



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TLAXIACO

MATERIA:

DESARROLLO MOVIL

CARRERA:

INGENIERÍA EN SISTEMAS COMPUTACIONALES.

SEMESTRE: 7US.

ALUMNOS:

JOSÉ ANTONIO BALBUENA CRUZ
HEIDI LUCY BAUTISTA SANJUAN
ANDRÉ EMMANUEL LUNA GÓNZALES
FEDERICO VICTORIANO SIERRA.

DOCENTE: ING. ELOY SANCHEZ SALMORAN

REPORTE FINAL:

APLICACIÓN MÓVIL PARA LA ADOPCIÓN DE MASCOTAS
“MASCOTA-IDEAL”

FECHA DE ENTREGA: 08/12/2023

HEROICA CIUDAD DE TLAXIACO, OAXACA A 08 DE DICIEMBRE DE 2023.

Objetivos

Objetivo General:

- Encontrar un hogar apropiado para mascotas en situación de calle.

Objetivos específicos:

- Localizar mascotas en lugares aproximados
- Buscar gente interesada y en condiciones de adoptar mascotas
- Realizar una evaluación al usuario para ver si es apto de tener una mascota

Justificación:

Esta aplicación surge como respuesta a la problemática persistente de las mascotas en situación de calle, buscando ofrecer una solución integral para encontrarles hogares apropiados. Al utilizar tecnología de geolocalización, "Mascota Ideal" agiliza la ubicación de animales cercanos, optimizando el proceso de rescate y adopción.

Además, la aplicación establece un puente entre las mascotas y posibles adoptantes interesados. A través de la plataforma, se busca no solo encontrar un nuevo hogar, sino uno adecuado. Para ello, se implementa un sistema de evaluación que determina la aptitud del usuario para cuidar de una mascota, garantizando que la adopción sea un acto responsable y sostenible en el tiempo.

El compromiso de la aplicación va más allá de la resolución inmediata de la problemática. Busca generar un impacto positivo a nivel comunitario, contribuyendo a la reducción de la población de animales en situación de calle. Al fomentar la adopción responsable, la aplicación no solo mejora la calidad de vida de las mascotas, sino que también promueve una convivencia armoniosa entre la comunidad y sus integrantes de cuatro patas.

Desarrollo:

Mockup

← Volver

Crea una nueva cuenta

¿Ya estas registrado? Logueate Aquí.

NOMBRE

Jiara Martins

CORREO

hello@reallygreatsite.com

CONTRASEÑA

Registrar

Ilustración 1 Pantalla Crear cuenta

← Volver

Login

¿Aún no estas registrado? Registrate Aquí.

NOMBRE

Jiara Martins

CONTRASEÑA

Ingresar

Ingresa para continuar

Ilustración 2 Pantalla Login

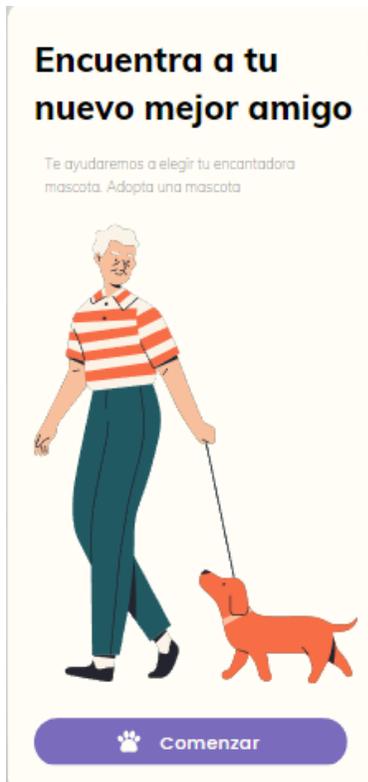


Ilustración 3 Pantalla 3



Ilustración 4 Pantalla Principal



Ilustración 5 Perfil Mascota

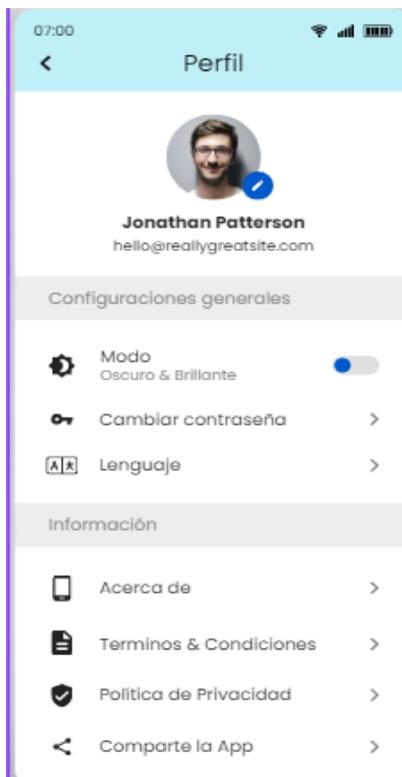


Ilustración 6 Perfil Usuario

Base de datos

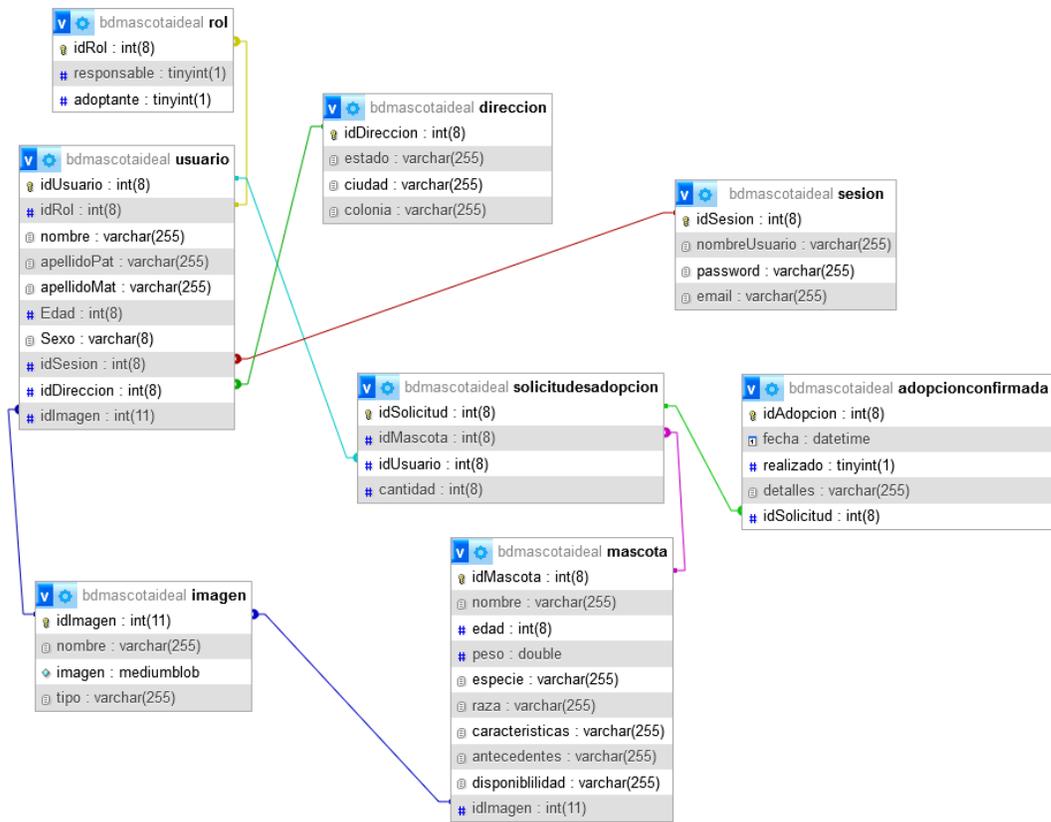


Ilustración 7 Diseño de la base de datos

Backend

1. Creación de la base de datos en MySQL Server.

```
CREATE DATABASE bdmascoideal;  
  
USE bdmascoideal;
```

```

+-----+
| Database |
+-----+
| bdmascotaideal |
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+

```

Tablas en la base de datos.

```

+-----+
| Tables_in_bdmascotaideal |
+-----+
| adopcionconfirmada |
| direccion |
| mascota |
| rol |
| sesion |
| solicitudesadopcion |
| usuario |
+-----+

```

Datos en la tabla “adopcionconfirmada”.

idAdopcion	fecha	realizado	detalles	idSolicitud
1	2023-12-03 00:36:14	No realizado	La mascota fue dada en adopcion anteriormente	1
2	2023-12-03 00:33:54	Realizado	La adopcion de la mascto fue un éxito	2
3	2023-12-01 17:37:22	No realizado	La mascota fue dada en adopcion anteriormente	3

Datos en la tabla "direccion".

idDireccion	estado	ciudad	colonia
1	Oaxaca	Tlaxiaco	Llano Yosovee
2	Oaxaca	Huajuapan de Leon	El mirador
3	Oaxaca	Pinotepa Nacional	Centro

Datos en la tabla "mascota".

idMascota	nombre	edad	peso	especie	raza	caracteristicas	antecedentes	disponibilidad
1	Loki	3	50	Canino	Labrador	Color cafe	Callejero	No disponible
2	Garfield	5	5.5	Felino	Gato egipcio	Blanco	Domestico	Disponible
3	Pepe	2	0.2	Ave	Loro	Color de plumas verde con rojo	Domestico	No disponible

Datos en la tabla "rol".

idRol	responsable	adoptante
1	responsable	adoptante
2	responsable	adoptante

Datos en la tabla "sesión".

idSesion	nombreUsuario	password	email
1	Juan123	juanchis	juan@outlook.com
2	SofiaVR	sofia123	sofia12@gmail.com
3	Jose12SR	joselucho	joseSVR@gmail.com

Datos en la tabla "solicitudopcion".

idSolicitud	idMascota	idUsuario	cantidad
1	1	2	1
2	3	1	1
3	2	3	1

Datos en la tabla "usuario".

idUsuario	idRol	nombre	apellidoPat	apellidoMat	Edad	Sexo	idSesion	idDireccion
1	1	Juan	Reyes	Castro	28	Hombre	1	1
2	2	Sofia	Vazquez	Rojas	18	Mujer	2	2
3	1	Jose	Valencia	Santiago	23	Hombre	3	3

2. Creación de la API y backend de la aplicación.

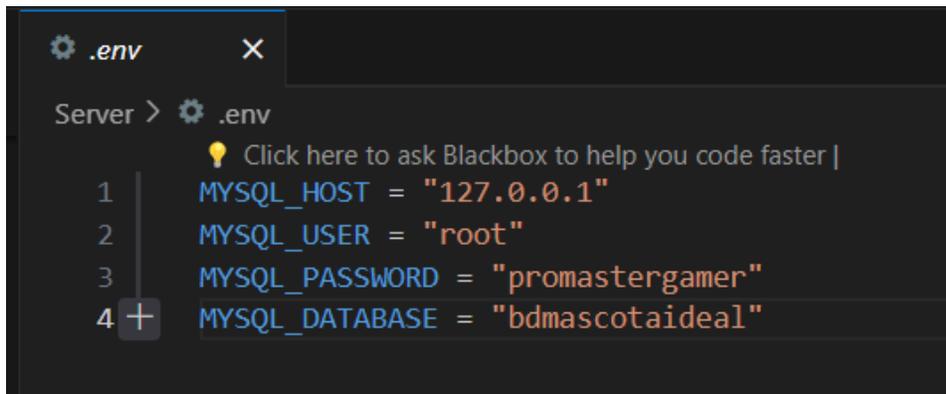
Para trabajar la API con Node js se tienen que ocupar los siguientes comandos en la terminal de la carpeta que contendrá los archivos necesarios para la API.

- **npm init -y:** Este comando inicializa un nuevo proyecto Node.js y crea un archivo package.json con la configuración predeterminada. La opción -y o --yes acepta automáticamente todas las configuraciones predeterminadas sin pedir confirmación.
- **npm add mysql2:** Este comando agrega el paquete mysql2 a las dependencias del proyecto mysql2 es un controlador de MySQL para Node.js que permite interactuar con bases de datos MySQL desde la aplicación.
- **npm add cors:** Este comando agrega el paquete cors a las dependencias del proyecto cors es un middleware para Express que facilita la gestión de las políticas de mismo origen (CORS) en la aplicación, permitiendo o restringiendo las solicitudes HTTP desde diferentes dominios.

- **npm add dotenv:** Este comando agrega el paquete dotenv a las dependencias del proyecto. dotenv permite cargar variables de entorno desde un archivo .env en la aplicación, lo que es útil para gestionar configuraciones sensibles, como claves de API o datos de conexión a la base de datos.
- **npm add express:** Este comando agrega el paquete express a las dependencias del proyecto. express es un marco web para Node.js que facilita la creación de aplicaciones web y API RESTful.
- **npm add nodemon -D:** Este comando agrega el paquete nodemon como una dependencia de desarrollo (-D). nodemon es una herramienta que ayuda a desarrollar aplicaciones Node.js reiniciando automáticamente la aplicación cuando detecta cambios en los archivos del código fuente.

```
{} package.json X
Server > {} package.json > ...
  Click here to ask Blackbox to help you code faster |
1  {
2    "name": "server",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "type": "module",
7    "scripts": {
8      "test": "echo \"Error: no test specified\" && exit 1",
9      "dev": "npx nodemon app.js"
10   },
11   "keywords": [],
12   "author": "",
13   "license": "ISC",
14   "dependencies": {
15     "cors": "^2.8.5",
16     "dotenv": "^16.3.1",
17     "express": "^4.18.2",
18     "mysql2": "^3.6.5"
19   },
20   "devDependencies": {
21     "nodemon": "^3.0.2"
22   }
23 }
24 +
```

Una vez teniendo el archivo package.json listo se hace la conexión con la base de datos a través de un archivo .env el cual contiene lo siguiente:



```
.env
Server > .env
  Click here to ask Blackbox to help you code faster |
1  MYSQL_HOST = "127.0.0.1"
2  MYSQL_USER = "root"
3  MYSQL_PASSWORD = "promastergamer"
4  MYSQL_DATABASE = "bdmascotaideal"
```

Ya que esté la conexión con la base de datos , en otro archivo se realizan las queries que se vayan a ocupar para cada tabla.



```
JS database.js
Server > JS database.js > crearRol > result
  Click here to ask Blackbox to help you code faster |
1  import mysql from 'mysql2';
2  import dotenv from 'dotenv';
3  dotenv.config();
4
5  const pool = mysql
6  .createPool({
7    host: process.env.MYSQL_HOST,
8    user: process.env.MYSQL_USER,
9    password: process.env.MYSQL_PASSWORD,
10   database: process.env.MYSQL_DATABASE,
11  })
12  .promise();
13
14
15  Comment Code
16  export async function getMascotaByID(idMascota){
17
18     const [row] = await pool.query(
19       `SELECT * FROM mascota WHERE idMascota = ?`,
20       [idMascota]
21     )
22     return row[0]; //console.log(row[0]) -> para ejecutar en consola
23     //node database.js
24   }
25   //getMascotaByID(1);
26
27   //Mascotas disponibles para adopcion
28   Comment Code
29   export async function getMascotasDisponibles() {
30     try {
31       const [rows] = await pool.query(
32         `SELECT * FROM mascota WHERE disponiblilidad = 'Disponible'`
33       );
34       return rows;
35     } catch (error) {
36       console.error('Error en la consulta de mascotas disponibles:', error);
37     }
38   }
39 }
```

```
//Filtrar mascota por especie y raza
```

```
Comment Code
```

```
export async function getMascotasPorEspecieYRaza(especie, raza) {  
  try {  
    const [rows] = await pool.query(  
      `SELECT * FROM mascota WHERE especie = ? AND raza = ?`,  
      [especie, raza]  
    );  
    return rows;  
  } catch (error) {  
    console.error('Error en la consulta de mascotas por especie y raza:', error);  
    throw error;  
  }  
}
```

```
+  
//agregar mascota
```

```
Comment Code
```

```
export async function agregarMascota(mascota) {  
  try {  
    const result = await pool.query(  
      `INSERT INTO mascota SET ?`,  
      [mascota]  
    );  
    return result.insertId; // Retorna el ID de la mascota recién insertada  
  } catch (error) {  
    console.error('Error al agregar una nueva mascota:', error);  
    throw error;  
  }  
}
```

```
//Actualizar informacion de una mascota por su ID
```

```
Comment Code
```

```
export async function actualizarMascotaPorID(idMascota, nuevaInfo) {
```

Después de tener listas las queries en otro archivos llamado app.js se realizan las operaciones que la API realizará a través del Frontend.

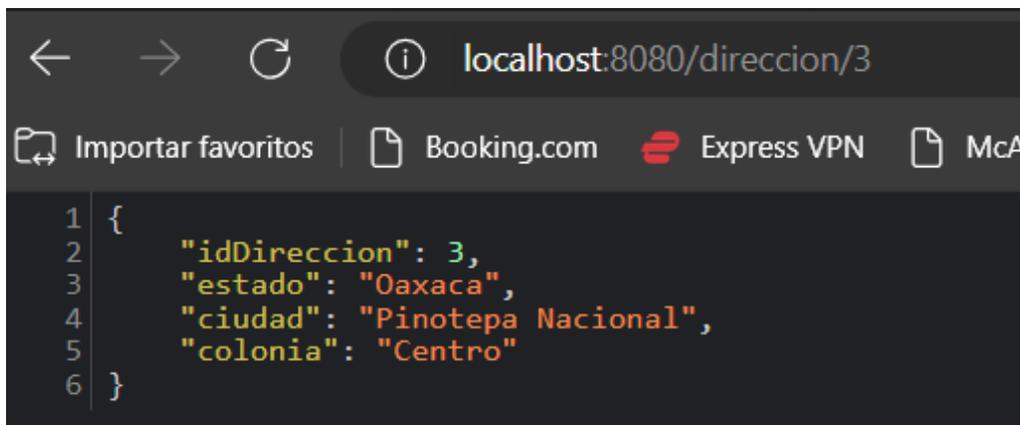
```
JS app.js ×
Server > JS app.js > ...
  Click here to ask Blackbox to help you code faster |
1  import express from "express";
2  import cors from 'cors';
3  import {
4    getMascotaByID,
5    getMascotasDisponibles,
6    getMascotasPorEspecieYRaza,
7    agregarMascota,
8    actualizarMascotaPorID,
9    eliminarMascotaPorID,
10  crearDireccion,
11  getDireccionPorID,
12  actualizarDireccion,
13  eliminarDireccion,
14  crearSesion,
15  getSessionPorID,
16  actualizarSesion,
17  eliminarSesion,
18  crearRol,
19  getRolPorID,
20  actualizarRol,
21  eliminarRol,
22  crearUsuario,
23  getUsuarioPorNombre,
24  actualizarUsuario,
25  eliminarUsuario,
26  crearSolicitudAdopcion,
27  getSolicitudAdopcionPorID,
28  actualizarSolicitudAdopcion,
29  eliminarSolicitudAdopcion,
30  crearAdopcionConfirmada,
31  getAdopcionConfirmadaPorID,
32  actualizarAdopcionConfirmada,
33  eliminarAdopcionConfirmada,
34  } from "./database.js";
35
```

```
134 //Obtener direccion por ID
135 Comment Code
136 app.get("/direccion/:idDireccion", async(req, res) => {
137   const direccion = await getDireccionPorID(req.params.idDireccion);
138   res.status(200).send(direccion);
139 });
140 //Actualizar informacion de una direccion por ID
141 Comment Code
142 app.put("/direccion/:idDireccion", async (req, res) => {
143   const idDireccion = req.params.idDireccion;
144   const nuevaInfo = req.body; // Suponiendo que los nuevos datos están en el cuerpo de la solicitud
145   try {
146     const filasAfectadas = await actualizarDireccion(idDireccion, nuevaInfo);
147     if (filasAfectadas > 0) {
148       res.status(200).send('Dirección actualizada correctamente');
149     } else {
150       res.status(404).send('Dirección no encontrada');
151     }
152   } catch (error) {
153     console.error('Error al actualizar la información de una dirección:', error);
154     res.status(500).send('Error interno del servidor');
155   }
156 });
```

Para las pruebas de que la API esté funcionando, se ejecuta el servidor para poner el puerto y la dirección en el navegador.

```
> server@1.0.0 dev
> npx nodemon app.js

[nodemon] 3.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
Server running on port 8080
```



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/direccion/3'. The browser's address bar includes navigation icons (back, forward, refresh) and a search icon. Below the address bar, there are several tabs: 'Importar favoritos', 'Booking.com', 'Express VPN', and 'McAfee'. The main content area of the browser displays a JSON object with the following structure:

```
1 {
2   "idDireccion": 3,
3   "estado": "Oaxaca",
4   "ciudad": "Pinotepa Nacional",
5   "colonia": "Centro"
6 }
```

Finalmente, la parte del backend está lista y está funcionando.

Frontend:

1. **Página de bienvenida:** esta página se creó para poder seleccionar entre crear una cuenta o iniciar sesión en dado caso de ya tener una.



3. Si seleccionamos crear cuenta, nos manda a la vista con el formulario correspondiente

Bienvenido a Mascota Ideal



Crea una nueva cuenta

Registrarse

4. En caso de que ya cuente con una cuenta se le manda al formulario en donde ingresara su nombre de usuario al igual que su contraseña, y si es que llego a equivocarse o quiere crear una cuenta, se implementó un botón para que pueda ser direccionado al formulario anterior.

← Login

Bienvenido a Mascota Ideal

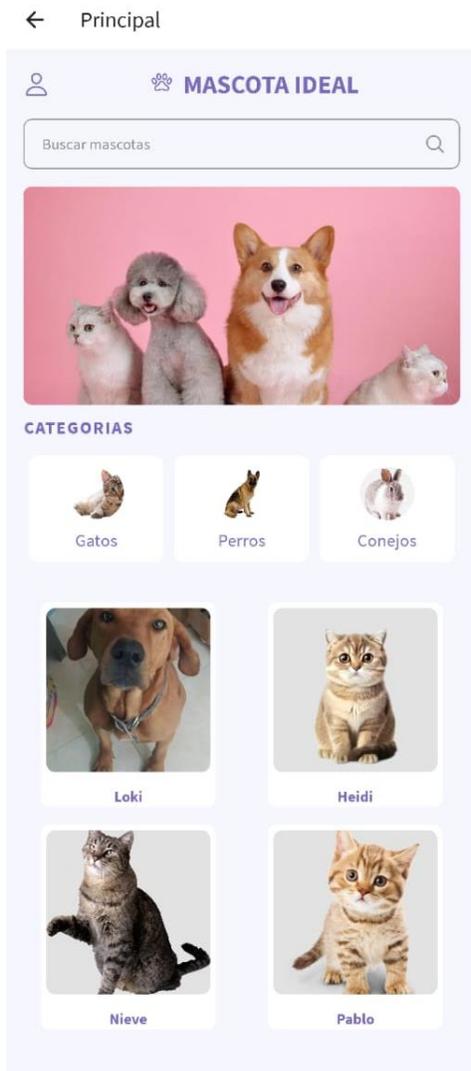


Inicio de Sesión

Iniciar Sesión

[¿No tienes una cuenta? Crear cuenta](#)

- Una vez iniciada sesión o creada la cuenta nos mandara a la página principal de la aplicación en la cual podremos buscar alguna mascota dependiendo de su especie o raza, al igual que al seleccionar las categorías nos mostrara únicamente mascotas de esa especie, abajo se muestran sugerencias de mascotas al igual que sus nombres que al seleccionar alguna, nos mostrara la información necesaria sobre esa mascota



- Una vez iniciada sesión, vemos la pagina principal, podemos ingresar a la imagen de la mascota que queremos adoptar y nos muestra la informacion de la mascota

← Detalles

Informacion



Loki

📍 Agencia La Lobera, Tlaxiaco 

4.6 Km

 **Heidi Lucy Sanjuan Bautista**
Propietaria

¿Estas buscando un nuevo amigo?
Perro melancolico que puede aocmpañarte en los días en los que odias al mundo

ADOPTAME

7. Pero de igual forma, si damos clic en el icono de perfil nos muestra las información de la cuenta

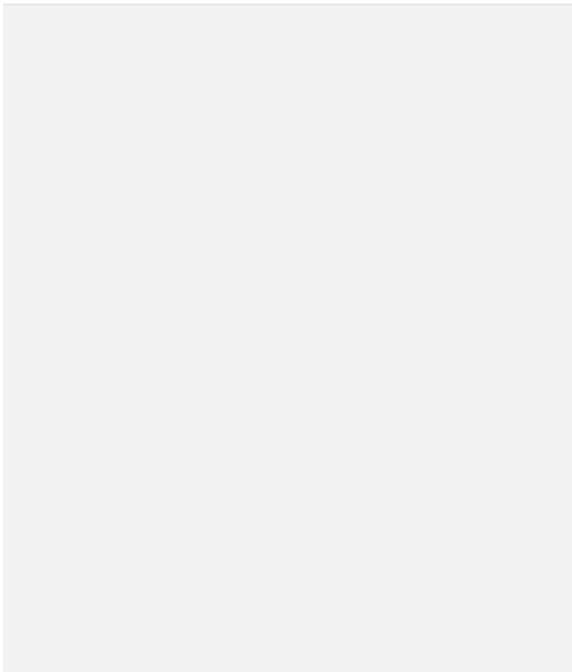
← User



Nombre de la Sesión

Descripción de la Sesión

[Editar Perfil](#) 



8. Si damos clic en editar perfil, nos manda a la ventana de ajustes para poder editar los ajustes del usuario



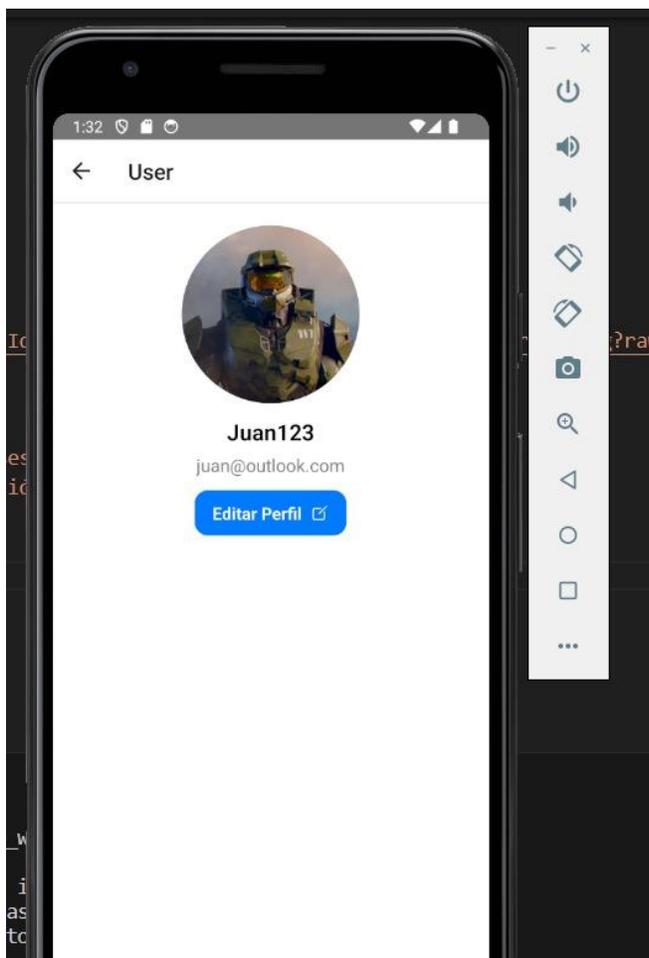
Conexión de la API con el frontend.

Para conectar la API con el frontend se ocupó el método fetch con la respectiva IP de la computadora para que la API pueda ser consumida por el frontend.

```
export const UserProfile: React.FC<UserProps> = ({ navigation }) => {
  const [sesion, setSesion] = useState<any>({});

  useEffect(() => {
    const fetchSesionData = async () => {
      try {
        const response = await fetch('http://192.168.20.64:8080/sesion/3'); // Reemplaza con la UR
        if (!response.ok) {
          throw new Error(`Error ${response.status}: ${response.statusText}`);
        }
        const sesionData = await response.json(); // carga los datos de la sesión en una constante
        setSesion(sesionData); // carga la sesión activa
      } catch (error) {}
      console.error('Error al obtener la sesión:', error);
    };

    fetchSesionData();
  }, []);
};
```



Conclusión:

"Mascota Ideal" no solo se limita a ser una herramienta eficiente para la adopción de mascotas, sino que también se erige como un defensor activo del bienestar animal y la responsabilidad en la tenencia de mascotas. A través de su sistema de evaluación, la aplicación no solo busca encontrar un hogar para los animales, sino el hogar más adecuado, teniendo en cuenta el estilo de vida, las capacidades y el compromiso de los posibles adoptantes.

La tecnología de geolocalización no solo agiliza el proceso de adopción, sino que también facilita la identificación de áreas con mayor concentración de animales en situación de calle. Esto permite a organizaciones y defensores de los derechos de los animales enfocar sus esfuerzos de manera más efectiva, implementando estrategias específicas para abordar las necesidades de esas comunidades.

Además, "Mascota Ideal" fomenta la participación activa de la comunidad al crear conciencia sobre la importancia de la adopción responsable. La aplicación actúa como una plataforma de educación, proporcionando información sobre el cuidado adecuado de las mascotas, la esterilización y la importancia de tratar a los animales con compasión.

El impacto de la aplicación no se limita solo a la reducción de la población de animales en situación de calle, sino que también contribuye a la construcción de comunidades más compasivas y conscientes de la importancia de brindar un hogar amoroso a los animales necesitados.

Bibliografía:

Introduction · React native. (2023, 6 diciembre). <https://reactnative.dev/docs/getting-started>

About Node.js® / Node.js. (s. f.). <https://nodejs.org/en/about>

Carlos Azaustre - Aprende JavaScript. (2022, 20 julio). *API REST con NODE.js // GUÍA de BUENAS PRÁCTICAS* [Vídeo]. YouTube.

<https://www.youtube.com/watch?v=qFmwRriNJWs>

REST API. (s. f.). <https://www.redhat.com/es/topics/api/what-is-a-rest-api>

¿Qué es una API REST? / IBM. (s. f.). <https://www.ibm.com/mx-es/topics/rest-apis>

Bluuweb. (2021, 26 julio). *¿Qué es API, REST y RESTFul? (Explicación simple para programadores JS)* [Vídeo]. YouTube.

<https://www.youtube.com/watch?v=sB6Vc3gze3w>