



Trabajo fin de grado

Grado Superior de desarrollo de aplicaciones multiplataforma

Diseño y desarrollo de una aplicación móvil para la Peluquería canina Requeteguaupos

Autor

Ibai Ventura Arévalo

DNI

53808480A

Tutor

Anahí Mula de la Banda

Curso académico 2018 | 2020

1. Resumen

En pleno siglo XXI todo el mundo tiene a su alcance un Smartphone ya que facilita mucho el acceso a internet o estar comunicado con familiares y realizar solicitudes a empresas tanto públicas como privadas. Android es el sistema operativo más utilizado en dispositivos móviles ya que resulta más asequible un Smartphone que posea este sistema operativo en comparación con iOS.

Por este motivo he decidido crear una aplicación Android para facilitar la consulta de nuestros servicios y precios de los mismo, solicitar citas en nuestra peluquería, solicitar un servicio de recogida a domicilio e incluso obtener información sobre la peluquería.

Lo que diferencia a nuestra aplicación es que se creó con la intención de facilitar a nuestros clientes la puesta en contacto con nosotros ya que pocas empresas permiten solicitar citas en sus instalaciones a través de una aplicación móvil. Esto otorga comodidad a nuestros clientes lo que conlleva a obtener mayores ingresos a la empresa.

Este proyecto intenta ofrecer una aplicación fiable y estable que permitirá solicitar una cita previa sin tener que abonar dinero a través de internet, los únicos datos que pedirá la aplicación para la reserva serán: Número de teléfono y nombre del cliente, nombre de la mascota, peso, raza y por último el servicio que desea solicitar.

A lo largo del documento se presentarán las tecnologías utilizadas. También podremos encontrar un manual de uso para entender el funcionamiento correcto de la aplicación.

Palabras Clave

La siguiente lista contiene palabras clave para catalogar el proyecto.

- Peluquería.
- Mascota.
- Animales.
- Perro.
- Gato.
- Higiene.
- Corte de pelo.
- Baño.
- Limpieza.
- Cepillado.
- Secado.

2. Abstract

In the 21st century, everyone has a Smartphone at their range as it facilitates internet access or communicates with family members and makes requests to both public and private companies. Android is the most used operating system on mobile devices that is more affordable on a smartphone that owns this operating system compared to iOS.

For this reason, I have decided to create an Android application to facilitate consultation of our services and their prices and request appointments at our hairdresser.

What differences our application is that it was created with the intention of making it easier for our customers to contact us and that few companies request appointments through a mobile application. This gives comfort to our clients, which leads to higher income for the company.

This project tries to offer a reliable and stable application that requests an appointment without having to pay money online, the only information that the application will request will be: Telephone number and name of the client, name of the pet, weight, breed and finally the service that you want to request.

Over the document the technologies used will be presented, we can also find a user manual to understand the correct operation of the application.

Keywords

The following list contains key words to catalog the project.

- Hairdresser.
- Pet.
- Animals.
- Dog.
- Cat.
- Hygiene.
- Haircut.
- Bath.
- Cleanliness.
- Brushing.
- Drying.

Índice

1.	<i>Resumen</i>	2
2.	<i>Abstract</i>	3
3.	<i>Índice de ilustraciones</i>	6
4.	<i>Justificación del proyecto</i>	9
5.	<i>Introducción</i>	10
6.	<i>Objetivos</i>	11
6.1.	Objetivo principal	11
6.2.	Objetivos secundarios	11
7.	<i>Herramientas a utilizar</i>	11
8.	<i>Desarrollo</i>	13
8.1.	Organización	14
8.2.	Análisis de requerimiento de la aplicación	15
8.2.1.	Presupuesto	16
8.2.2.	Diagrama de Gantt	17
8.2.3.	Requerimientos	17
8.3.	Hardware, Sistema Operativo y aplicaciones	17
8.3.1.	Hardware	17
8.3.2.	Sistema Operativo	18
8.3.3.	Aplicaciones	18
8.4.	Red	20
8.5.	Base de datos	21
8.5.1.	Diagrama entidad – relación	21
8.5.2.	MySQL Workbench	21
8.5.3.	Modelo relacional	21
8.6.	Diseño de aplicación	22
8.6.1.	Diagrama de uso	22
8.6.2.	Diagrama de clases	22
8.7.	Programación	23
8.8.	Desarrollo de interface	30
8.9.	Sistema de gestión empresarial	31
8.10.	Documentación	31
8.10.1.	Manual de instalación	31
8.10.2.	Manual de usuario	32
8.10.3.	Manual de administrador	37
8.11.	Documentación en inglés	42
8.11.1.	Installation manual	42
8.11.2.	User's manual	42
8.11.3.	Administrator's manual	48
8.12.	Seguridad	52
8.13.	Entornos de desarrollo	53
9.	<i>Conclusiones</i>	54

10.	Anexo.....	55
11.	Bibliografía	56

3. Índice de ilustraciones

Ilustración 6-1 Sistema Operativo	18
Ilustración 6-2 Aplicaciones: Android Studio	18
Ilustración 6-3 Aplicaciones: Adobe Photoshop.....	19
Ilustración 6-4 Aplicaciones: Microsoft Office 2019.....	19
Ilustración 6-5 Aplicaciones: Firefox	19
Ilustración 6-6 Aplicaciones: Visual Studio Code.....	20
Ilustración 6-7 Aplicaciones: Unarchiver.....	20
Ilustración 6-8 Red: Topología en estrella.....	20
Ilustración 6-9 Base de datos: Diagrama Entidad - Relación	21
Ilustración 6-10 Base de datos: Modelo relacional	22
Ilustración 6-11 Diagrama de uso.....	22
Ilustración 6-12 Diagrama de clases.....	23
Ilustración 6-13 Programación: Inicio Sesión	24
Ilustración 6-14 Programación: Inicio Sesión 2	25
Ilustración 6-15 Programación: Registro	25
Ilustración 6-16 Programación: Registro 2	26
Ilustración 6-17 Programación: Registro 3	26
Ilustración 6-18 Programación: Registro 4	27
Ilustración 6-19 Programación: Información personal.....	27
Ilustración 6-20 Programación: Información personal 2	28
Ilustración 6-21 Programación: Información personal archivo java.....	28
Ilustración 6-22 Programación: Información personal lista	29
Ilustración 6-23 Programación: Información personal 3	29
Ilustración 6-24 Programación: Cita previa	30
Ilustración 6-25 Programación: Cita previa 2	30
Ilustración 6-26 Sistema de gestión empresarial: Odoo	31
Ilustración 6-27 Manual de usuario: Inicio sesión.....	32
Ilustración 6-28 Manual de usuario: Registro 2	32
Ilustración 6-29 Manual de usuario: Registro	32
Ilustración 6-30 Manual de usuario: Inicio sesión 2	33
Ilustración 6-31 Manual de usuario: Menú principal 2	33
Ilustración 6-32 Manual de usuario: Menú principal	33
Ilustración 6-33 Manual de usuario: Información personal 2	34
Ilustración 6-34 Manual de usuario: Información personal	34
Ilustración 6-35 Manual de usuario: Precios 2	34
Ilustración 6-36 Manual de usuario: Precios	34
Ilustración 6-37 Manual de usuario: Horario 2	35
Ilustración 6-38 Manual de usuario: Horario	35
Ilustración 6-39 Manual de usuario: Recogida a domicilio 2.....	35
Ilustración 6-40 Manual de usuario: Recogida a domicilio	35
Ilustración 6-41 Manual de usuario: Cita previa 2	36
Ilustración 6-42 Manual de usuario: Cita previa	36
Ilustración 6-43 Manual de usuario: Cita previa 4	36
Ilustración 6-44 Manual de usuario: Cita previa 3	36
Ilustración 6-45 Manual de usuario: Información 2	37
Ilustración 6-46 Manual de usuario: Información	37
Ilustración 6-47 Manual de administrador: Desconexión VPN	37

Ilustración 6-48 Manual de administrador: Conexión VPN	38
Ilustración 6-49 Manual de administrador: Conexión MySQL	38
Ilustración 6-50 Manual de administrador: Conexión establecida con la base de datos	39
Ilustración 6-51 Manual de administrador: Tablas de la base de datos.....	39
Ilustración 6-52 Manual de administrador: Consulta de datos de una tabla	39
Ilustración 6-53 Manual de administrador: Datos de la tabla.....	40
Ilustración 6-54 Manual de administrador: Botones de edición de las tablas	40
Ilustración 6-55 Manual de administrador: Edición de registros	40
Ilustración 6-56 Manual de administrador: Edición de registros 1	40
Ilustración 6-57 Manual de administrador: Añadir registros	41
Ilustración 6-58 Manual de administrador: Añadir registros 1	41
Ilustración 6-59 Manual de administrador: Eliminar registros.....	41
Ilustración 6-60 Manual de administrador: Eliminar registros 1.....	42
Ilustración 6-61 User Manual: Login	42
Ilustración 6-62 User Manual: Registration 2.....	43
Ilustración 6-63 User Manual: Registration.....	43
Ilustración 6-64 User Manual: Login 2	43
Ilustración 6-65 User Manual: Main Menu 2	44
Ilustración 6-66 User Manual: Main Menu	44
Ilustración 6-67 User Manual: Personal information 2	44
Ilustración 6-68 User Manual: Personal information	44
Ilustración 6-69 User Manual: Prices 2.....	45
Ilustración 6-70 User Manual: Prices	45
Ilustración 6-71 User Manual: Schedule 2	45
Ilustración 6-72 User Manual: Schedule	45
Ilustración 6-73 User Manual: Home collection 2	46
Ilustración 6-74 User Manual: Home collection.....	46
Ilustración 6-75 User Manual: Appointment 2.....	46
Ilustración 6-76 User Manual: Appointment.....	46
Ilustración 6-77 User Manual: Appointment 4.....	47
Ilustración 6-78 User Manual: Appointment 3.....	47
Ilustración 6-79 User Manual: Information 2.....	47
Ilustración 6-80 User Manual: Information.....	47
Ilustración 6-81 Administrator Manual: VPN disconnection.....	48
Ilustración 6-82 Administrator Manual: VPN Connection	48
Ilustración 6-83 Administrator Manual: MySQL connection	49
Ilustración 6-84 Administrator Manual: Connection established with the database	49
Ilustración 6-85 Administrator Manual: Database Tables.....	49
Ilustración 6-86 Administrator Manual: Querying data from a table.....	50
Ilustración 6-87 Administrator Manual: Table Data	50
Ilustración 6-88 Administrator Manual: Buttons for editing tables	50
Ilustración 6-89 Administrator Manual: Editing Records	51
Ilustración 6-90 Administrator Manual: Editing Records 1	51
Ilustración 6-91 Administrator Manual: Adding Records	51
Ilustración 6-92 Administrator Manual: Adding Records 1	51
Ilustración 6-93 Administrator Manual: Delete Records.....	52
Ilustración 6-94 Administrator Manual: Delete Records 1.....	52
Ilustración 6-95 Entornos de desarrollo: Android Studio	53
Ilustración 6-96 Entornos de desarrollo: Visual Studio Code	54

4. Justificación del proyecto

El motivo por el que elegí el tema de realizar una aplicación para una peluquería canina se debe a que la mayoría de las personas tenemos un animal de compañía. Según datos oficiales hay unos 13 millones de mascotas registradas en la Red Española de Identificación de Animales de Compañía (REIAC), es decir que por familia poseemos una media de 2 animales de compañía ya sean perros o gatos que son los animales más comunes. En ellos, cada familia invierte unos 800€ anuales sin tener en cuenta la higiene y vacunación de los mismos, si lo incluimos llegarían a invertir 1200€ al año aproximadamente.

El presente proyecto se enfoca en la higiene de las mascotas, para ello muchas familias acuden a centros especializados para ello, como puede ser Requeteguaupos, en esto se basa mi proyecto, en un salón de belleza para mascotas cuyas actividades están dedicadas al cuidado de la imagen de las mascotas. Se ofrecerán servicios de baño, desparasitación, cepillado, secado o corte de pelo entre otros servicios, para nuestras mascotas que beneficia a todas las personas que tengan un animal de compañía y así facilitarle la higiene de los mismos.

Para facilitar y ayudar a que los clientes se beneficien de realizar consultas de precios, horarios, recogida a domicilio, citas previa e información sobre la peluquería hemos creado una aplicación para Requeteguaupos.

Decimos crear una aplicación móvil, ya que en pleno siglo XXI generalmente a partir de los 15 años disponemos de un dispositivo móvil a nuestro alcance.

En este último año las marcas comerciales de dispositivos Android mas vendidos han sido Xiaomi, Samsung, Huawei u Oppo entre otros, en ellos podríamos instalar nuestra aplicación para facilitar a nuestros clientes dicha información mencionada anteriormente y así no se tendrán que desplazar para poder informarse o solicitar una cita previa en las instalaciones de Requeteguaupos.

5. Introducción

Hace unos años atrás tener una mascota no era muy común, pero se ha incrementado el número de personas que deciden tener un animal de compañía en casa.

Actualmente el cuidado de las mascotas domésticas nos preocupa a todos los dueños. El nuevo estilo de vida de las personas deja poco tiempo para que los dueños cuiden a sus mascotas por lo que la mayoría los bañan y cortan el pelo en centros de belleza de mascotas, además hay razas de perro como podrían ser caniches que requieren un cuidado especial o una atención más frecuente.

El perfil actual de estos establecimientos se va diversificando, se trata de personas que, por diversos motivos, principalmente por la escasez de tiempo del que disponen, comodidad y las necesidades específicas de sus mascotas como el tamaño, raza, etc. Acuden a una peluquería canina para mantener así la imagen e higiene de sus mascotas.

Cabe destacar que cada vez es más frecuente que se acuda a un centro para mantener la estética cuidada del mismo y no sólo tener en cuenta el cuidado que conlleva su salud.

El número de mascotas que hay en España se estima que irá aumentando en los próximos años, por ello cada vez tenemos más peluquerías para ellas. De esta manera aumentaría también la competencia de Requeteguaupos, por ello, se decidió poner a disposición de los clientes esta nueva aplicación móvil, para que los clientes, tengas más facilidades a la hora de hacer consultas, pedir citas u obtener información sobre nosotros en Requeteguaupos

6. Objetivos

6.1. **Objetivo principal.**

El objetivo principal de este proyecto es desarrollar una aplicación para el sistema operativo Android para la consulta de precios, horarios, reservas de cita previa, información sobre la pulquería canina y registro de usuarios para almacenar sus datos, con el fin de aumentar las ganancias de Requeteguaupos.

6.2. **Objetivos secundarios.**

- Diseño y desarrollo de una interfaz de usuario atractiva visualmente y fácil de usar para todas las personas que tenga acceso a ella.
- Diseño y desarrollo de una base de datos para los datos necesarios de la aplicación.

7. Herramientas a utilizar.

Para realizar la aplicación usaremos las siguientes herramientas:

- **Lenguajes de programación:**
 - Java: para la programación de la aplicación en Android.
 - PHP: para comunicarme y poder introducir y sacar datos a la base de datos.
- **Bases de datos:**
 - MySQL WorkBench: para gestionar los datos de la base de datos.
- **Servidor Web:**
 - Apache: para que la aplicación se comunique con la base de datos.

Java

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Fue creado por Sun Microsystems y actualmente es de Oracle.

Es un lenguaje orientado a objetos y la intención es permitir que los desarrolladores de aplicaciones escriban sola una vez el proyecto y lo puedan ejecutar en cualquier dispositivo.

Características de Java que se han tenido en cuenta:

- **Es simple.**

Java ofrece la funcionalidad de un lenguaje potente, derivado de C y C++, pero sin las características menos usadas y más confusas de estos, haciéndolo más sencillo.

- **Orientado a objetos.**

El enfoque orientado a objetos (OO) es uno de los estilos de programación más populares. Permite diseñar el software de forma que los distintos tipos de datos que se usen estén unidos a sus operaciones.

- **Es distribuido.**

Java proporciona una gran biblioteca estándar y herramientas para que los programas puedan ser distribuidos.

- **Independiente a la plataforma.**

Esto significa que programas escritos en el lenguaje Java pueden ejecutarse en cualquier tipo de hardware, lo que lo hace portable.

- **Es el leguaje usado para Android Studio.**

Java es un leguaje usado para programar en Android ya que puedes desarrollar sencillas aplicaciones u otras más complejas, también potentes aplicaciones web y todas pueden tener acceso a bases de datos.

PHP:

PHP, acrónimo recursivo en inglés de PHP: Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el preprocesado de texto plano en UTF-8. Posteriormente se aplicó al desarrollo web de contenido dinámico, dando un paso evolutivo en el concepto de aplicación en línea, por su carácter de servicio.

Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el texto plano, dando como resultado, en los exploradores, una salida al usuario perfectamente entendible.

Características PHP que se han tenido en cuenta:

- Es de propósito general. Aunque está especializado en la programación de sitios web dinámicos, se puede usar PHP para acceder y manejar una base de datos, generar cálculos aritméticos (con scripts), comunicarse entre un servidor y un cliente, etc.

- Soporte para base de datos, específicamente con MySQL. También tiene un buen soporte con PostgreSQL, SQLite y Oracle. El acceso se hace mediante librerías o extensiones ya configuradas, que ayudan en la escritura del código.
- Es un lenguaje interpretado, es decir, no se compila. Además, el código se almacena en el servidor.
- Es multiplataforma. Podemos tener un servidor de PHP en multitud de sistemas operativos, como Linux, Mac OS X, Windows.
- Es de código libre, así que hay un buen soporte, además de mucha información por internet.
- Tiene una amplia documentación oficial, y no oficial como en GitHub o StackOverflow.

MySQL Workbench:

MySQL Workbench es un software creado por Sun Microsystems, esta herramienta permite modelar diagramas de Entidad-Relación para bases de datos MySQL.

Con esta herramienta se puede elaborar una representación visual de las tablas, vistas, procedimientos almacenados y claves foráneas de la base de datos. Además, es capaz de sincronizar el modelo en desarrollo con la base de datos real.

Características MySQL Workbench que se han tenido en cuenta:

- Permite diseñar visualmente, modelar, gestionar y administrar bases de datos.
- Es multiplataforma. Podemos usarlo en cualquier sistema operativo como Linux, Mac OS o Windows.

Apache:

Apache es un software especializado en ofrecer servicios de servidor web. Es versátil, ligero y muy útil, además de ser completamente gratuito y de código abierto. Su popularidad es tal que, actualmente, cerca del 50% de las páginas web de todo el mundo se ejecutan en un servidor de este tipo.

Características Apache que se han tenido en cuenta:

- Es gratuito y de código abierto.
- Transfiere los datos alojados en una base de datos con una petición de una persona.
- Soporta diferentes lenguajes.

8. Desarrollo

8.1. Organización.

Empresa:

La empresa a la que se destina la aplicación móvil que estamos creando es Requeteguaupos. Forma parte del sector servicios a mascotas, más detalladamente es una peluquería canina, presta servicios como podría ser lavado, deslanado, corte de pelo, etc... además presta un servicio de recogida a domicilio que cada vez es más común en este sector.

Empresario:

La persona que emprendió es Francisco Martín, la empresa fue fundada en el año 2010.

Francisco Martín, es una persona apta para emprender ya que siempre que consigue una meta busca otra nueva, decidió emprender ya que en la zona no se encontraba ninguna empresa que aportase este mismo servicio, además es una persona amante de los animales lo cual es imprescindible para este negocio junto con la experiencia que posee de anteriormente haber trabajado en empresas de este mismo sector.

También podría considerarse una persona dinámica ya que interactúa con la empresa tanto dentro como fuera de ella.

Como ya hemos mencionado, Requeteguaupos es una empresa prestadora de servicios a canes, es una sociedad limitada al capital aportado, fundada en Marzo del 2010.

Trabajadores:

La empresa consta de cinco trabajadores, de los cuales dos se encargan de recoger a las mascotas en su domicilio y al finalizar el servicio llevarlas de nuevo a su hogar, otros dos trabajadores prestarían servicios en las instalaciones de Requeteguaupos, y por último Francisco Martín, el empresario el cual se encargará del departamento de RRHH, finanzas y marketing de la empresa entre otros cargos.

Ubicación:

Las instalaciones de Requeteguaupos se encuentran en la calle Ronda Arco de Ugena, Nº10, 45200 Illescas, Toledo.

A pesar de no disponer de Parking privado, hay suficiente espacio como para que aparcar sea bastante sencillo. Además, acceder a sus instalaciones es muy fácil.

Se puede acceder a Requeteguaupos, de varias formas sencillas ya que esta en el centro de Illescas, así como: A pie, coche propio, taxi o incluso autobús.

Listado de servicios:

Servicio de Baño:

Baño con los mejores productos. En caso de solicitar Baño Sanitario, este se hace con productos que repelen pulgas y garrapatas. Además, el servicio incluye limpieza externa de orejas, drenaje de glándulas anales, (en caso de ser necesario) corte de uñas de ser necesario y cepillado en aire caliente (Brushing).

Servicio de Corte de Pelo Estético:

Corte de pelo correspondiente a cada raza, limpieza externa de orejas, corte de uñas en caso de ser necesario, peinado y cepillado.

Servicio de Corte a Máquina:

Corte de pelo a máquina, teniendo en cuenta el largo necesario para la protección de la piel.

Servicio de Corte a Tijera:

Corte a tijera, generalmente para perros de pelo largo.

Servicio de Deslanado:

Deslanado, técnica que consiste en retirar el subpelo muerto con el fin de que tu mascota no pase calor.

Servicio Completo de Baño y Corte:

Baño sanitario (en caso de solicitarlo), corte de pelo estético, limpieza externa de orejas, drenaje de glándulas anales, corte de uñas en caso de ser necesario, cepillado en aire caliente (Brushing).

Finalidad de la empresa:

El objetivo principal de Requeteguaupos es tener clientes fieles, para así obtener beneficios y sacar adelante el negocio, además de esta manera, poder agrandar la peluquería y meter más servicios para ellos, incluso implantar productos y accesorios para animales.

La empresa se creó con la intención de satisfacer las necesidades que los clientes tienen de sus mascotas, y evitar que se tengan que desplazar hasta otros pueblos para obtener este servicio.

8.2. Análisis de requerimiento de la aplicación.

8.2.1. Presupuesto

Gastos operacionales

Gasto de operación	Precio por hora	Horas diarias	Total día
Gerente de proyecto	20€	6	120€
Analista	17€	8	136€
Diseñador	14€	8	112€
Desarrollador	15€	8	120€

Totales

Total días trabajados	Total horas trabajadas
21	630

Sueldos empleados

Sueldos empleados	Salario día	Días trabajados	Salario mensual
Gerente	120	21	2.520€
Analista	136	21	2.856€
Diseñador	112	21	2.352€
Desarrollador	120	21	2.520€

Licencias

Licencia	Precio
Mac Os	Gratis
Android Studio	Gratis
Office	7€ mensuales
Servidor	60€ mensuales

Total presupuesto

Total Ingresos (Precio venta aplicación)	Total gastos	Total beneficios
15.000€	10.315€	4.685€

8.2.2. Diagrama de Gantt

Anexo 1: Diagrama de Gantt

8.2.3. Requerimientos

La aplicación que vamos a realizar es una app móvil de una peluquería canina para los dispositivos Android, la aplicación constará de distintos apartados además de una base de datos que almacenará una tabla de datos de usuario y otra con datos de las mascotas.

Los apartados serán los siguientes:

La aplicación tendrá un registro para los usuarios que quieran tener sus datos almacenados como su nombre, teléfono, nombre de su mascota, raza de la mascota, etc.

Consta de un apartado el cual mostrará los datos almacenados en la base de datos del usuario logueado, también tendrá un apartado de horario, otro de precios y el apartado de recogida a domicilio.

También constará con el apartado para realizar citas previas, en el cual rellenarás la información solicitada y se enviará por correo electrónico para que los dueños del local se pongan en contacto con el usuario y así, concretar una cita. Por último, podrás encontrar un apartado el cual contendrá información sobre la peluquería, como la experiencia en este sector, el ambiente o la calidad.

El software mínimo del dispositivo para tener la aplicación sería de Android 5.0 o superior, además de tener como mínimo una conexión a internet y una memoria RAM de 512 MB.

8.3. Hardware, Sistema Operativo y aplicaciones.

A lo largo del proyecto se ha utilizado un equipo para poder llevarlo a cabo, como un ordenador en este caso.

8.3.1. Hardware

- Ordenador:
Macbook Air (Principios de 2015).
- Procesador:
Intel Core i5 (I5-5250) a 1.6 Ghz.
- Memoria Ram:
4 Gb LPDDR3 integrada
- Pantalla:

- 13,3 pulgadas
- Resolución pantalla:
1440 x 900
- Gráficos:
Intel HD Graphics 6000
- Almacenamiento:
Crucial P1 CT1000P1SSD8 1TB
- Sistema operativo:
macOS Catalina

Con este dispositivo se hará la guía de documentación, además del diseño de la aplicación móvil y testeo de la misma.

8.3.2. Sistema Operativo

- **MacOs Catalina.**

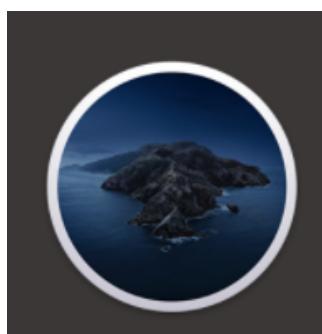


Ilustración 8-1 Sistema Operativo

Este es la decimosexta versión principal del macOS, es decir el sistema escritorio de Apple. Es el sucesor de macOS Mojave, el cual se anunció en junio de 2019 y se lanzó en octubre del 2019, una de las características a destacar es que macOS Catalina solo admite aplicaciones en 64 bits.

8.3.3. Aplicaciones

- **Android Studio 3.6.3**



Ilustración 8-2 Aplicaciones: Android Studio

Es el IDE oficial de Android que fue creado exclusivamente con el fin de acelerar el desarrollo y la ayuda de compilar apps de la más alta calidad para todos los dispositivos Android. Lo usaremos para la creación de la aplicación y probar la app en su propio emulador.

- **Adobe Photoshop 2019**



Ilustración 8-3 Aplicaciones: Adobe Photoshop

Es una herramienta de edición de imágenes y fotografía, se usa para retoques de fotos y para hacer montajes de carácter profesional. Con el podemos realizar cualquier diseño de logotipos o de imágenes para la aplicación.

- Microsoft Office 2019



Ilustración 8-4 Aplicaciones: Microsoft Office 2019

Es una herramienta que nos permite crear, acceder y compartir documentos de ofimática. Lo usaremos para hacer el desarrollo de la documentación.

- **Firefox**

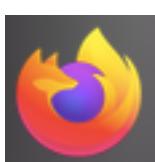


Ilustración 8-5 Aplicaciones: Firefox

Es un navegador web libre, el cual usaremos para buscar información.

- **Visual Studio Code**

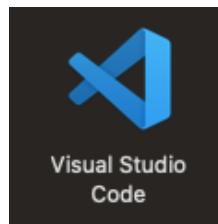


Ilustración 8-6 Aplicaciones: Visual Studio Code

Editor de código fuente desarrollado por Microsoft el cual se puede utilizar en Windows, MacOs y Linux. Incluye depuración de código integrado, control de Git, resultado de sintaxis, etc. Lo usaremos para poder crear y editar ficheros php.

- **Unarchiver**



Ilustración 8-7 Aplicaciones: Unarchiver

Es un programa pequeño y fácil de usar que permite comprimir y descomprimir un gran número de tipos de paquetes.

8.4. Red

Para la realización del proyecto, hemos decidido usar una topología de red en estrella ya que todas las personas que llevan a cabo el proyecto estamos en un mismo local, para ello todos los ordenadores utilizados para crear la aplicación están conectadas a un switch para poder estar en la misma red y estar conectados entre si.

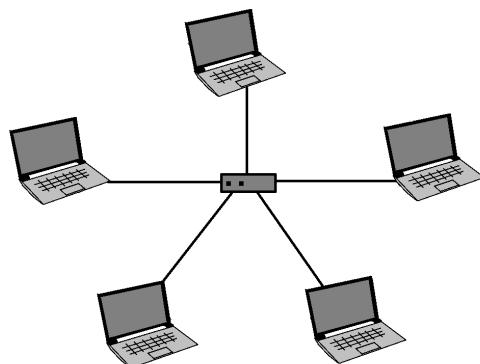


Ilustración 8-8 Red: Topología en estrella

En esta imagen vemos en lo que consiste la topología de estrella, la cual es la que hemos usado para la realización del proyecto.

8.5. Base de datos.

La base de datos que voy a usar en mi aplicación desde mi punto de vista es mejor que tenga únicamente dos tablas y que estén relacionadas entre si y no cuatro como se indica en la guía.

Ya que veo más cómodo actualizar la aplicación por si fuese necesario cambiar los precios u horarios del establecimiento.

8.5.1. Diagrama entidad – relación.

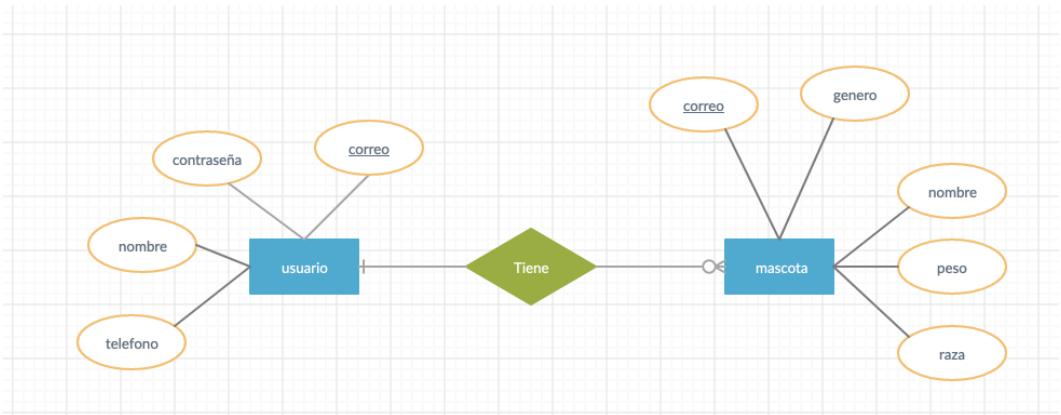


Ilustración 8-9 Base de datos: Diagrama Entidad - Relación

En esta imagen podemos ver como los usuarios están relacionados con la mascota mediante su correo, ya que al registrarse el usuario por un lado se almacenan los datos del usuario y por otro lado los datos de su mascota.

8.5.2. MySQL Workbench

Es una herramienta visual de diseño de las bases de datos que integran el desarrollo de software, administración de la base de datos, diseño, gestión y mantenimiento de la base de datos.

Esta herramienta es la que usarán los administradores para poder consultar, modificar, eliminar, etc. Los datos de la base de datos.

8.5.3. Modelo relacional

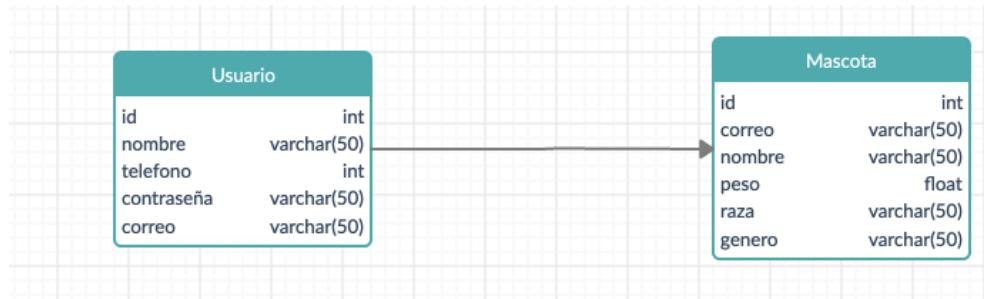


Ilustración 8-10 Base de datos: Modelo relacional

Aquí como podemos observar vemos que usuario esta relacionado con mascota para así poder obtener datos de ellas.

8.6. Diseño de aplicación.

8.6.1. Diagrama de uso:

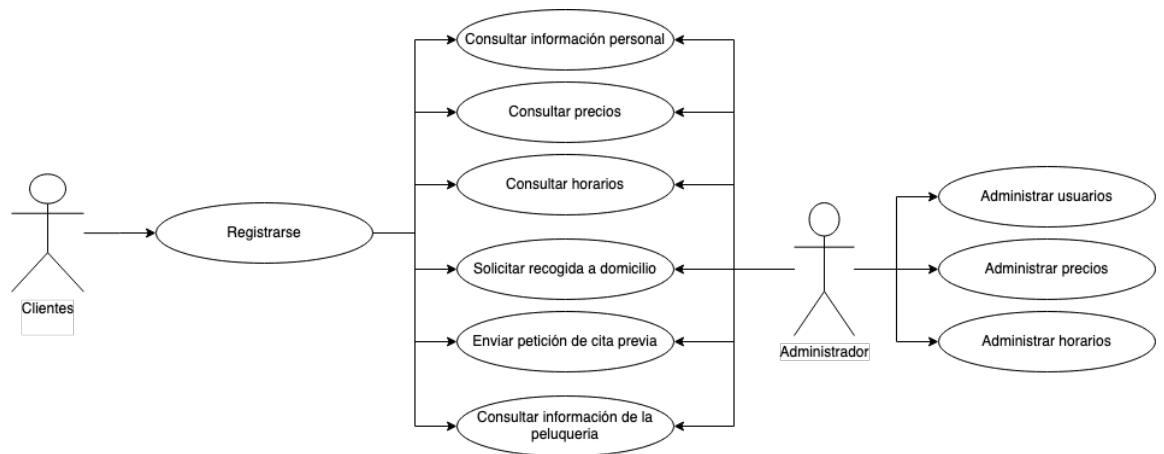


Ilustración 8-11 Diagrama de uso

En esta imagen se explica el uso que puede realizar tanto los clientes como los administradores.

Los clientes tendrán que registrarse, tras su registro podrán consultar y enviar las peticiones deseadas.

El administrador podrá administrar los usuarios, además de poder consultar y enviar las peticiones deseadas.

Los precios y horarios los podrán administrar en Android Studio y para guardar los cambios tendrán que hacer una nueva actualización de la aplicación.

8.6.2. Diagrama de clases:

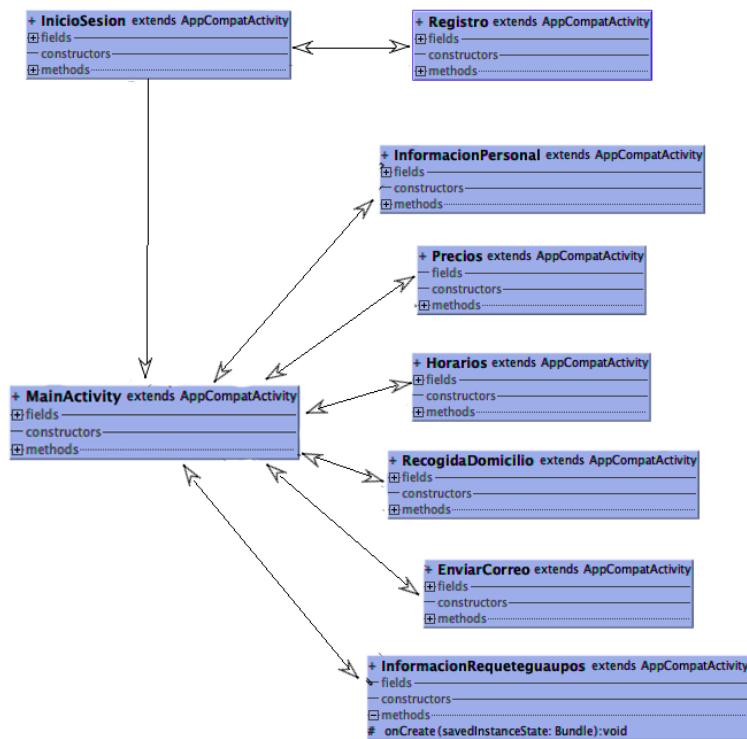
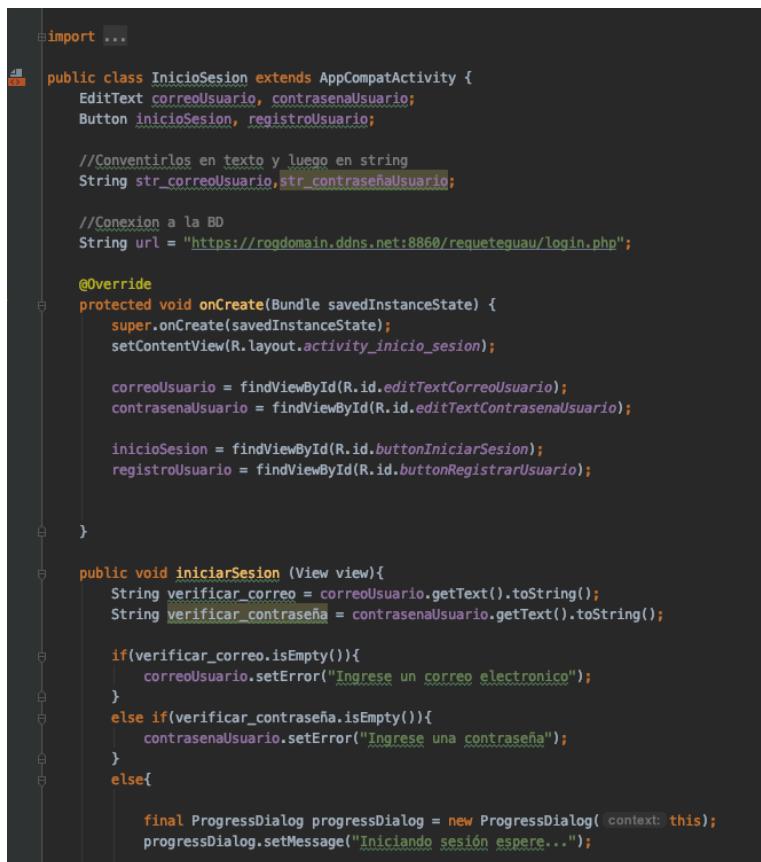


Ilustración 8-12 Diagrama de clases

8.7. Programación.

A mi parecer, explicaré el código más importante y que, probablemente necesita una explicación para entenderlo.

Código inicio sesión



```
import ...  
  
public class InicioSesion extends AppCompatActivity {  
    EditText correoUsuario, contrasenaUsuario;  
    Button inicioSesion, registroUsuario;  
  
    //Convertirlos en texto y luego en string  
    String str_correoUsuario, str_contrasenaUsuario;  
  
    //Conexion a la BD  
    String url = "https://rogdomain.ddns.net:8860/regueteguau/login.php";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_inicio_sesion);  
  
        correoUsuario = findViewById(R.id.editTextCorreoUsuario);  
        contrasenaUsuario = findViewById(R.id.editTextContrasenaUsuario);  
  
        inicioSesion = findViewById(R.id.buttonIniciarSesion);  
        registroUsuario = findViewById(R.id.buttonRegistrarUsuario);  
  
    }  
  
    public void iniciarSesion (View view){  
        String verificar_correo = correoUsuario.getText().toString();  
        String verificar_contraseña = contrasenaUsuario.getText().toString();  
  
        if(verificar_correo.isEmpty()){  
            correoUsuario.setError("Ingrese un correo electrónico");  
        }  
        else if(verificar_contraseña.isEmpty()){  
            contrasenaUsuario.setError("Ingrese una contraseña");  
        }  
        else{  
  
            final ProgressDialog progressDialog = new ProgressDialog( context: this);  
            progressDialog.setMessage("Iniciando sesión espere...");  
        }  
    }  
}
```

Ilustración 8-13 Programación: Inicio Sesión

En esta imagen podemos ver como conectamos con la base de datos a través del php. Más abajo observamos que cuando iniciamos sesión, almacenamos las variables con las que ha introducido el usuario, si están vacías saldrán un mensaje de error y si está bien saldrá un mensaje que está iniciando sesión.

```

        str_correoUsuario = correoUsuario.getText().toString().trim();
        str_contraseñaUsuario = contraseñaUsuario.getText().toString().trim();

        StringRequest request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                progressDialog.dismiss();

                if(response.equalsIgnoreCase("Sesion Iniciada")){
                    correoUsuario.setText("");
                    contraseñaUsuario.setText("");

                    Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                    intent.putExtra( name: "str_correoUsuario", str_correoUsuario);

                    startActivity(intent);
                    Toast.makeText( context: InicioSesion.this, response, Toast.LENGTH_SHORT).show();
                }
                else{
                    correoUsuario.setError("Correo electrónico erroneo");
                }
            }
        },new Response.ErrorListener(){

            @Override
            public void onErrorResponse(VolleyError error) {
                progressDialog.dismiss();
                Toast.makeText( context: InicioSesion.this, error.getMessage().toString(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

Ilustración 8-14 Programación: Inicio Sesión 2

Para que inicie sesión, comprobará el correo y el usuario, si esta bien iniciará sesión y abrirá el menú principal, si no, dará error y no podremos acceder.

Registro

```

import ...

public class Registro extends AppCompatActivity {

    EditText nomUsuario, correoUsuario, telefonoUsuario, contraseñaUsuario, nombreMascota, pesoMascota, razaMascota, generoMascota;
    Button registrar;

    //Convertirlos en texto y luego en string
    String str_nomUsuario, str_correoUsuario, str_telefonoUsuario, str_contraseñaUsuario, str_generoMascota, str_razaMascota, str_nombreMascota, str_pesoMascota;

    //Conexion a la BD
    String url = "https://rogdomain.ddns.net:8860/requeteguau/register.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registro);

        nomUsuario = findViewById(R.id.editTextNomUsuario);
        correoUsuario = findViewById(R.id.editTextCorreoElec);
        telefonoUsuario = findViewById(R.id.editTextTelefono);
        contraseñaUsuario = findViewById(R.id.editTextContraseña);
        nombreMascota = findViewById(R.id.editTextNomMascota);
        pesoMascota = findViewById(R.id.editTextPesoMascota);
        razaMascota = findViewById(R.id.editTextRazaMascota);
        generoMascota = findViewById(R.id.editTextGeneroMascota);

        registrar = findViewById(R.id.buttonRegistrarme);
    }

    public void Registro (View view){
        ...
    }
}

```

Ilustración 8-15 Programación: Registro

Como en el anterior código lo primero que haremos es recoger todos los datos y crear la conexión al php para que acceda a la base de datos.
Luego enlazaremos los edit text para recuperar los datos.

```

//Cuando le damos al boton sale una ventana para que esperemos
final ProgressDialog progressDialog = new ProgressDialog( context: this );
progressDialog.setMessage("Por favor espere...");
//-----
String verificar_nombre = nomUsuario.getText().toString();
String verificar_correo = correoUsuario.getText().toString();
String verificar_telefono = telefonoUsuario.getText().toString();
String verificar_contrasena = contrasenaUsuario.getText().toString();
String verificar_nombremascota = nombreMascota.getText().toString();
String verificar_pesomascota = pesoMascota.getText().toString();
String verificar_razamascota = razaMascota.getText().toString();
String verificar_generomascota = generoMascota.getText().toString();

if(verificar_nombre.isEmpty()){
    nomUsuario.setError("Ingrese un nombre de usuario");
}
else if(verificar_correo.isEmpty()){
    correoUsuario.setError("Ingrese un correo electronico");
}
else if(verificar_telefono.isEmpty()){
    telefonoUsuario.setError("Ingrese su telefono");
}
else if(verificar_contrasena.isEmpty()){
    contrasenaUsuario.setError("Ingrese una contraseña");
}
else if(verificar_nombremascota.isEmpty()){
    nombreMascota.setError("Ingrese el nombre de su mascota");
}
else if(verificar_pesomascota.isEmpty()){
    pesoMascota.setError("Ingrese el peso de su mascota");
}
else if(verificar_razamascota.isEmpty()){
    razaMascota.setError("Ingrese la raza de su mascota");
}
else if(verificar_generomascota.isEmpty()){
    generoMascota.setError("Ingrese el genero de su mascota");
}
else{

```

Ilustración 8-16 Programación: Registro 2

Aquí como podemos ver recogeremos los valores introducidos y comprobaremos que no estén vacíos ,si no, saldrá un mensaje para que introduzca el campo vacío.

```

        }
    else{

        progressDialog.show();
        str_nomUsuario = nomUsuario.getText().toString().trim();
        str_correoUsuario = correoUsuario.getText().toString().trim();
        str_telefonoUsuario = telefonoUsuario.getText().toString().trim();
        str_contrasenaUsuario = contrasenaUsuario.getText().toString().trim();
        str_generoMascota = generoMascota.getText().toString().trim();
        str_razaMascota = razaMascota.getText().toString().trim();
        str_pesomascota = pesoMascota.getText().toString().trim();
        str_nombreMascota = nombreMascota.getText().toString().trim();
    }

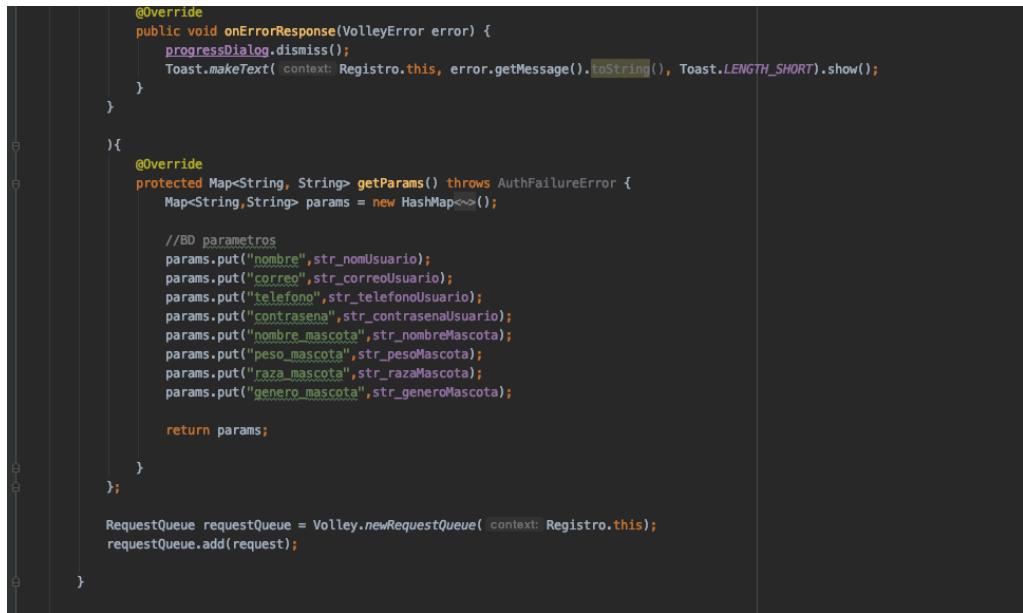
    StringRequest request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            progressDialog.dismiss();
            nomUsuario.setText("");
            correoUsuario.setText("");
            telefonoUsuario.setText("");
            contrasenaUsuario.setText("");
            nombreMascota.setText("");
            pesoMascota.setText("");
            razaMascota.setText("");
            generoMascota.setText("");

            if(response.equalsIgnoreCase("Registro guardado")){
                startActivity(new Intent(getApplicationContext(), InicioSesion.class));
                Toast.makeText(context: Registro.this, response, Toast.LENGTH_SHORT).show();
            }
            else{
                Toast.makeText(context: Registro.this, response, Toast.LENGTH_SHORT).show();
            }
        }
    });

```

Ilustración 8-17 Programación: Registro 3

Ahora recogeremos lo introducido por el usuario y lo guardaremos en la variable como se llama en la base de datos, si todo sale bien, nos deberá salir un mensaje de registro guardado y nos llevará a la pantalla de inicio de sesión.



```

@Override
public void onErrorResponse(VolleyError error) {
    progressDialog.dismiss();
    Toast.makeText(context, Registro.this, error.getMessage().toString(), Toast.LENGTH_SHORT).show();
}

@Override
protected Map<String, String> getParams() throws AuthFailureError {
    Map<String, String> params = new HashMap<>();

    //BD parametros
    params.put("nombre", str_nombreUsuario);
    params.put("correo", str_correoUsuario);
    params.put("telefono", str_telefonoUsuario);
    params.put("contrasena", str_contrasenaUsuario);
    params.put("nombre_mascota", str_nombreMascota);
    params.put("peso_mascota", str_pesoMascota);
    params.put("raza_mascota", str_razaMascota);
    params.put("genero_mascota", str_generoMascota);

    return params;
}

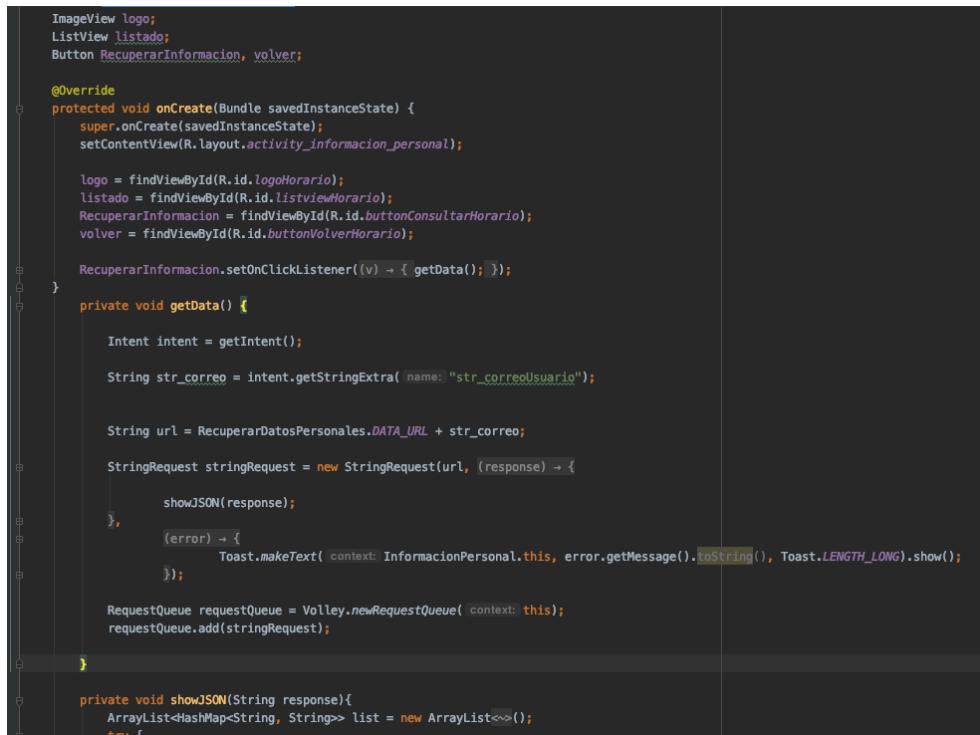
RequestQueue requestQueue = Volley.newRequestQueue(context: Registro.this);
requestQueue.add(request);
}
}

```

Ilustración 8-18 Programación: Registro 4

Por último, conectará con la base de datos y le añadirá los parámetros introducidos por el usuario del registro.

Información personal



```

ImageView logo;
ListView listado;
Button RecuperarInformacion, volver;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_informacion_personal);

    logo = findViewById(R.id.logoHorario);
    listado = findViewById(R.id.listviewHorario);
    RecuperarInformacion = findViewById(R.id.buttonConsultarHorario);
    volver = findViewById(R.id.buttonVolverHorario);

    RecuperarInformacion.setOnClickListener(v -> { getData(); });
}

private void getData() {
    Intent intent = getIntent();

    String str_correo = intent.getStringExtra(name: "str_correoUsuario");

    String url = RecuperarDatosPersonales.DATA_URL + str_correo;

    StringRequest stringRequest = new StringRequest(url, (response) -> {
        showJSON(response);
    }, (error) -> {
        Toast.makeText(context: InformacionPersonal.this, error.getMessage().toString(), Toast.LENGTH_LONG).show();
    });

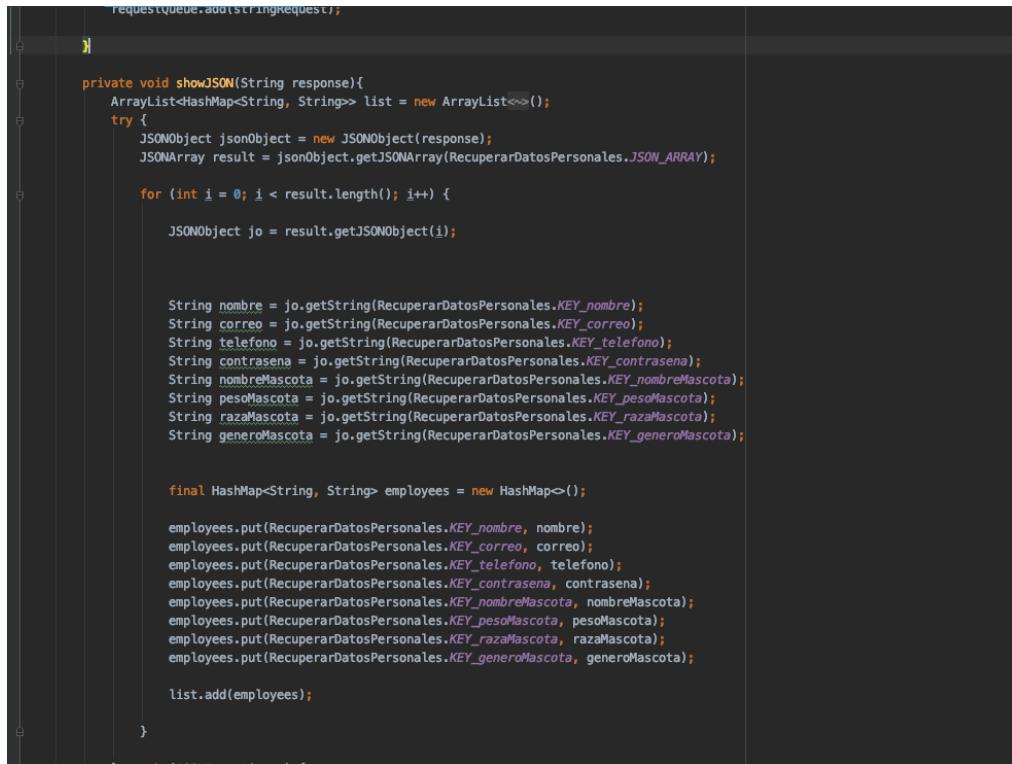
    RequestQueue requestQueue = Volley.newRequestQueue(context: this);
    requestQueue.add(stringRequest);
}

private void showJSON(String response){
    ArrayList<HashMap<String, String>> list = new ArrayList<>();
    try {

```

Ilustración 8-19 Programación: Información personal

Como podemos ver aquí, lo primero que haremos es enlazar las cosas con los botones, listas, imágenes, etc... Después, recogeremos el correo almacenado anteriormente que ha introducido en el inicio sesión para poder obtener su información personal.



```

    requestQueue.add(jsonObjectRequest);
}

private void showJSON(String response){
    ArrayList<HashMap<String, String>> list = new ArrayList<>();
    try {
        JSONObject jsonObject = new JSONObject(response);
        JSONArray result = jsonObject.getJSONArray(RecuperarDatosPersonales.JSON_ARRAY);

        for (int i = 0; i < result.length(); i++) {
            JSONObject jo = result.getJSONObject(i);

            String nombre = jo.getString(RecuperarDatosPersonales.KEY_nombre);
            String correo = jo.getString(RecuperarDatosPersonales.KEY_correo);
            String telefono = jo.getString(RecuperarDatosPersonales.KEY_telefono);
            String contrasena = jo.getString(RecuperarDatosPersonales.KEY_contrasena);
            String nombreMascota = jo.getString(RecuperarDatosPersonales.KEY_nombreMascota);
            String pesoMascota = jo.getString(RecuperarDatosPersonales.KEY_pesoMascota);
            String razaMascota = jo.getString(RecuperarDatosPersonales.KEY_razaMascota);
            String generoMascota = jo.getString(RecuperarDatosPersonales.KEY_generoMascota);

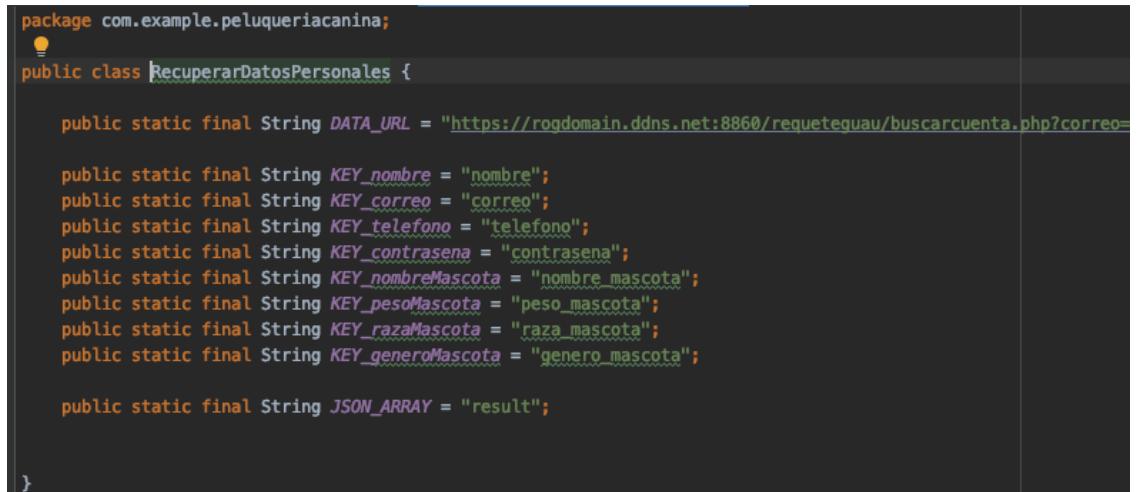
            final HashMap<String, String> employees = new HashMap<>();

            employees.put(RecuperarDatosPersonales.KEY_nombre, nombre);
            employees.put(RecuperarDatosPersonales.KEY_correo, correo);
            employees.put(RecuperarDatosPersonales.KEY_telefono, telefono);
            employees.put(RecuperarDatosPersonales.KEY_contrasena, contrasena);
            employees.put(RecuperarDatosPersonales.KEY_nombreMascota, nombreMascota);
            employees.put(RecuperarDatosPersonales.KEY_pesoMascota, pesoMascota);
            employees.put(RecuperarDatosPersonales.KEY_razaMascota, razaMascota);
            employees.put(RecuperarDatosPersonales.KEY_generoMascota, generoMascota);

            list.add(employees);
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

```

Ilustración 8-20 Programación: Información personal 2



```

package com.example.peluqueriacanina;

public class RecuperarDatosPersonales {

    public static final String DATA_URL = "https://rogdomain.ddns.net:8860/requetegau/buscarcuenta.php?correo=";

    public static final String KEY_nombre = "nombre";
    public static final String KEY_correo = "correo";
    public static final String KEY_telefono = "telefono";
    public static final String KEY_contrasena = "contrasena";
    public static final String KEY_nombreMascota = "nombre_mascota";
    public static final String KEY_pesoMascota = "peso_mascota";
    public static final String KEY_razaMascota = "raza_mascota";
    public static final String KEY_generoMascota = "genero_mascota";

    public static final String JSON_ARRAY = "result";

}

```

Ilustración 8-21 Programación: Información personal archivo java

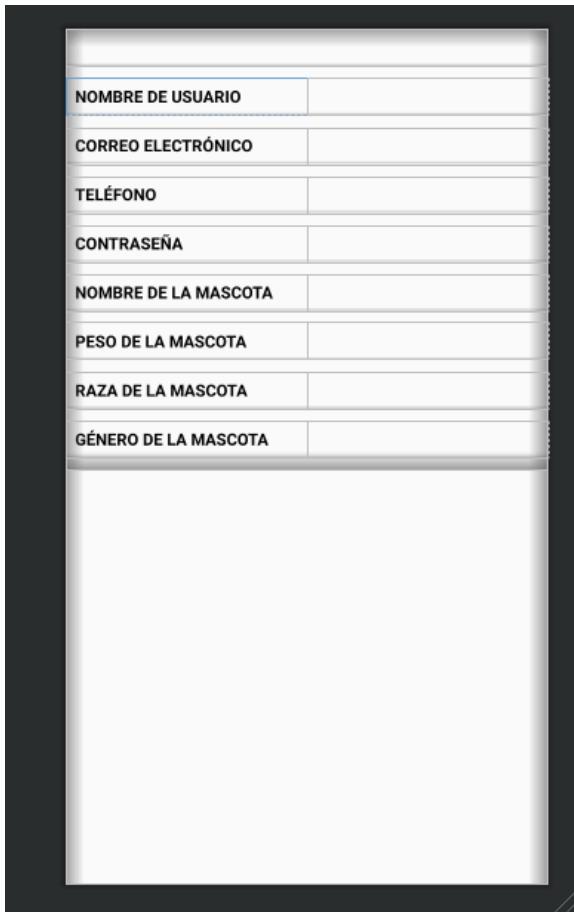


Ilustración 8-22 Programación: Información personal lista

Ahora crearemos unos string para almacenar los datos recogidos de la base de datos, para obtener esos datos tenemos creado un archivo ``.java`` el cual accede a la base de datos mediante un php y almacena esos datos recogidos, después los añadiremos a una lista que tenemos creada con esos campos.

```

        } catch (JSONException e) {
            e.printStackTrace();
        }
       ListAdapter adapter = new SimpleAdapter(
                context: InformacionPersonal.this, list, R.layout.activity_mi_lista,
                new String[]{RecuperarDatosPersonales.KEY_nombre ,RecuperarDatosPersonales.KEY_correo, RecuperarDatosPersonales.KEY_telefon
                new int[]{R.id.usuario, R.id.correo, R.id.telefono, R.id.contrasena, R.id.nombreMascota, R.id.pesoMascota, R.id.razaMascota
        listado.setAdapter(adapter);
    }
    public void VOLVER (View view){
        Intent volver = new Intent( packageContext: this, MainActivity.class);
        startActivityForResult(volver);
    }
}

```

Ilustración 8-23 Programación: Información personal 3

Por último, accederemos a la lista y empezaremos a enlazar los id de la lista con la información almacenada para que la muestre en ella donde queremos.

Cita previa

```

import ...;

public class EnviarCorreo extends AppCompatActivity {

    Button btnVolver, btnEnviar;
    EditText informacioncita;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_enviar_correo);

        informacioncita = findViewById(R.id.Etinformacion);

        btnVolver = findViewById(R.id.btnVolver);
        btnEnviar = findViewById(R.id.btnEnviar);
    }

    public void volver(View view) {
        Intent intent1 = new Intent(getApplicationContext(), MainActivity.class);
        startActivity(intent1);
    }

    public void enviarCorreo(View view){
        Intent email = new Intent(Intent.ACTION_SEND);
        //Decimos que sera para que lo envie
        email.setData(Uri.parse("mailto:"));

        //Metemos el email a donde tiene que enviar
        String[] destinatario = {"peluqueriarequeteguaupos@gmail.com"};
        email.putExtra(Intent.EXTRA_EMAIL, destinatario);

        //Creamos el asunto
        email.putExtra(Intent.EXTRA_SUBJECT, "Cita Previa");

        //Ahora que recoja el texto
        email.putExtra(Intent.EXTRA_TEXT, informacioncita.getText().toString());
    }
}

```

Ilustración 8-24 Programación: Cita previa

Como en todos, lo primero será enlazar las cosas visuales para que hagan lo que queremos, luego crearemos el botón volver y que te vuelva al menú al darle clic y luego creamos el que envía el correo, el cual le diremos que lo envíe, le diremos el destinatario al que queremos que se lo envíe, le añadiremos el asunto y después que recoja el texto introducido en el edit text.

```

//Ahora que recoja el texto
email.putExtra(Intent.EXTRA_TEXT, informacioncita.getText().toString());

email.setType("message/rfc822");

//Iniciamos el intent
startActivity(Intent.createChooser(email, "Envio de Correo"));
}
}

```

Ilustración 8-25 Programación: Cita previa 2

Ahora pondremos el tipo de mensaje, el cual es de texto y que lo envíe cuando le demos a enviar.

8.8. Desarrollo de interface.

La aplicación que he creado para Requeteguaupos, tiene de fondo un degradado de color que va de azul a blanco, lo cual hace referencia al baño. Generalmente esta hecha en tonos azules, simulando el agua; también consta de tonos blancos lo cual se

basa en la espuma que hacen los champús o geles utilizados para el aseo de la mascota.

Estos colores, dan sensación de relajación y limpieza. Para que los clientes sientan que la empresa es leal, digna de confianza y tolerante, nos decantamos por el color azul. Según el tipo de letra da una sensación u otra, y en este caso he decidido incorporar una letra delgada lo cual aporta elegancia a la aplicación móvil de Requeteguaupos. Además, el diseño es sencillo pero atractivo a la vez, así resultará más fácil su uso e intuitiva para todo tipo de clientes, ya sean jóvenes, o de avanzada edad.

8.9. Sistema de gestión empresarial.



Ilustración 8-26 Sistema de gestión empresarial: Odoo

Para el sistema de gestión empresarial usaremos Odoo, ya que es un software que no requiere ningún pago de licencias para ser utilizado, todas las herramientas que tiene están integradas y tienen una interfaz sencilla pero intuitiva a la vez.

Con Odoo podremos cubrir muchas de las necesidades necesarias que tenemos como pueden ser los proyectos, agendas de clientes, contabilidad, facturación, gastos, documentación, empleados, recursos humanos, chat en directo con clientes o empleados, entre otras muchas necesidades que nos cubre en la empresa.

8.10. Documentación.

8.10.1. Manual de instalación.

Para poder instalar la aplicación de Requeteguaupos, deberemos hacer lo siguiente:

En nuestro terminal móvil Android deberemos buscar la Play Store.

Una vez dentro de la Play Store deberemos buscar la aplicación por nombre, en este caso PeluqueriaCaninaRequeteguaupos.

Una vez encontrada la app la pulsaremos y daremos al botón de instalar.

Tras darle al botón de instalar la aplicación empezará a descargarse.

Cuando finalice la instalación podremos abrir la aplicación en nuestro terminal Android ya instalada en el mismo y podremos empezar a utilizarla.

8.10.2. Manual de usuario.



Ilustración 8-27 Manual de usuario: Inicio sesión

Nada más abrir la aplicación la primera ventana que se nos abrirá, será la de inicio sesión, en la cual deberemos introducir nuestro correo y contraseña para iniciar sesión, en el caso de que no sea así daremos en registrarme.



Ilustración 8-29 Manual de usuario: Registro



Ilustración 8-28 Manual de usuario: Registro 2

Al darle a registrarme se nos abrirá una ventana con un pequeño formulario en el cual introduciremos nuestros datos y al final daremos a registrarme el cual nos llevará a la pantalla de inicio de sesión.

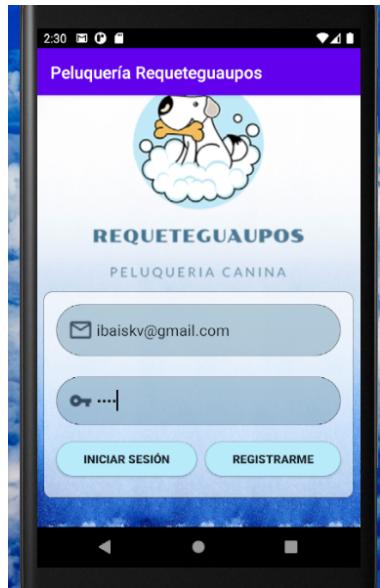


Ilustración 8-30 Manual de usuario: Inicio sesión 2

Nuevamente en el inicio de sesión pondremos nuestros datos y daremos a iniciar sesión el cual nos llevará al menú principal de la aplicación.



Ilustración 8-32 Manual de usuario: Menú principal



Ilustración 8-31 Manual de usuario: Menú principal 2

Una vez en el menú principal tendremos unos botones los cuales nos llevarán al apartado indicado en cada botón.



Ilustración 8-34 Manual de usuario: Información personal

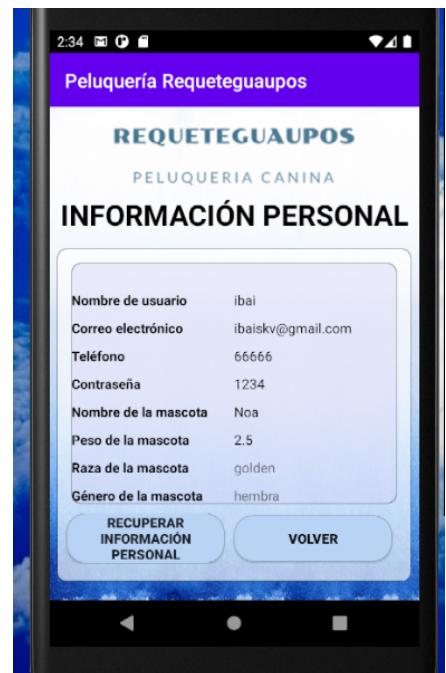


Ilustración 8-33 Manual de usuario: Información personal 2

En el apartado de información personal como podemos ver, tendremos dos botones uno para ver nuestros datos introducidos y otro para volver al menú principal.



Ilustración 8-36 Manual de usuario: Precios

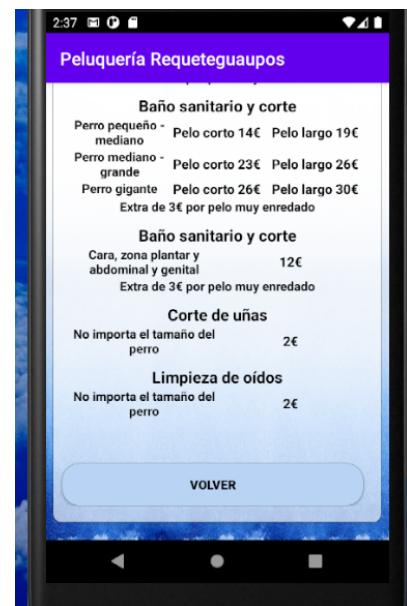


Ilustración 8-35 Manual de usuario: Precios 2

El apartado de precios como podemos observar, nos irá mostrando una lista del servicio y los precios según el tamaño de la mascota y al final tendremos un botón para poder volver al menú principal.



Ilustración 8-38 Manual de usuario: Horario



Ilustración 8-37 Manual de usuario: Horario
2

En el apartado de horarios como se puede ver, nos mostrará los horarios de la peluquería tanto de mañana como de tarde y los días que se encuentran abiertos, además debajo del horario tendremos un botón para volver al menú principal.



Ilustración 8-40 Manual de usuario: Recogida a domicilio



Ilustración 8-39 Manual de usuario:
Recogida a domicilio 2

El apartado de recogida a domicilio explica de qué trata este servicio, quién lo puede solicitar y cómo se puede solicitar además de tener debajo un botón para volver al menú principal.

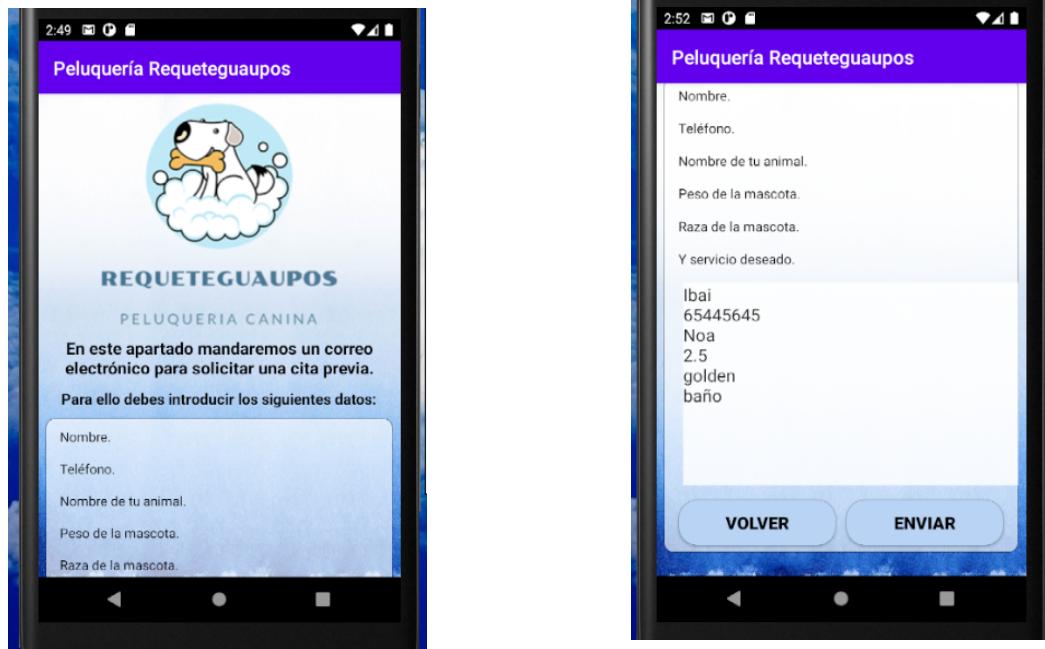


Ilustración 8-42 Manual de usuario: Cita previa

Ilustración 8-41 Manual de usuario: Cita previa 2

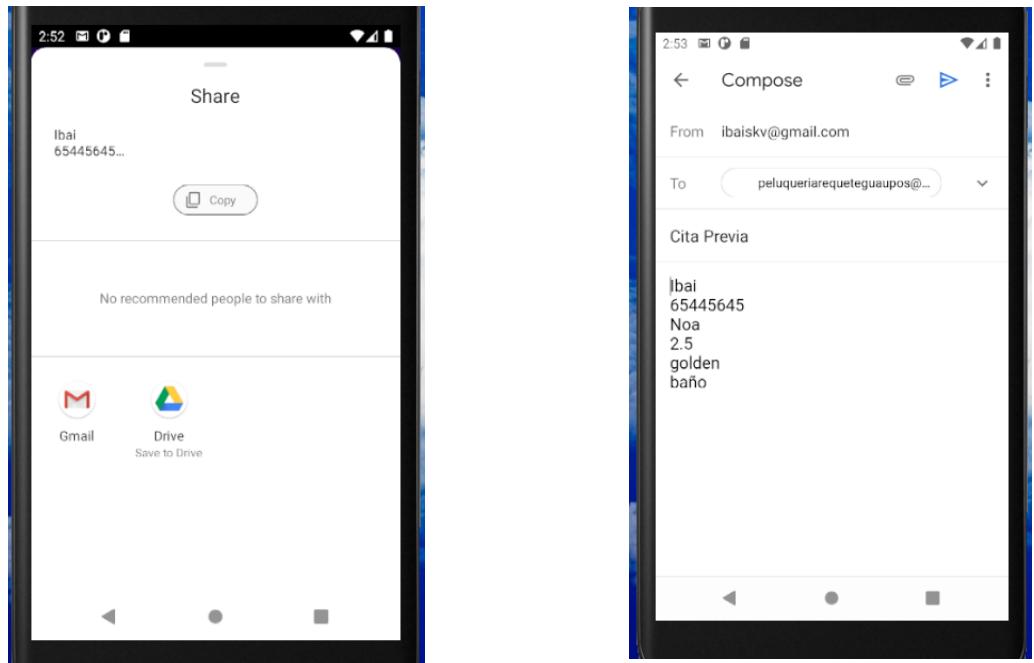


Ilustración 8-44 Manual de usuario: Cita previa 3

Ilustración 8-43 Manual de usuario: Cita previa 4

El apartado de cita previa, explica en lo que consiste este apartado y nos viene unas instrucciones de como solicitarlo, si las seguimos, deberemos introducir los datos solicitados y daremos a enviar. Se nos abrirá un pequeño desplegable el cual le daremos a Gmail que abrirá la aplicación y nosotros lo único que deberemos hacer es darle a enviar para que se envíe el correo automáticamente y podremos darle al botón volver para ir al menú principal.

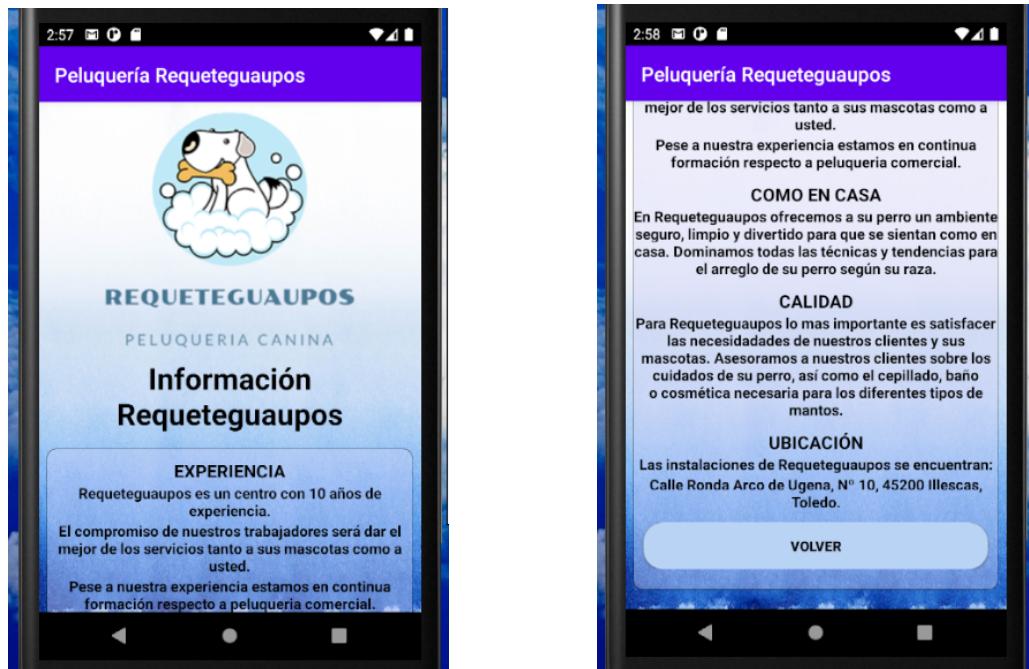


Ilustración 8-46 Manual de usuario: Información

Ilustración 8-45 Manual de usuario: Información 2

Por último, el apartado de sobre nosotros trata un poco de explicar la experiencia que tienen en este sector, la calidad con lo que lo hacen, la ubicación y el ambiente que le ofrecen a las mascotas, además de tener el botón de volver al menú principal.

8.10.3. Manual de administrador.

Para poder administrar la base de datos, el administrador contará con una VPN para poder conectar al servidor. Siempre que quiera editar algo deberá estar conectado a la VPN para tener acceso al servidor.

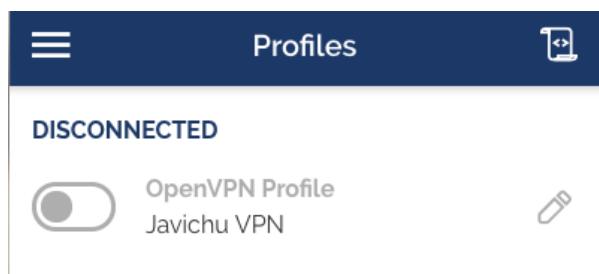


Ilustración 8-47 Manual de administrador: Desconexión VPN

En la imagen anterior podemos ver que está desconectada, le daremos clic al botón de encendido para conectar.

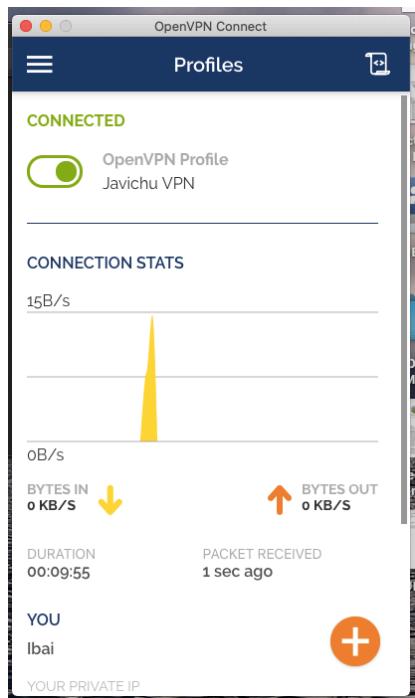


Ilustración 8-48 Manual de administrador: Conexión VPN

Como podemos ver en la imagen, la conexión ya está encendida y tenemos acceso al servidor, ahora continuaremos abriendo MySQL Workbench para poder conectarnos a la base de datos.

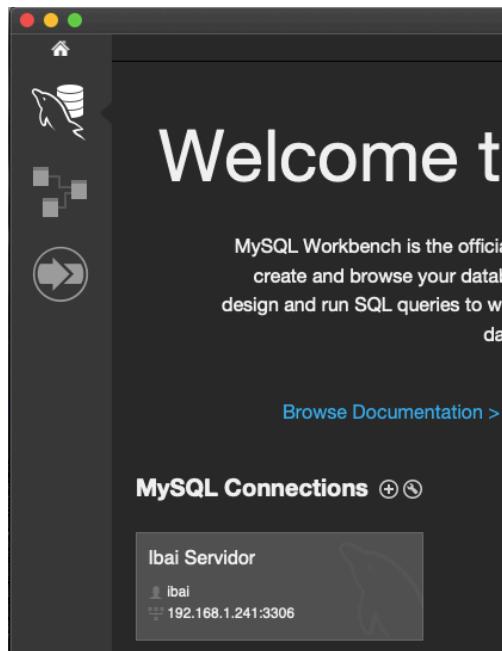


Ilustración 8-49 Manual de administrador: Conexión MySQL

En esta imagen podemos ver que cada administrador tendrá una conexión al servidor, con nuestro usuario, contraseña y la IP donde se encuentra el servidor. Daremos doble clic para conectarnos a la base de datos.

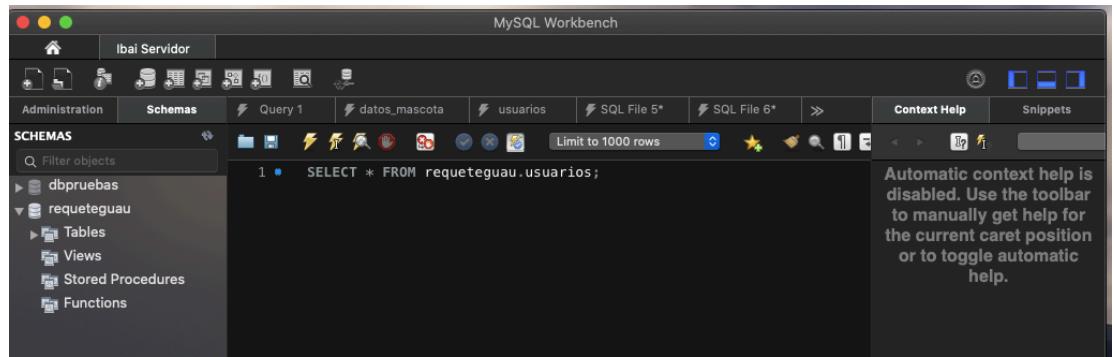


Ilustración 8-50 Manual de administrador: Conexión establecida con la base de datos

Una vez iniciada la sesión del administrador podemos ver como en el lado izquierdo nos saldrá la base de datos de la aplicación, dando clic podremos ver las tablas que tiene, funciones, etc.

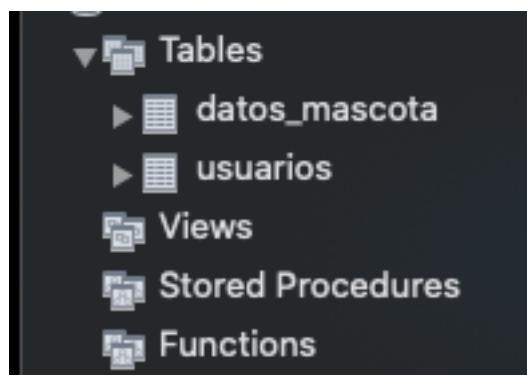


Ilustración 8-51 Manual de administrador: Tablas de la base de datos

En la anterior imagen podemos observar las tablas que tiene nuestra base de datos, para poder editarlas haremos lo siguiente:

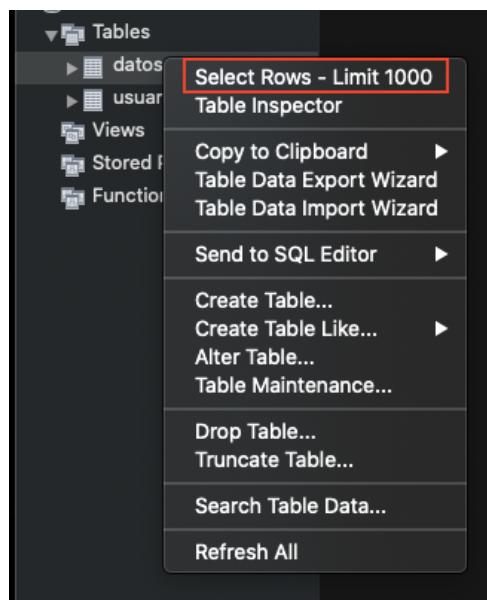


Ilustración 8-52 Manual de administrador: Consulta de datos de una tabla

Seleccionaremos la tabla que deseamos editar, daremos clic derecho y entraremos en select rows – limit 1000 para poder visualizar la tabla y editarla.

Ilustración 8-53 Manual de administrador: Datos de la tabla

Como podemos ver en la imagen nos saldrán todos los registros que están almacenados en la base de datos, para poder editarlos si nos fijamos arriba nos sale Edit: con tres opciones.



Ilustración 8-54 Manual de administrador: Botones de edición de las tablas

- La primera opción sirve para editar los registros.
- La segunda opción sirve para añadir registros.
- La tercera opción sirve para eliminar registros.

Editar registros.

Ilustración 8-55 Manual de administrador: Edición de registros

Como vemos para editar deberemos seleccionar la columna que deseamos editar y daremos clic al botón y ya nos dejará editar el campo.

Ilustración 8-56 Manual de administrador: Edición de registros 1

Para que se guarden los cambios realizados deberemos seleccionar otra celda y ya se guardaría automáticamente en la base de datos.

Añadir registros.

id	nombre_mascota	peso_mascota	raza_mascota	genero_mascota	correo
2	Cascabel	3.2	Siames	Macho	franjfernandez18@gmail.com
3	Noa	28.9	golden	hembra	ibaiskv@gmail.com
► 4	NULL	NULL	NULL	NULL	NULL

Ilustración 8-57 Manual de administrador: Añadir registros

Para añadir registros daremos clic arriba en el apartado edit en el icono de la tabla y un + como podemos ver en la imagen, una vez dado clic automáticamente se nos pondrá en el primer atributo de la tabla para llenar.

id	nombre_mascota	peso_mascota	raza_mascota	genero_mascota	correo
2	Cascabel	3.2	Siames	Macho	franjfernandez18@gmail.com
3	Noa	28.9	golden	hembra	ibaiskv@gmail.com
► 4	Lau	36	Doberman	Macho	ibai@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

Ilustración 8-58 Manual de administrador: Añadir registros 1

Para llenar por completo la tabla como vemos en la imagen cada vez que rellenemos un atributo de la tabla, daremos tabulador para ir añadiendo los datos.

Si queremos añadir otro registro volveremos a repetir los pasos realizados anteriormente y en la tabla deseada.

Eliminar registros.

id	nombre_mascota	peso_mascota	raza_mascota	genero_mascota	correo
2	Cascabel	3.2	Siames	Macho	franjfernandez18@gmail.com
3	Noa	28.9	golden	hembra	ibaiskv@gmail.com
► 4	Lau	36	Doberman	Macho	ibai@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

Ilustración 8-59 Manual de administrador: Eliminar registros

Para poder eliminar registros debemos seleccionar la fila que queremos borrar, en la imagen tenemos la ultima fila seleccionada.

Tras seleccionar la final, deberemos darle clic arriba donde pone edit en el icono de la tabla y un – y automáticamente se nos eliminará de nuestra base de datos.

Result Grid						Filter Rows:	Search	Edit:	Export/Import:
	id	nombre_mascota	peso_mascota	raza_mascota	genero_mascota	correo			
	2	Cascabel	3.2	Siames	Macho	franjfernandez18@gmail.com			
	3	Noa	28.9	golden	hembra	ibaiskv@gmail.com			
	▶	NULL	NULL	NULL	NULL	NULL			

Ilustración 8-60 Manual de administrador: Eliminar registros 1

Para eliminar más registros haremos los mismos pasos en la tabla deseada.

Desconectar MySQL Workbench

Para salir del gestor de la base de datos lo único que deberemos hacer es cerrar la aplicación y después desconectar la VPN.

8.11. Documentación en inglés

8.11.1. Installation manual.

In order to install the Requeteguaupos application, we must do the following:

In our Android mobile terminal, we should look for the Play Store.

Once inside the Play Store we will have to look for the application by name, in this case PeluqueriaCaninaRequeteguaupos.

Once the app has been found, press it and click on the install button.

After hitting the install button, the application will start downloading.

When the installation is finished we can open the application in our terminal Android, already installed on it and we can start using it.

8.11.2. User's manual.



Ilustración 8-61 User Manual: Login

As soon as the application is opened, the first window that will be opened is the login window, in which we will have to enter our email and password to log in.



Ilustración 8-63 User Manual: Registration

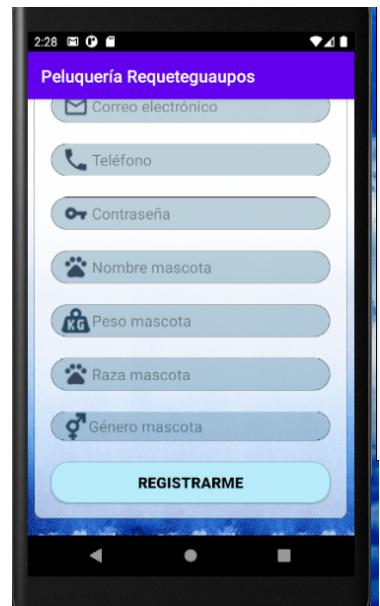


Ilustración 8-62 User Manual: Registration 2

When you hit to register a window will open with a small form in which we will introduce our data and at the end we will give to register which will take us to the login screen.

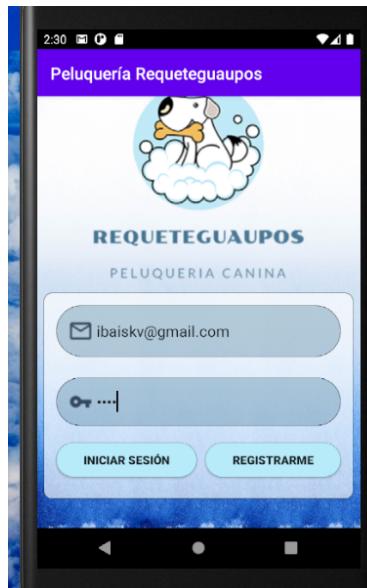


Ilustración 8-64 User Manual: Login 2

Again, in the login we will put our data and we will give to initiate session which will take us to the main menu of the application



Ilustración 8-66 User Manual: Main Menu



Ilustración 8-65 User Manual: Main Menu 2

Once in the main menu we will have some buttons which will take us to the section indicated in each button.



Ilustración 8-68 User Manual: Personal information



Ilustración 8-67 User Manual: Personal information 2

In the section of personal information as we will have two buttons one to see our data entered and another to return to the main menu.



Ilustración 8-70 User Manual: Prices

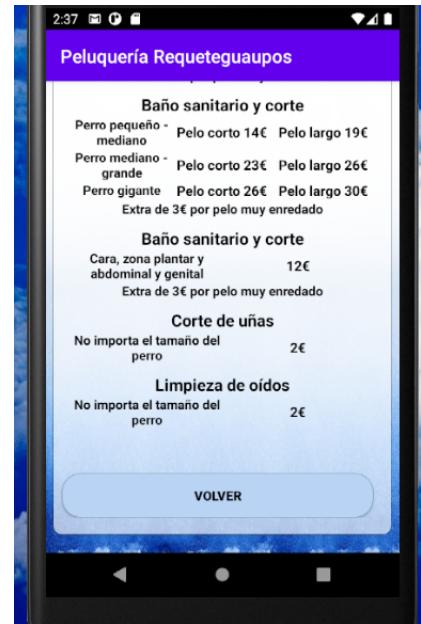


Ilustración 8-69 User Manual: Prices 2

The section of prices as we can see will show a list of the service and prices according to the size of the pet and at the end, we will have a button to return to the main menu.



Ilustración 8-72 User Manual: Schedule



Ilustración 8-71 User Manual: Schedule 2

In the section of schedules as you can see will show us the schedules of the hairdresser's both morning and afternoon and the days they open, also under the schedule we will have a button to return to the main menu.



Ilustración 8-74 User Manual: Home collection



Ilustración 8-73 User Manual: Home collection 2

The section of home collection explains what this service is about, who can request it and how it can be requested besides having a button below to return to the main menu.



Ilustración 8-76 User Manual: Appointment



Ilustración 8-75 User Manual: Appointment 2

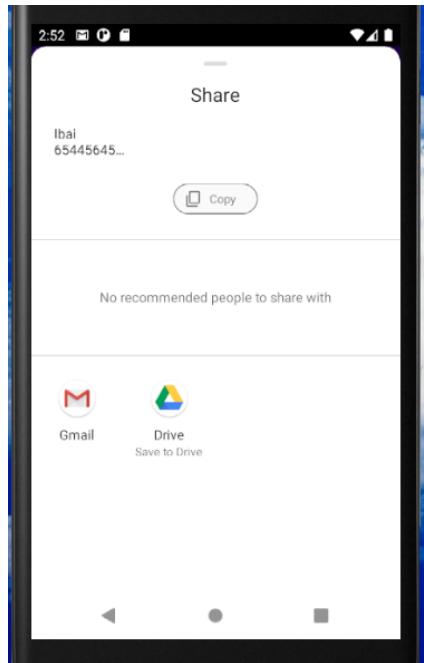


Ilustración 8-78 User Manual: Appointment 3

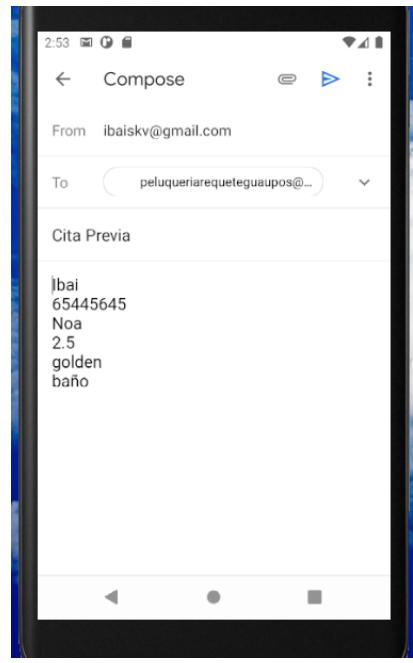


Ilustración 8-77 User Manual: Appointment 4

The section of previous appointment explains in what consists this section and comes to us a few instructions of how to request it, if we follow them we will have to introduce the requested information and we will give to send, a small unfolding will be opened to us which we will add to Gmail which will open the application and the only thing that we will have to do is to give to send so that the mail is sent automatically and we will be able to give to the button to return to go to the main menu.



Ilustración 8-80 User Manual: Information 1

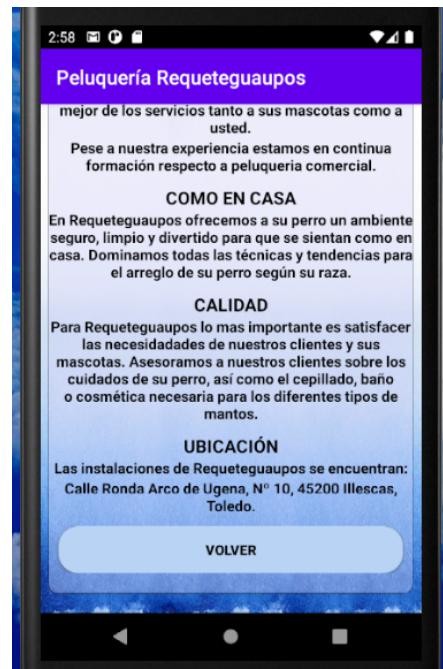


Ilustración 8-79 User Manual: Information 2

Finally, the section about us tries a little to explain the experience they have in this sector, the quality with which they do it, the location and the atmosphere they offer to the pets, besides having the button to return to the button to return to the main menu.

8.11.3. Administrator's manual

In order to administer the database, the administrator will have a VPN to connect to the server. Always when you want to edit something you must be connected to the VPN to have access to the server.

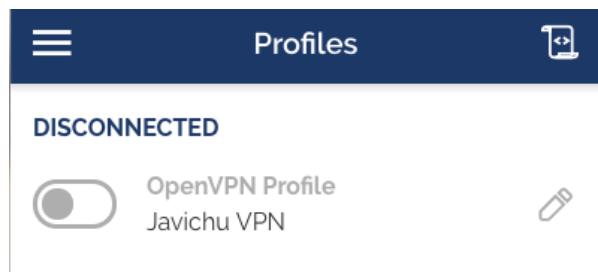


Ilustración 8-81 Administrator Manual: VPN disconnection

In the previous image we can see that it is disconnected, we will click the power button to connect.

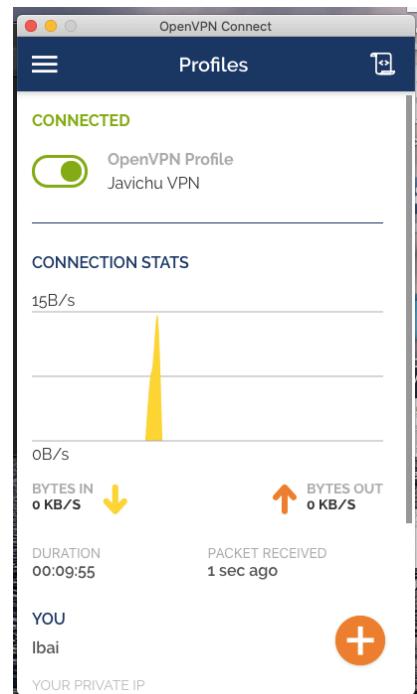


Ilustración 8-82 Administrator Manual: VPN Connection

As we can see in the screenshot, the connection is already on and we have access to the server, now we will continue opening MySQL Workbench to be able to connect to the database.

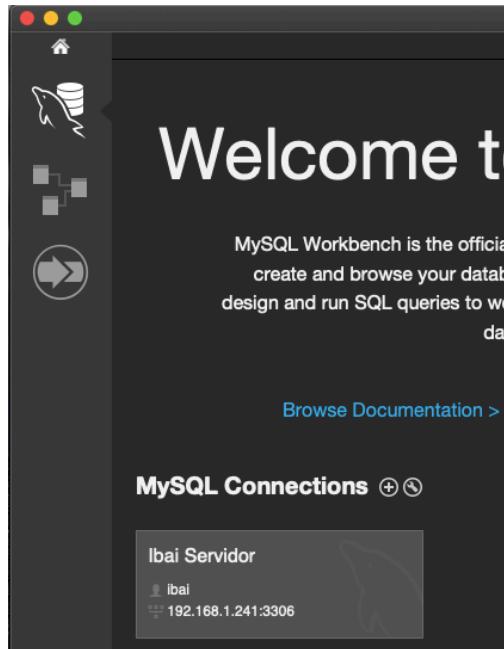


Ilustración 8-83 Administrator Manual: MySQL connection

In this image we can see that each administrator will have a connection to the server, with our user, password and the IP where the server is located. We will double click to connect to the database.

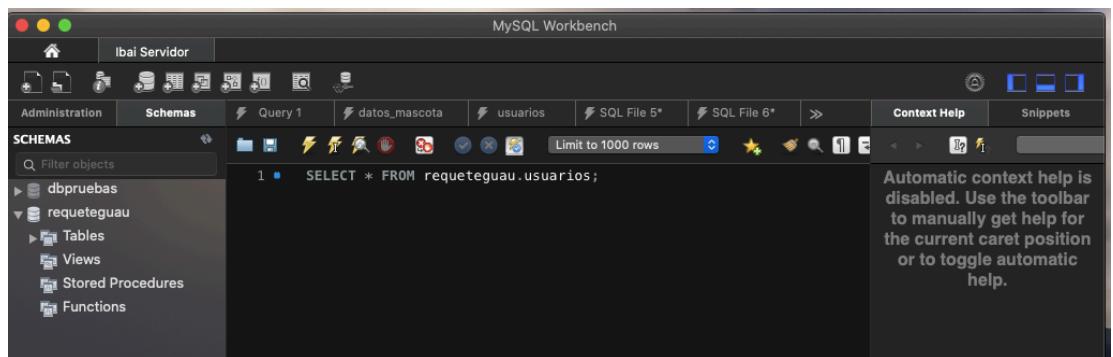


Ilustración 8-84 Administrator Manual: Connection established with the database

Once the administrator's session has started, we can see how the application's database will appear on the left side, by clicking on it we can see the tables it has, functions, etc...

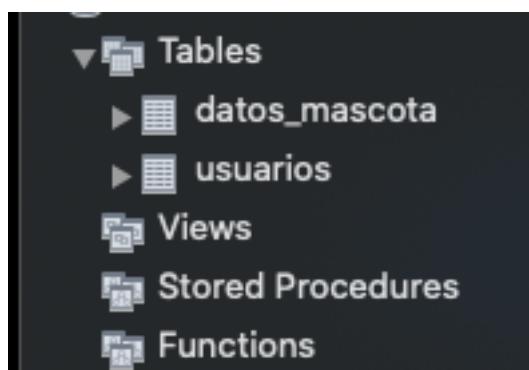


Ilustración 8-85 Administrator Manual: Database Tables

In the previous image we can see the tables that our database has, to be able to edit them we will do the following:

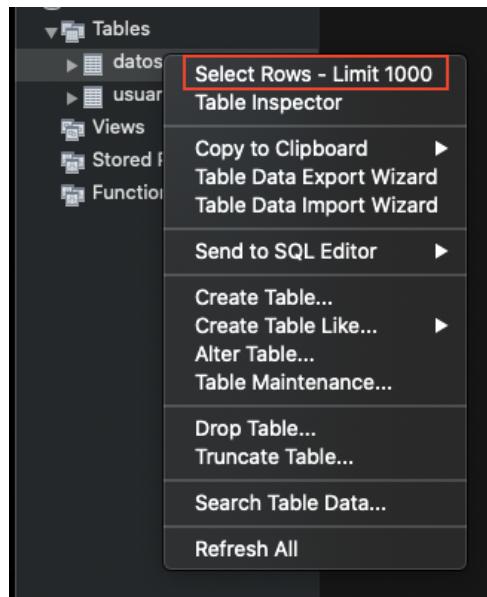


Ilustración 8-86 Administrator Manual: Querying data from a table

Select the table you want to edit, right click and enter select rows – limit 1000 to view the table and edit it.

	id	nombre_mascota	peso_mascota	raza_mascota	genero_mascota	correo
▶	2	Cascabel	3.2	Siames	Macho	franfernandez18@gmail.com
▶	3	Noa	2.5	golden	hembra	ibaiskv@gmail.com
	HULL	HULL	HULL	HULL	HULL	HULL

Ilustración 8-87 Administrator Manual: Table Data

As we can see in the image, we will get all the records that are stores in the database, to be able to edit it if we look above we get Edit: with three options.

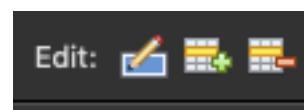


Ilustración 8-88 Administrator Manual: Buttons for editing tables

- The first option is for editing the records.
- The second option is for adding records.
- The third option serves to delete records.

Edit records.

id	nombre_mascota	peso_mascota	raza_mascota	genero_mascota	correo
2	Cascabel	3.2	Siames	Macho	franjfernandez18@gmail.com
3	Noa	28.1	golden	hembra	ibaiskv@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

Ilustración 8-89 Administrator Manual: Editing Records

As we see to edit, we must select the column we want to edit and click the button and let us edit the field.

id	nombre_mascota	peso_mascota	raza_mascota	genero_mascota	correo
2	Cascabel	3.2	Siames	Macho	franjfernandez18@gmail.com
3	Noa	28.9	golden	hembra	ibaiskv@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

Ilustración 8-90 Administrator Manual: Editing Records 1

In order to save the changes made we must select another cell and it will be automatically saved in the database.

Adding records.

id	nombre_mascota	peso_mascota	raza_mascota	genero_mascota	correo
2	Cascabel	3.2	Siames	Macho	franjfernandez18@gmail.com
3	Noa	28.9	golden	hembra	ibaiskv@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

Ilustración 8-91 Administrator Manual: Adding Records

To add records, we will click above in the edit section on the icon of the table and a + as we can see in the image, once clicked it will automatically put us in the first attribute of the table to fill.

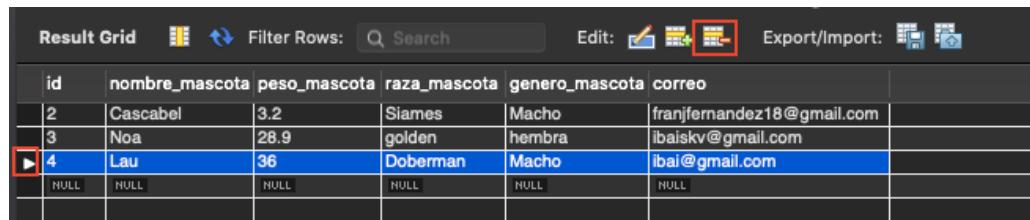
id	nombre_mascota	peso_mascota	raza_mascota	genero_mascota	correo
2	Cascabel	3.2	Siames	Macho	franjfernandez18@gmail.com
3	Noa	28.9	golden	hembra	ibaiskv@gmail.com
4	Lau	36	Doberman	Macho	ibai@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

Ilustración 8-92 Administrator Manual: Adding Records 1

To completely fill the table as we see in the image every time, we fill an attribute of the table, we will give tab to add the data.

If we want to add another record, we will repeat the steps done previously and in the desired table.

Delete records.

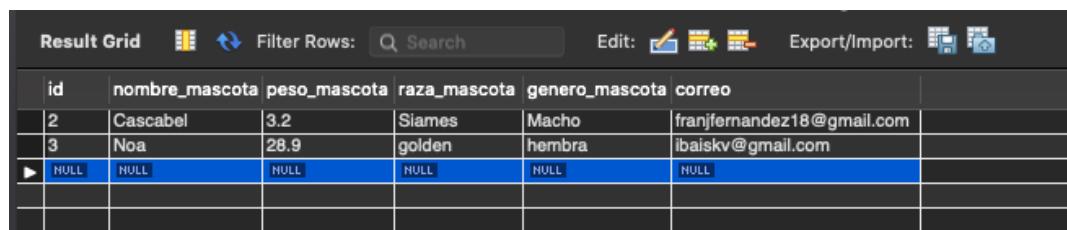


id	nombre_mascota	peso_mascota	raza_mascota	genero_mascota	correo
2	Cascabel	3.2	Siames	Macho	franjfernandez18@gmail.com
3	Noa	28.9	golden	hembra	ibaiskv@gmail.com
4	Lau	36	Doberman	Macho	ibai@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

Ilustración 8-93 Administrator Manual: Delete Records

To be able to delete records we must select the row we want to delete, in the image we have the last row selected.

After selecting the final we must click above where it says edit on the icon of the table and a – and automatically we will be removed from our database.



id	nombre_mascota	peso_mascota	raza_mascota	genero_mascota	correo
2	Cascabel	3.2	Siames	Macho	franjfernandez18@gmail.com
3	Noa	28.9	golden	hembra	ibaiskv@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

Ilustración 8-94 Administrator Manual: Delete Records 1

To remove more records, we will do the same steps in the wished table.

Disconnect MySQL Workbench

To exit the database manager all we have to do, is close the application and then disconnect the VPN.

8.12. Seguridad

En este proyecto prestamos atención a la seguridad para ello nuestros usuarios disponen de acceso a la aplicación con un correo electrónico y contraseña para poder acceder a ella.

Los datos de los usuarios que se intercambian de la aplicación al servidor, como contraseñas, correos, nombres, etc.... irán cifrados con una encriptación SSL para así en el caso de que un atacante consiguiese los datos, no le sirvan de nada.

Nuestra base de datos por seguridad hace una copia cada dos días y lo sube a un servidor en la nube, por si nuestro propio servidor fallase o se cállese por cualquier motivo siempre tendremos una copia de seguridad de respaldo.

Además, para evitar cortes de luz, subidas o bajadas de tensión nuestro servidor tiene instalado un sistema de alimentación interrumpida (SAI).

Hemos realizado una serie de pruebas para verificar el buen funcionamiento de la aplicación.

Entre estas pruebas se encuentran las pruebas de registros, pruebas de inicio de sesión, pruebas de consulta de datos y pruebas de solicitud de cita previa, las cuales se han llevado a cabo para verificar el buen funcionamiento de la base de datos tanto en el registro, para comprobar que sean almacenados dichos datos introducidos, o en el inicio de sesión comprobando que son válidos los datos del usuario y consultando en la propia base de datos de la aplicación.

Además de conseguir el envío de cita previa por correo, llegando al correo de la peluquería canina y recibiendo una respuesta automática al usuario que pide dicha cita.

8.13. Entornos de desarrollo

Android Studio

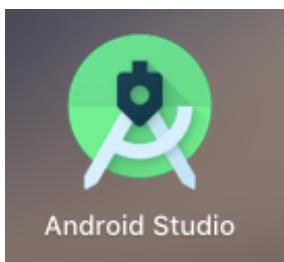


Ilustración 8-95 Entornos de desarrollo: Android Studio

Decidí crear una aplicación móvil en Android ya que actualmente la mayoría de las personas tenemos un dispositivo Android a nuestro alcance ya que es el sistema operativo más demandado en el mercado.

Para ello estuve buscando IDE'S para programar en Android y me decanté por usar Android Studio, sobre todo porque es el IDE oficial para el desarrollo de aplicaciones en Android, que se basa en IntelliJ.

Otra característica por la que decidí utilizar este IDE, es que su sistema es de código abierto, a la hora de subir una aplicación en la Play Store no es tan estricta como la Apple Store, además de que la mayoría son gratuitas, en cambio en Apple Store son de pago casi todas las aplicaciones, también me decanté porque el lenguaje de programación es Java ya que es mucho más amigable y confiable que otros lenguajes de programación, además tiene muchísima información en la página oficial de Android Studio y guías para poder programar en él con facilidad.

Visual Studio Code



Ilustración 8-96 Entornos de desarrollo: Visual Studio Code

Para la edición de los php que uso para las consultas a la base de datos he decidido usar Visual Studio Code ya que es un editor ligero de Microsoft con una interfaz sencilla y fácil de usar.

Lo bueno que tiene es que es gratuito y multiplataforma, lo puedes usar en cualquier sistema operativo deseado, también detecta la sintaxis del código coloreándolo, tiene soporte a los lenguajes más populares como es en mi caso que usaré php y además esta integrado con Git.

9. Conclusiones

Las conclusiones que se pueden sacar de la realización de este proyecto de fin de grado son muy positivas.

Ya que muchos de los requisitos iniciales que se presentaban se han cumplido, lo que resulta satisfactorio.

El hecho de trabajar en un proyecto como este me ha permitido ampliar mis conocimientos de desarrollo de aplicaciones, diseño y bases de datos, lo cual se valora positivamente.

La realización del proyecto me supuso un reto y una gran fuente de aprendizaje respecto a mis objetivos principales.

Gracias al uso de la base de datos he tenido la posibilidad de aprender un poco de php para las comunicaciones al servidor. Me gustaría que el aprendizaje de php no acabe aquí ya que siento que todavía tengo mucho que aprender de este lenguaje.

El resultado final podría haber sido mejor, pero debido a unos problemas que me han surgido con la base de datos he tenido que tomar medidas para así, poder llevarlo a cabo. En general creo que he conseguido completar prácticamente todos los objetivos propuestos en la guía.

10. Anexo

Anexo 1: Diagrama de Gantt

Diagrama de Gantt

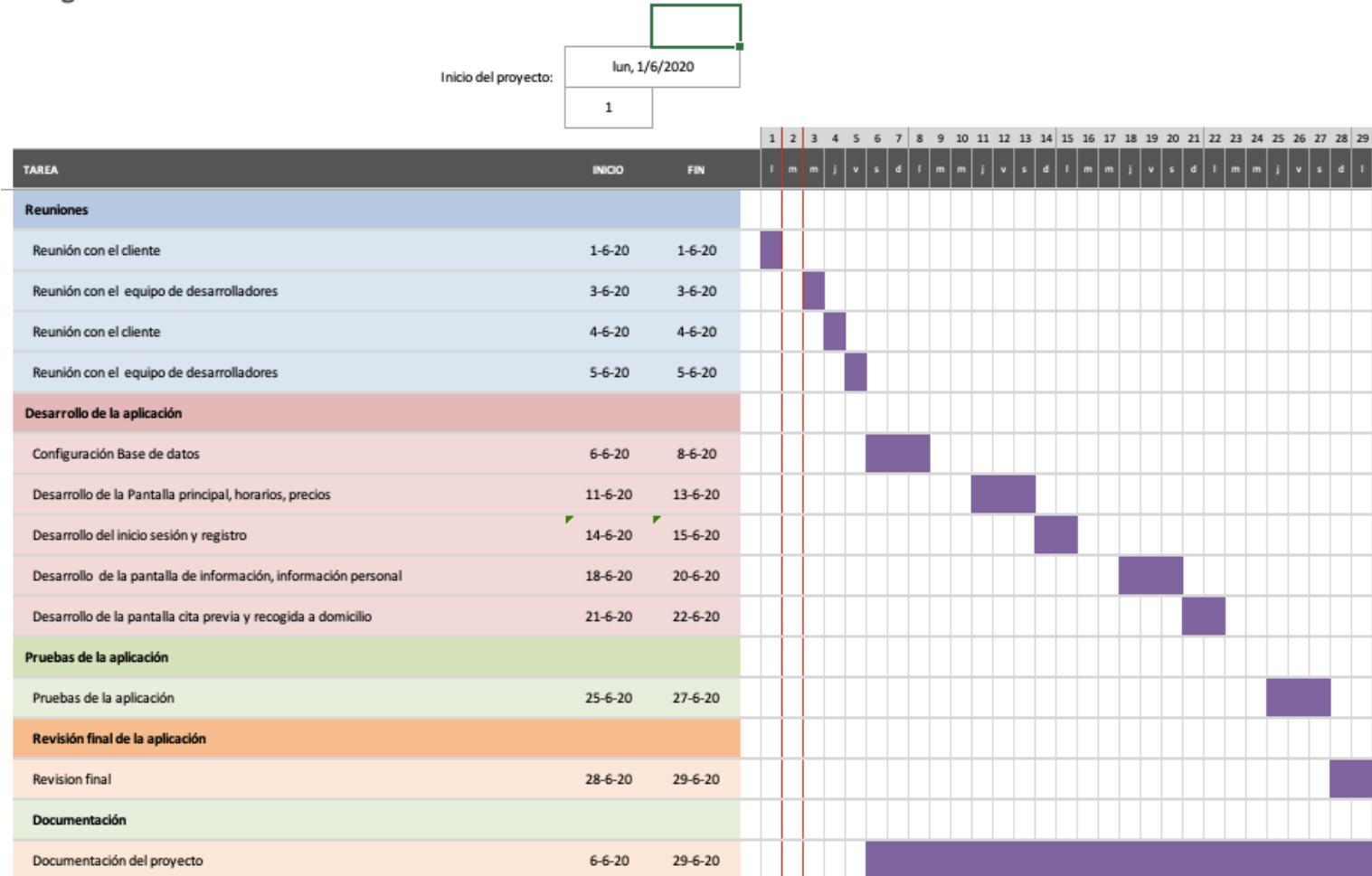


Ilustración 10-1 Diagrama de Gantt

11. Bibliografía

Información de estadísticas de animales en España:

https://www.huffingtonpost.es/entry/radiografia-mascotas-espana-datos_es_5e159aacc5b66361cb5cebdb

Marcas de móviles mas vendidos:

<https://www.movilzona.es/2019/12/26/moviles-mas-vendidos-2019/>

Precio de los servicios de Requeteguapos:

<https://es-la.facebook.com/1477249495865687/posts/1675541559369812/>

Descripción de la empresa:

<https://www.lifeder.com/descripcion-empresa/#Ejemplo>

Especificaciones técnicas del Macbook Air 2015:

<https://www.appleshop.es/producto/macbook-air-core-i5-1-6-11-early-2015/>

MacOS Catalina información:

https://es.wikipedia.org/wiki/MacOS_Catalina

Android información:

<https://developer.android.com/studio/features?hl=es>

Photoshop Información:

<https://neoattack.com/neowiki/photoshop/>

Información Office 365 2019

<https://www.profesionalreview.com/2018/04/29/que-es-office-365/>

Información Firefox

<https://conceptodefinicion.de/firefox/>

Información Visual Studio Code:

https://es.wikipedia.org/wiki/Visual_Studio_Code

Información Unarchiver:

<https://apps.apple.com/es/app/the-unarchiver/id425424353?mt=12>

Información Java:

<https://desarrolloweb.com/articulos/497.php>

<https://rockcontent.com/es/blog/que-es-java/>

<https://www.digitallearning.es/curso-java-para-android.html>

Información PHP:

<https://lenguajesdeprogramacion.net/php/>

<https://es.wikipedia.org/wiki/PHP>

Información MySQL Workbench:

<https://www.monografias.com/trabajos88/mysql-worckbench/mysql-worckbench.shtml#quesmysqla>

<https://ubunlog.com/mysql-workbench-bases-datos/>

Información Apache:

<https://neoattack.com/neowiki/apache/>

<http://culturacion.com/que-es-apache/>

Información Visual Studio Code:

<https://thegroyne.com/2015/05/por-que-usar-visual-studio-code-microsoft/>

Información Android Studio:

<https://lccopen.tech/5-ventajas-programar-para-android/>

<https://developer.android.com/studio/intro?hl=es-419>

Información Odoo:

<https://diagram.es/que-es-el-software-odoo/>

https://www.odoo.com/es_ES/page/all-apps