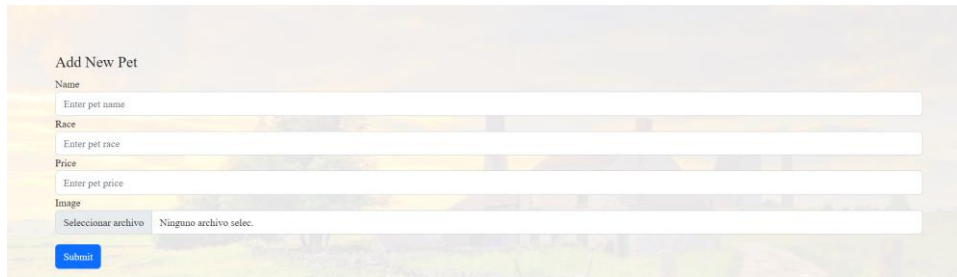
Ilustración 51 - Interfaz Gráfica: Profile Vista de Smartphone



3.2.1.11. Manage Pets

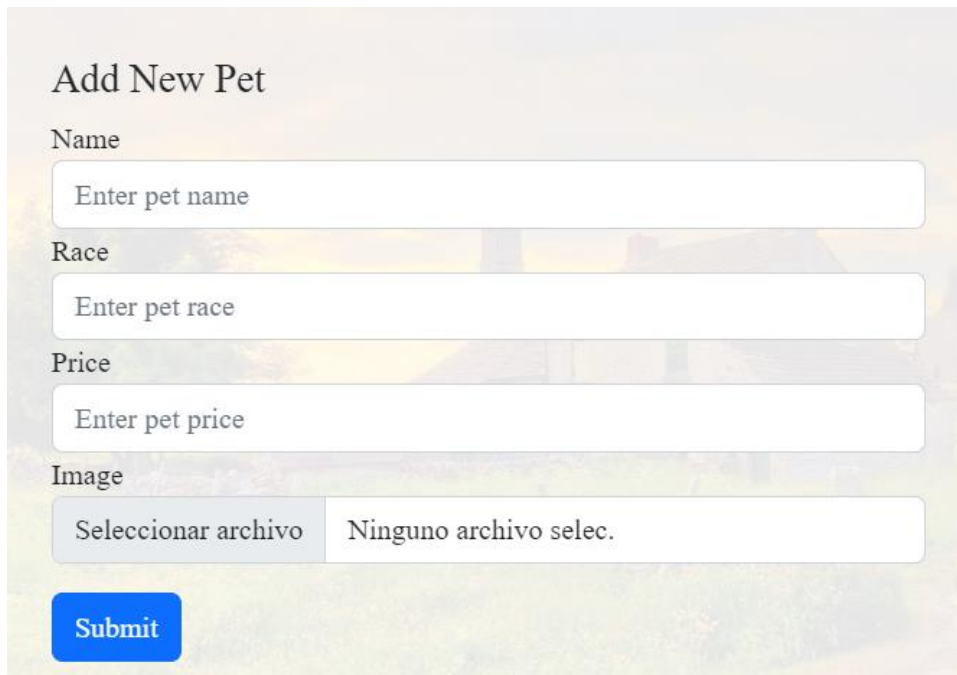
La ventana Manage Pets solo es accesible para usuarios administradores. Contiene dos secciones, una para agregar una mascota nueva al sistema, y otra para visualizar las mascotas existentes.

Ilustración 52 - Interfaz Gráfica: Manage Pets: New Pet vista de Computador



The screenshot shows a desktop view of the 'Add New Pet' form. The form is titled 'Add New Pet' and contains the following fields: 'Name' with a text input field containing 'Enter pet name'; 'Race' with a text input field containing 'Enter pet race'; 'Price' with a text input field containing 'Enter pet price'; and 'Image' with a file selection area containing a button labeled 'Seleccionar archivo' and the text 'Ninguno archivo selec.'. A blue 'Submit' button is located at the bottom left of the form.

Ilustración 53 - Interfaz Gráfica: Manage Pets: New Pet vista de Smartphone

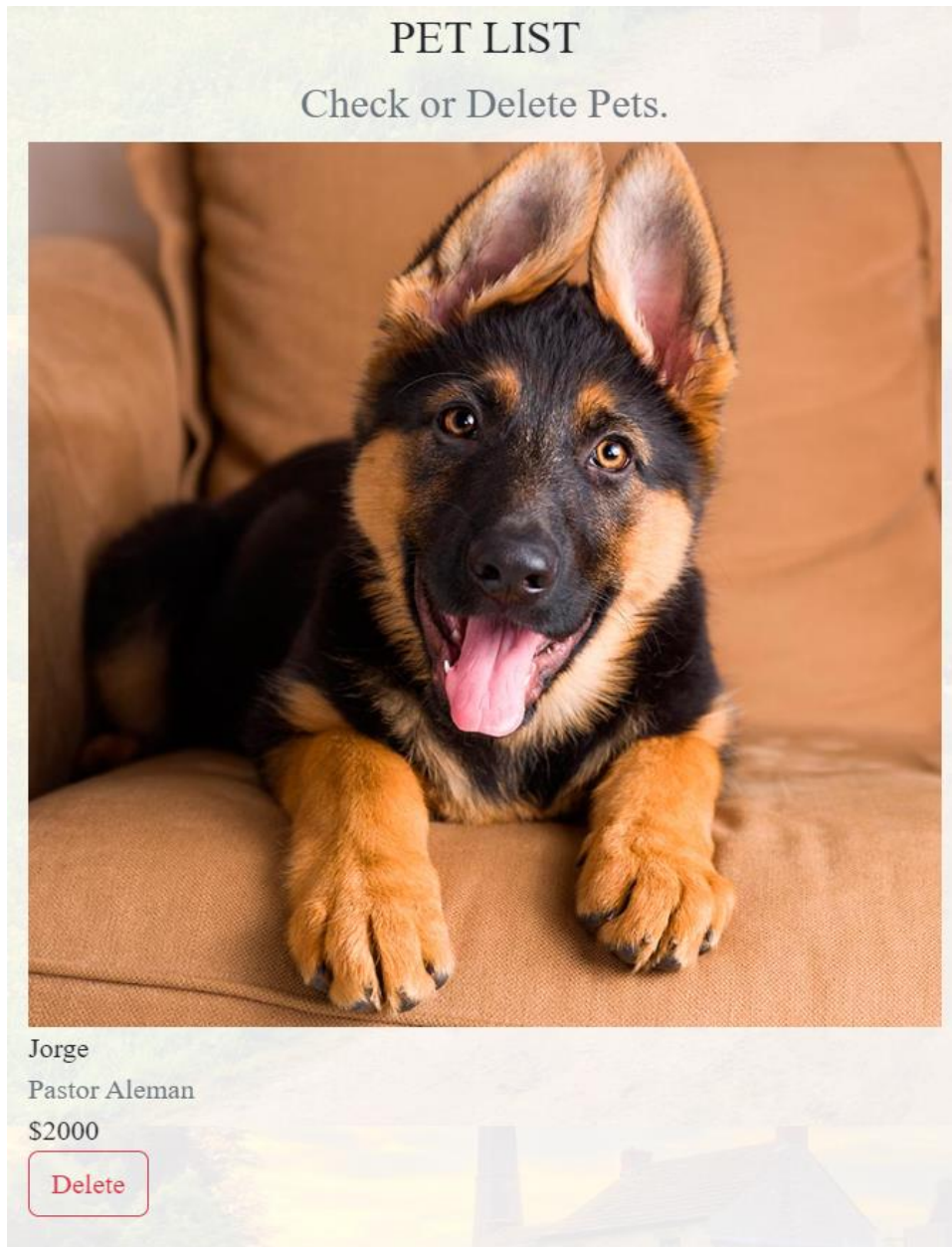


The screenshot shows a smartphone view of the 'Add New Pet' form. The form is titled 'Add New Pet' and contains the following fields: 'Name' with a text input field containing 'Enter pet name'; 'Race' with a text input field containing 'Enter pet race'; 'Price' with a text input field containing 'Enter pet price'; and 'Image' with a file selection area containing a button labeled 'Seleccionar archivo' and the text 'Ninguno archivo selec.'. A blue 'Submit' button is located at the bottom left of the form.

Ilustración 54 - Interfaz Gráfica: Manage Pets: Pet List vista de Computador



Ilustración 55 - Manage Pets: Pet List vista de Smartphone



3.2.1.12. Manage Posts

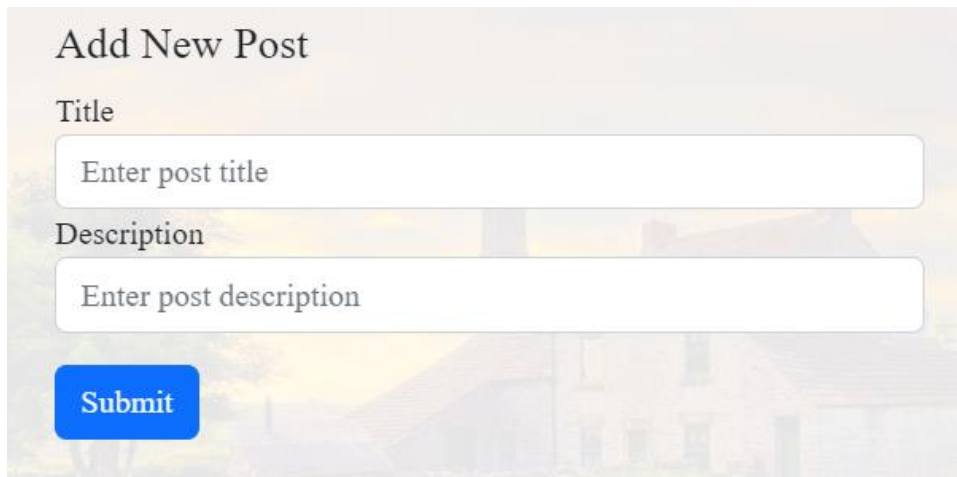
La ventana Manage Posts también es de acceso único para administradores. Tiene el mismo formato de la ventana Manage Pets; una ventana para agregar una publicación nueva, y otra donde se enlistan las publicaciones existentes.

Ilustración 56 - Interfaz Gráfica: Manage Posts: New Post vista de Computador



The screenshot shows a desktop view of the 'Add New Post' form. The form is titled 'Add New Post' and contains two input fields: 'Title' with the placeholder text 'Enter post title' and 'Description' with the placeholder text 'Enter post description'. A blue 'Submit' button is located at the bottom left of the form.

Ilustración 57 - Interfaz Gráfica: Manage Posts: New Post vista de Smartphone



The screenshot shows a smartphone view of the 'Add New Post' form. The form is titled 'Add New Post' and contains two input fields: 'Title' with the placeholder text 'Enter post title' and 'Description' with the placeholder text 'Enter post description'. A blue 'Submit' button is located at the bottom left of the form.

Ilustración 58 - Interfaz Gráfica: Manage Posts: Post List vista de Computador



The screenshot shows a desktop view of the 'POST LIST' page. The page is titled 'POST LIST' and has the subtitle 'Check or Delete Posts.'. There are two columns of text. The left column contains the text 'Having a problem? If you are having a problem, you can contact us by email or social networks at the footer of the page.' and a 'Delete' button. The right column contains the text 'November Discount! All our pet companions have 50% discount all month of November!' and a 'Delete' button.

Ilustración 59 - Interfaz Gráfica: Manage Posts: Post List vista de Smartphone



3.3. Implementación de la Base de Datos

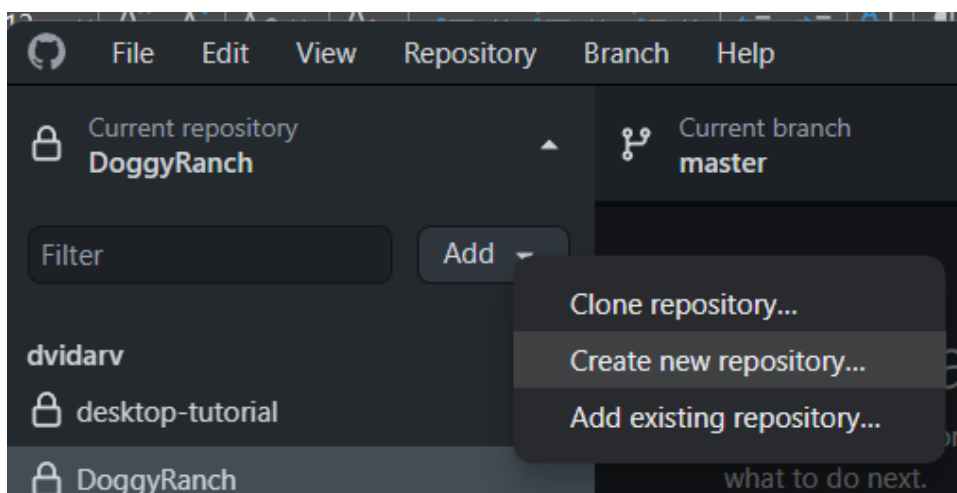
3.3.1. Requisitos para la conexión con Firebase

3.3.1.1. Github

Para poder usar los servicios de Firebase tanto para Hosting como para almacenamiento y base de datos, primero se necesita crear un repositorio en Github, para esto, bajaremos Github desktop del sitio oficial en el siguiente link <https://desktop.github.com/>.

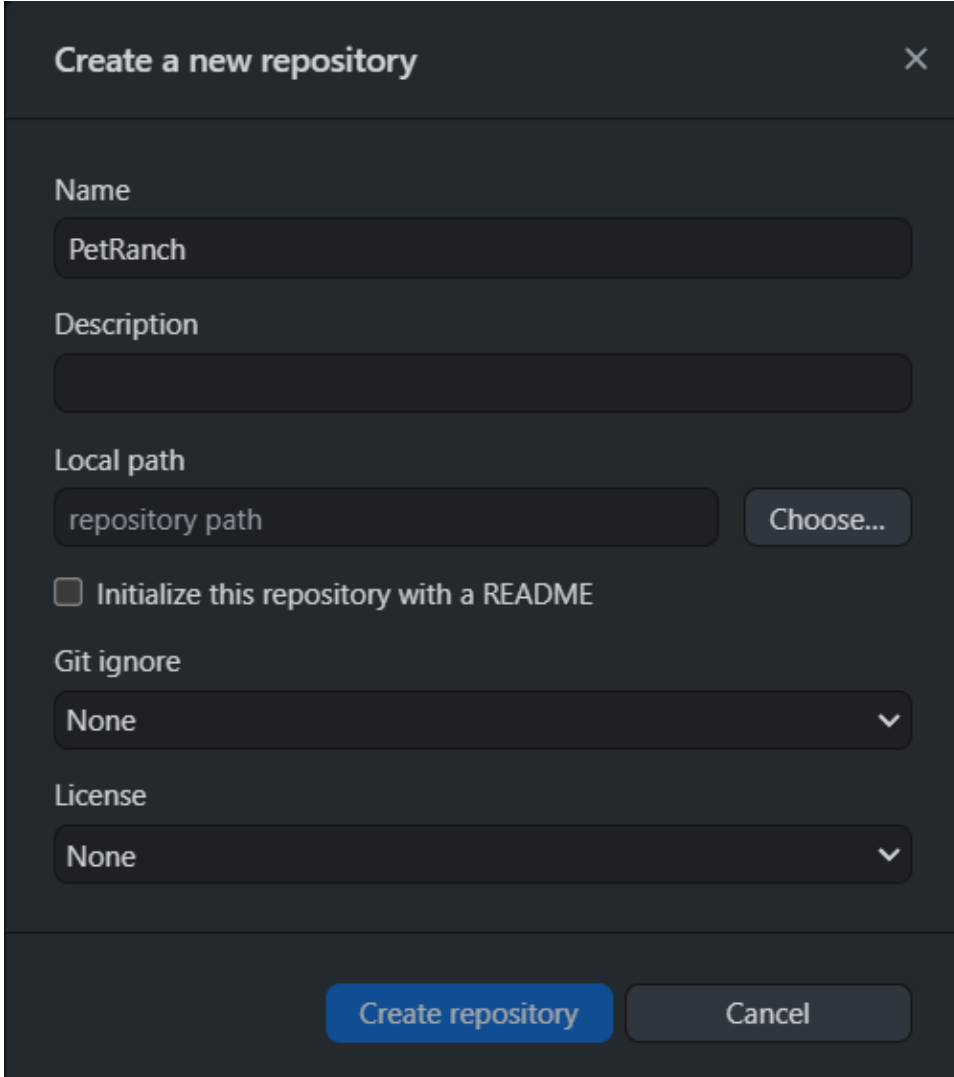
Una vez instalado, abrimos Github Desktop, iniciamos sesión, y nos dirigimos a la parte superior izquierda. Donde habrá un menú desplegable llamado "Add". Seleccionamos el menú y de las opciones desplegadas, seleccionaremos "Create new repository".

Ilustración 60 - Nuevo Repositorio en Github 1



Se abrirá una ventana de creación de un repositorio nuevo, donde daremos un nombre a nuestro proyecto, una descripción opcional, y nos pedirá ingresar la ubicación del proyecto que deseamos subir a Github. Al llenar estos campos le daremos en “Create repository” y nuestro proyecto se añadirá.

Ilustración 61 - Nuevo Repositorio en Github 2



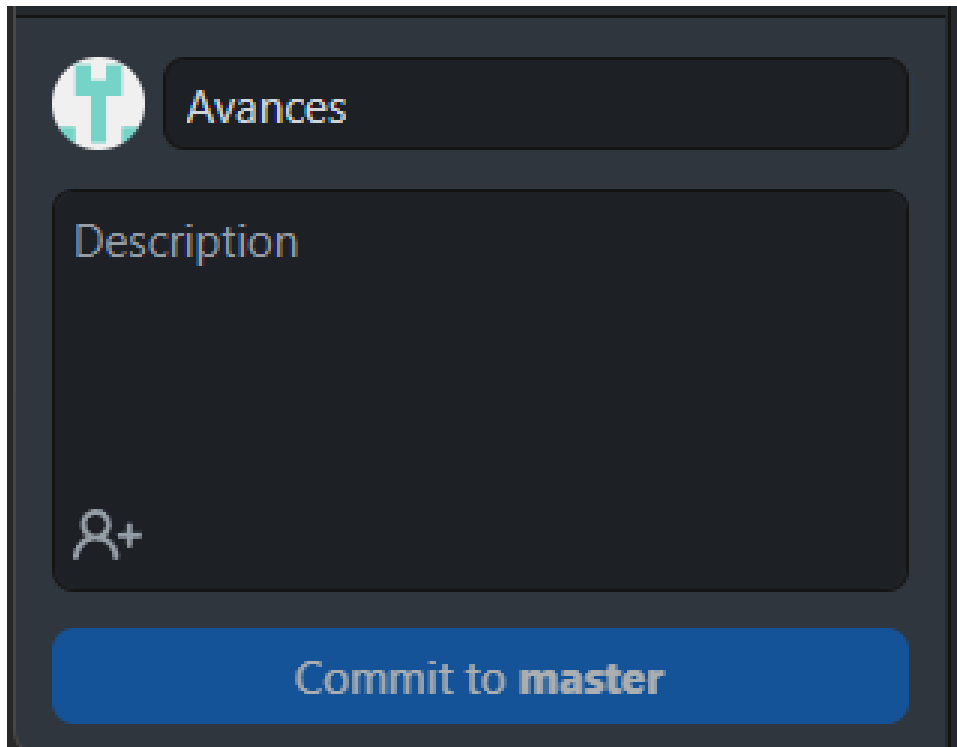
The image shows a "Create a new repository" dialog box. The title bar says "Create a new repository" with a close button (X) on the right. The form contains the following fields and options:

- Name:** A text input field containing "PetRanch".
- Description:** An empty text input field.
- Local path:** A text input field containing "repository path" and a "Choose..." button to the right.
- Initialize this repository with a README:** An unchecked checkbox.
- Git ignore:** A dropdown menu currently set to "None".
- License:** A dropdown menu currently set to "None".

At the bottom of the dialog, there are two buttons: "Create repository" (highlighted in blue) and "Cancel".

Se debe recordar que, siempre que se realicen cambios en el aplicativo, se deben subir los cambios desde la parte inferior izquierda de Github. Simplemente ponemos un nombre al cambio deseado, una descripción opcional, y seleccionamos la opción “Commit to Master”. Finalmente En la interfaz principal de Github seleccionamos la opción “Push Origin” para subir los cambios al repositorio en la nube.

Ilustración 62 - Exportar Avances en Github

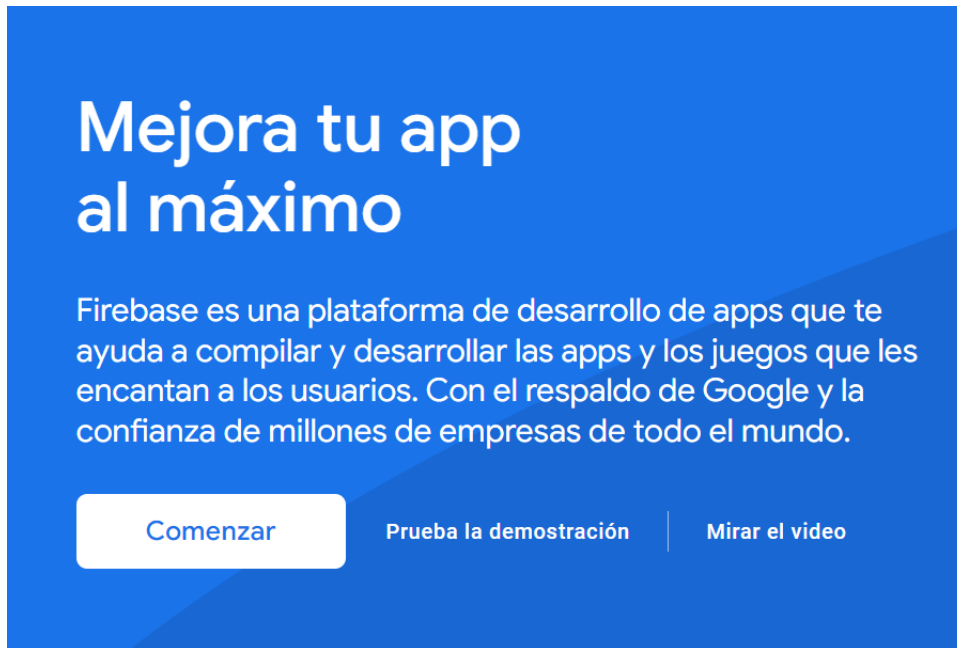


3.3.1.2. Firebase Project

Otro requisito importante es la creación del proyecto de Firebase que nos permitirá directamente crear la base de datos, conectar con un almacenamiento en la nube y Hostear nuestro aplicativo.

Para esto tenemos que ir al sitio oficial de Firebase en <https://firebase.google.com/?hl=es> y registrarnos o iniciar sesión. Una vez iniciada sesión, seleccionamos la opción “Comenzar” que se encuentra en la página principal del sitio.

Ilustración 63 - Página Principal de Firebase



Esto nos redirigirá a nuestra sección de proyectos, donde seleccionaremos la opción “Agregar Proyecto”.

Ilustración 64 - Agregar Nuevo Proyecto Firebase



Luego solo tendremos que nombrar nuestro proyecto y lo crearemos.

3.3.1.3. Instalación de herramientas de Firebase en el proyecto angular

Regresamos a nuestro aplicativo en Visual Studio Code y abrimos una terminal nueva. Para instalar las herramientas de firebase tendremos que ejecutar el comando *npm install Firebase*, esto instalará archivos necesarios para trabajar con firebase en nuestro proyecto. Luego ejecutaremos el comando *ng add '@angular/fire'* y la terminal nos pedirá seleccionar una cuenta de firebase y un proyecto para trabajar. Esto instalará los archivos necesarios de firebase en nuestro proyecto de angular y procederá a preguntar que servicios deseamos usar. Seleccionaremos 4 servicios: Hosting para poder hacer un deploy de nuestro proyecto en la web, Storage, para poder almacenar archivos en la nube, Real Time Database para poder tener el servicio de base de datos, y Authentication para poder manejar los perfiles del aplicativo. Una vez seleccionados estos servicios, procedemos con la instalación, donde se nos pedirá iniciar sesión en github y elegir un perfil de trabajo. Las siguientes opciones que aparezcan en la terminal solo hay que aceptarlas hasta que haya concluido el proceso.

Al terminar el proceso deben haber ocurrido los siguientes cambios en nuestro proyecto angular:

Creación de las carpetas *.firebase*, y *.github* dentro de la carpeta principal del proyecto; creación de los archivos *firebaserc*, *firebase.json*, *firestore.indexes.json*, *firestore.rules* y *.gitignore* en la carpeta principal del proyecto; creación de la carpeta *firebase* dentro de la carpeta *node_modules*.

También hay que verificar que los archivos de la carpeta *environments* dentro de la carpeta *src* dentro de la carpeta principal del proyecto, se hayan actualizado con la información del proyecto de firebase.

Ilustración 65 - Archivo environment.ts

```
TS environment.ts x
src > environments > TS environment.ts > ...
1 // This file can be replaced during build by using the `fileReplacements` array.
2 // `ng build` replaces `environment.ts` with `environment.prod.ts`.
3 // The list of file replacements can be found in `angular.json`.
4
5 export const environment = {
6   firebase: {
7     projectId: 'doggyranch-b36ea',
8     appId: '1:660502245300:web:f908f8e150b44e5d3d8ce1',
9     databaseURL: 'https://doggyranch-b36ea-default-rtdb.firebaseio.com',
10    storageBucket: 'doggyranch-b36ea.appspot.com',
11    locationId: 'us-central',
12    apiKey: 'AizaSyBvyiC7XmL2Lrjw0dU3gIIo9QkFIzyFQA0',
13    authDomain: 'doggyranch-b36ea.firebaseio.com',
14    messagingSenderId: '660502245300',
15    measurementId: 'G-TK01WDZCGG',
16  },
17  production: false
18 };
19
20 /*
21  * For easier debugging in development mode, you can import the following file
22  * to ignore zone related error stack frames such as `zone.run`, `zoneDelegate.invokeTask`.
23  *
24  * This import should be commented out in production mode because it will have a negative impact
25  * on performance if an error is thrown.
26  */
```

Finalmente, dentro de la carpeta app, dentro de src, hay que verificar que en el archivo app.module.ts se hayan importado las herramientas para trabajar con firebase.

Ilustración 66 – Imports Firebase en el Archivo app.module.ts

```
],
imports: [
  BrowserModule,
  AppRoutingModule,
  provideFirebaseApp(() => initializeApp(environment.firebase)),
  provideDatabase(() => getDatabase()),
  provideFirestore(() => getFirestore()),
  ReactiveFormsModule,
  provideStorage(() => getStorage()),
  provideAuth(() => getAuth()),
],
providers: [
```

Con esto concluye la vinculación hacia firebase. Para verificar que todo funcione perfectamente, se debe ejecutar el comando *firebase deploy*. Esto compilará el proyecto en una url asignada al mismo en la que podremos ver nuestro aplicativo. Este comando también debe ejecutarse siempre que se deseen guardarlos datos en la web y antes de hacer un push desde github.

3.3.2. Incorporación de los servicios de Firebase al aplicativo

Para poder trabajar con crud's desde el aplicativo y que los cambios se vean reflejados en la base de datos es necesario trabajar con dos tipos de componentes: interfaces y servicios.

Las interfaces son el modelo de los objetos que va a ocupar la base de datos junto con sus atributos. A continuación, se presenta como ejemplo la interfaz de usuarios.

Ilustración 67 - Estructura de la Interfaz User

```
TS user.interface.ts X
src > app > interfaces > TS user.interface.ts > User
1  import Dog from "./dog.interface";
2
3  export default interface User{
4      id?: string;
5      name: string;
6      email: string;
7      password: string;
8      country: string;
9      image: string;
10     phone: number;
11     cart: Dog[];
12 }
```

Para el aplicativo, se trabajará con una interfaz usuario, una de mascota, una de publicaciones y una para las reviews.

El otro componente son los servicios, estos se encargan de generar las funciones necesarias para comunicar la aplicación con la base de datos. Dentro de los servicios se encuentran los cruds. Por ejemplo podemos ver el siguiente servicio User como tiene funciones para el registro, login, autenticación, creación e impresión de usuarios.

Ilustración 68 - Estructura del Servicio User

```
TS user.service.ts X
src > app > services > TS user.service.ts > UserService
24   register( email:any, password : any){
25     | return createUserWithEmailAndPassword(this.auth, email, password);
26   }
27
28   login({ email, password }: any){
29     | return signInWithEmailAndPassword(this.auth, email, password);
30   }
31
32   loginWithGoogle(){
33     | return signInWithPopup(this.auth, new GoogleAuthProvider());
34   }
35
36   logout(){
37     | return signOut(this.auth);
38   }
39
40   async getId(){
41     const user = await this.auth.currentUser;
42     if(user === null){
43       | return null;
44     } else{
45       | return user?.uid;
46     }
47   }
48
49   addUser(user: User, userId: any){
50     const userRef = collection(this.firestore, 'users');
51     return setDoc(doc(userRef, userId ), user);
52   }
53
54   stateAuth(){
55     return authState(this.auth);
56   }
57
58   async getUser(id: any){
59     const userRef = doc(this.firestore, 'users', id);
60     const usersnap = await getDoc(userRef);
61     return usersnap.data();
62   }
```

La manera en que se trabaja entre componentes, servicios e interfaces es la siguiente: Los componentes llaman a los servicios para aplicar funciones y estos servicios se comunican con la interfaz para conocer el modelo con que trabaja el objeto en uso. Para entender esto de mejor manera se mostrará un ejemplo con un usuario:

- Desde el archivo .html del componente Profile se llama un objeto usuario con los atributos name, email, country y phone.

Ilustración 69 - Mostrar Usuarios de la Base de Datos: Archivo Html

```
<div class="col-sm-3">
  <p class="mb-0">Full Name</p>
</div>
<div class="col-sm-9">
  <p class="text-muted mb-0">{{user.name}}</p>
</div>
</div>
<hr>
<div class="row">
  <div class="col-sm-3">
    <p class="mb-0">Email</p>
  </div>
  <div class="col-sm-9">
    <p class="text-muted mb-0">{{user.email}}</p>
  </div>
</div>
<hr>
<div class="row">
  <div class="col-sm-3">
    <p class="mb-0">Country</p>
  </div>
  <div class="col-sm-9">
    <p class="text-muted mb-0">{{user.country}}</p>
  </div>
</div>
<hr>
<div class="row">
  <div class="col-sm-3">
    <p class="mb-0">Phone</p>
  </div>
  <div class="col-sm-9">
    <p class="text-muted mb-0">{{user.phone}}</p>
  </div>
</div>
```

- Dentro del archivo .ts del componente Profile se llama al servicio .userService y la función getUser().

Ilustración 70 - Mostrar Usuarios de la Base de Datos: Llamada al Servicio User

```
    this.userService.stateAuth().subscribe( async res => {
      if(res !== null){
        this.uid = res.uid;
        this.userService.getUser(res.uid);
        const response = await this.userService.getUser(res.uid);
        this.user = response as User;
        this.userService.getCart(res.uid).subscribe(dogs =>{
          this.cart = dogs;
        })
      }
    })
  }
}
```

- En el servicio User podemos visualizar la función a la que se está llamando GetUser

Ilustración 71 - Mostrar Usuarios de la Base de Datos: Función Get del Servicio User

```
async getUser(id: any){
  const userRef = doc(this.firestore, 'users', id);
  const usersnap = await getDoc(userRef);
  return usersnap.data();
}
```

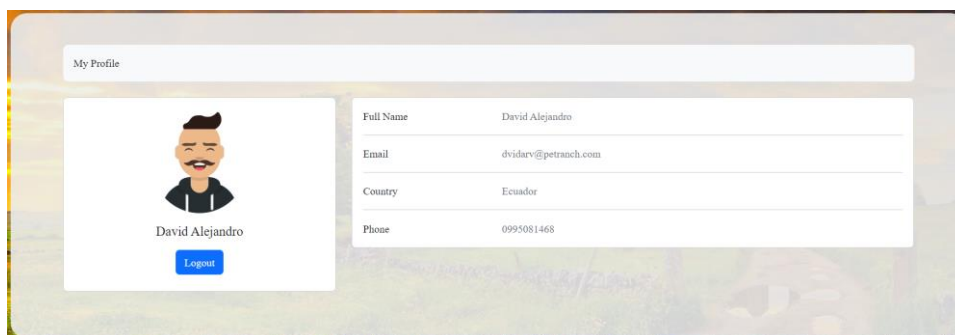
- Vemos que esta función se está comunicando a la base de datos de firebase para importar un usuario.
- Finalmente, para saber que atributos debe importar del usuario, se debe importar la interfaz User.

Ilustración 72 - Mostrar Usuarios de la Base de Datos: Interfaz User

```
2
3  export default interface User{
4      id?: string;
5      name: string;
6      email: string;
7      password: string;
8      country: string;
9      image: string;
10     phone: number;
11     cart: Dog[];
12 }
```

- La respuesta a todo este proceso es que la información de un usuario se pueda visualizar desde su perfil de la siguiente manera:

Ilustración 73 - Mostrar Usuarios de la Base de Datos: Vista Final



3.4. Implementación de las Funcionalidades

Tabla 1 - Tabla de Funcionalidades

Funcionalidad	Descripción	Comunicación con la Base de Datos
Register	Funcionalidad para registrar un usuario nuevo en el aplicativo.	Se usa AddDoc para crear un registro de usuario en la base de datos.
Login	Funcionalidad para iniciar sesión en el aplicativo. Es necesario haberse registrado previamente.	Se usa un GetDoc de todos los registros de usuario para verificar la existencia del usuario en la base de datos y una función de autenticación para permitir el acceso.
Logout	Funcionalidad para cerrar sesión en el aplicativo. Debe haberse iniciado sesión previamente.	Se usa un GetDoc para identificar el usuario que está en sesión y una función de autenticación cerrar la sesión.
Show Profile Info	Funcionalidad para mostrar la información del usuario que inició sesión.	Se usa un GetDoc para recibir la información del usuario de la base de datos.

Manage User Permission	Funcionalidad para restringir el uso de funcionalidades dependiendo del usuario.	Se usa un GetDoc para recibir la información del usuario en sesión y verificar si es administrador o no.
Add Pet	Funcionalidad para añadir una mascota nueva al aplicativo. (Solo para administradores).	Se usa un AddDoc para añadir un registro de mascota nueva a la base de datos.
Delete Pet	Funcionalidad para eliminar una mascota del aplicativo (Solo para administradores).	Se usa un DeleteDoc para eliminar un registro de mascota de la base de datos.
Show Pets	Funcionalidad para mostrar todas las mascotas del aplicativo.	Se usa un GetDoc para recibir todos los registros de mascotas de la base de datos.
Add Post	Funcionalidad para añadir una publicación nueva al aplicativo (Solo para administradores).	Se usa un AddDoc para añadir un registro de publicación a la base de datos.
Delete Post	Funcionalidad para eliminar una publicación del aplicativo.	Se usa un DeleteDoc para eliminar un registro de publicación de la base de datos.
Show Posts	Funcionalidad para mostrar todas las publicaciones en el aplicativo.	Se usa un GetDoc para recibir todos los registros de publicaciones de la base de datos.
Add Review	Funcionalidad para añadir un comentario en el aplicativo. Es necesario haber iniciado sesión previamente.	Se usa un GetDoc para identificar el usuario en sesión y un AddDoc para añadir un registro de comentario a la base de datos.

Show Reviews	Funcionalidad para mostrar todos los comentarios en el aplicativo.	Se usa un GetDoc para recibir todos los registros de comentarios de la base de datos.
Add Pet to Cart	Funcionalidad para añadir una mascota al carrito de compras del usuario en sesión. Es necesario haber iniciado sesión previamente.	Se usa un GetDoc para identificar el usuario en sesión y un SetDoc para añadir una mascota al carrito de compras dentro del usuario en sesión en la base de datos.
Remove Pet from Cart	Funcionalidad para eliminar una mascota del carrito de compras del usuario en sesión. Es necesario haber iniciado sesión previamente.	Se usa un GetDoc para identificar el usuario en sesión y un DeleteDoc para eliminar un registro de mascota del carrito de compras del usuario.
Show Cart	Funcionalidad para mostrar todas las mascotas del carrito de compras del usuario en sesión. Es necesario haber iniciado sesión previamente.	Se usa un GetDoc para identificar el usuario en sesión y un GetDoc para recibir todos los registros de mascotas dentro del registro de carrito de compras del usuario en sesión de la base de datos.

4. CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Los criaderos de animales son centros de cría y venta de seres que pueden ayudar al ser humano a sobrellevar la soledad y problemas emocionales, si bien es cierto que en muchos lugares no se considera moral la venta de animales, muchos de los criaderos son lugares de buen trato hacia estos.

- Bootstrap es una herramienta que facilita y reduce el tiempo de diseño de un aplicativo web debido a sus componentes prediseñados lindos para el ojo humano y por su programación responsive.
- Las metodologías de desarrollo ágil como por ejemplo SCRUM son muy útiles para el desarrollo de aplicativos web ya que en el mundo laboral estos no pueden requerir de mucho tiempo, así que esta metodología ayuda a terminar un trabajo bien hecho y en el menor tiempo posible.
- El framework angular es una herramienta poderosa para el desarrollo de páginas web debido a su estructura simple y lenguajes de trabajo sencillos de entender y manejar.

4.2. Recomendaciones

- Se recomienda añadir un registro con Google para simplificar el proceso de creación de usuarios.
- Es buena idea implementar un sistema de verificación de correos y recuperación de contraseñas para mejorar la seguridad del aplicativo.
- La activación de un sistema de cobros con tarjeta o Paypal es importante al momento vender el aplicativo a un cliente.
- Se podría modificar al aplicativo para poder añadir y mostrar más de una imagen por producto.

ANEXOS

- Link del proyecto en Github: <https://github.com/dvidarv/DoggyRanch.git>
- Link del aplicativo web: [DoggyRanch \(doggyranch-b36ea.web.app\)](https://doggyranch-b36ea.web.app)
- Usuario Administrador para acceder a las funciones administrativas (notar que las opciones administrativas se encuentran en el footer de la página luego de haber iniciado sesión como administrador):

Usuario: dvidarv@petranch.com

Contraseña: 123456

BIBLIOGRAFÍA

- Angular. (2022). *What is Angular?* Obtenido de Angular.io: <https://angular.io/guide/what-is-angular>
- Bootstrap. (s.f.). *Get started with Bootstrap*. Obtenido de [getbootstrap.com](https://getbootstrap.com/docs/5.2/getting-started/introduction/): <https://getbootstrap.com/docs/5.2/getting-started/introduction/>
- Edix, R. (2022). *Framework: qué es, para qué sirve y algunos ejemplos*. Obtenido de Edix España: <https://www.edix.com/es/instituto/framework/>
- Equipo Editorial Etecé. (2021). *Base de Datos - Conceptos, tipos y ejemplos*. Obtenido de Concepto.de: <https://concepto.de/base-de-datos/>
- Equipo Editorial Etecé. (s.f.). *Página Web*. Obtenido de 2021: <https://concepto.de/pagina-web/>
- GCF Global. (s.f.). *Informática Básica - ¿Qué son las aplicaciones web?* Obtenido de GCFGlobal: <https://www.scribbr.es/citar/generador/folders/40dfINbzxbwtf4pE9fm05v/lists/4EXGHXtDpFaOB3FG5fKerX/?lastAddedSourceId=5SugP7dx4bhMg2z73DXRjf>
- Gonzales, S. (2019). *¿Qué es un diseño web responsive?* Obtenido de Cyberclick: <https://www.scribbr.es/citar/generador/folders/40dfINbzxbwtf4pE9fm05v/lists/4EXGHXtDpFaOB3FG5fKerX/?lastAddedSourceId=5SugP7dx4bhMg2z73DXRjf>
- Google. (s.f.). *Make your app the best it can be*. Obtenido de Firebase: <https://firebase.google.com/>
- Hurtado, J. S. (2021). *Cómo funciona la Metodología Scrum: Qué es cómo utilizarla*. Obtenido de [iebschool.com](https://www.iebschool.com): <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>
- Orraiz, L. (2020). *Editores de código: ¿Qué son y para qué sirven?* Obtenido de PinguinoDigital: <https://pinguinodigital.com/blog/editores-de-codigo/>
- Souza. (2021). *Entiende la diferencias entre Front-End y Back-End en el ambiente de los sitios web*. Obtenido de Rock Content: <https://rockcontent.com/es/blog/front-end-y-back-end/>

Unidad de Bienestar Animal - Ministerio de Agricultura, G. y. (2022). *Maga*. Obtenido de <https://www.maga.gob.gt/sitios/uba/criaderos-2/#:%7E:text=Un%20criadero%20es%20cualquier%20lugar,y%20con%20fines%20de%20lucro>

Visual Studio. (2019). *Documentation for Visual Studio Code*. Obtenido de Visual Studio Code: <https://code.visualstudio.com/docs>