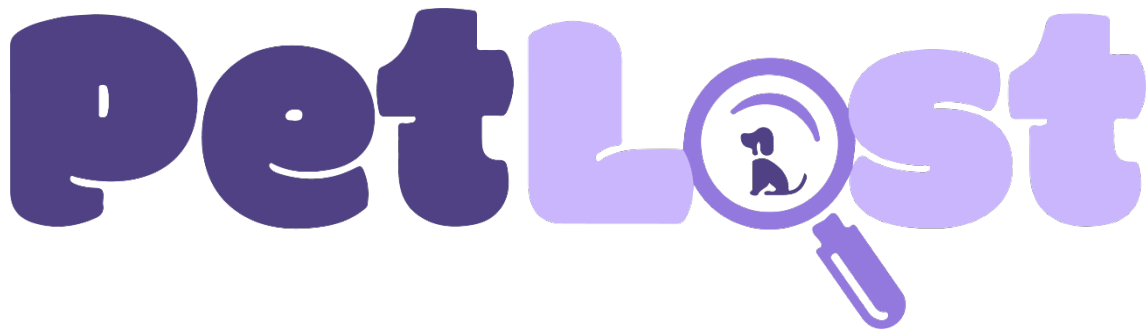




# Sprint II



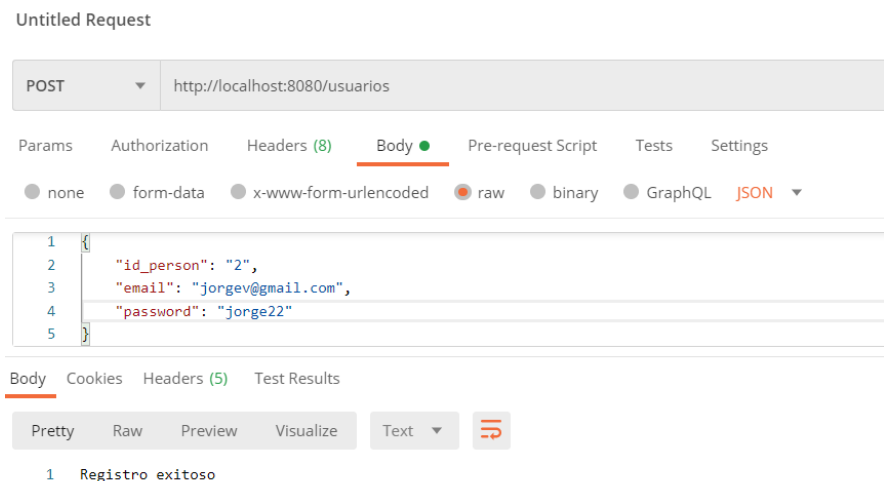
## Localización de mascotas perdidas

Apellidos, Nombres	Correo electrónico	Rol
<b>Vallejo Salazar, Jorge Antonio</b>	jorgevallejo2990@gmail.com	Gestor de Proyecto
<b>Castro Valencia, Cristian</b>	cristiancastro9909@gmail.com	Gestor Base de datos
<b>Montero Corzo, Jesús David</b>	jesusda71.2@gmail.com	Desarrollador Back-end
<b>Useche Perdomo, Joshua Daniel</b>	joshua_dup@hotmail.com	Desarrollador Front-end
<b>Ardila Madero, Leonardo José</b>	lejoarma@hotmail.com	Tester

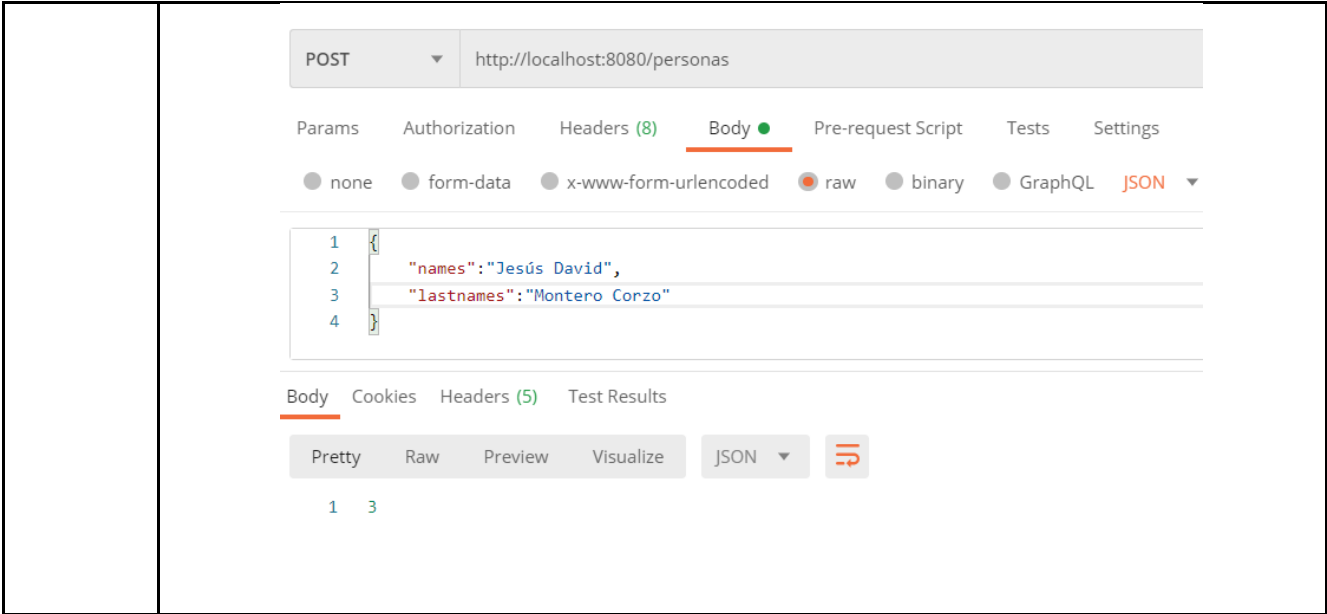
Fecha de presentación: 18/09/2022

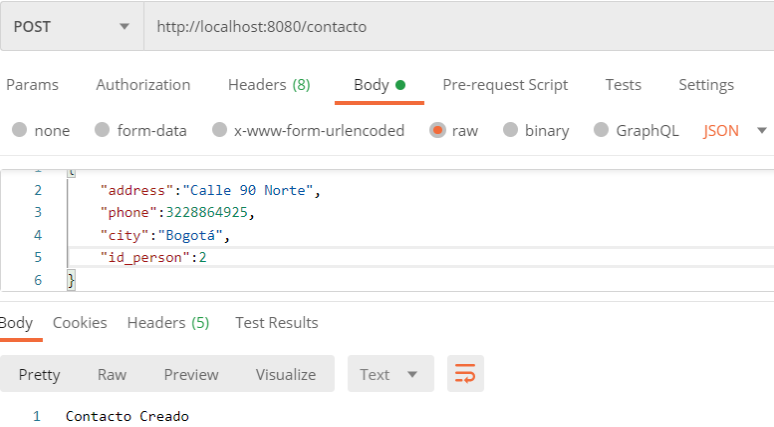
## TESTER (Postman - Workbench)

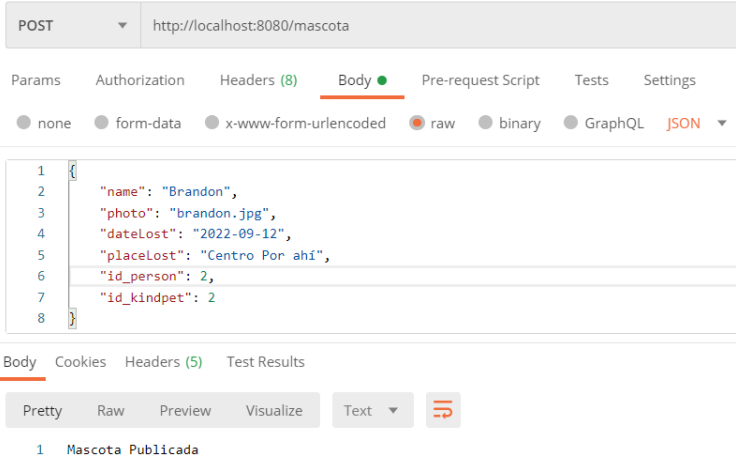
Pruebas de funcionalidades (Registrar, Loguear, Eliminar, Crear, Actualizar.)

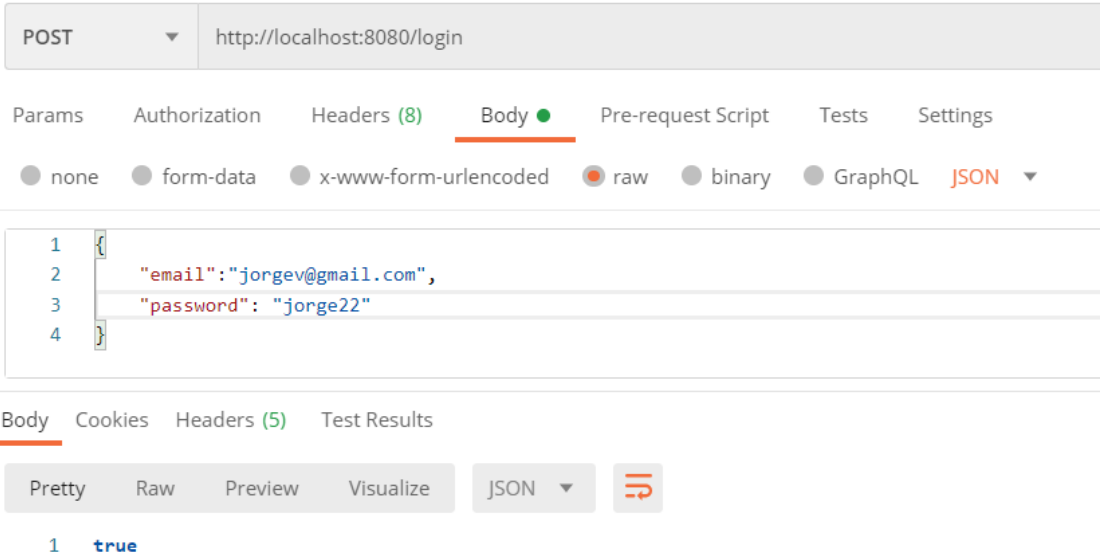
Prueba Funcional (Test Case) N° 1	
Nombre	Registrar Usuario
Descripción	Se verifica la funcionalidad del método Registrar Usuario del backend
Parámetros de Entrada	* id Persona (Persona a la que va ir asociada el correo y contraseña): 2 * Correo: <a href="mailto:jorgev@gmail.com">jorgev@gmail.com</a> * Contraseña: jorge22
Resultado Esperado	Se debe recibir un mensaje indicando el registro exitoso
Resultado Obtenido	<p>Mensaje de "Registro exitoso", además se evidencia en base de datos que existe el registro</p> 

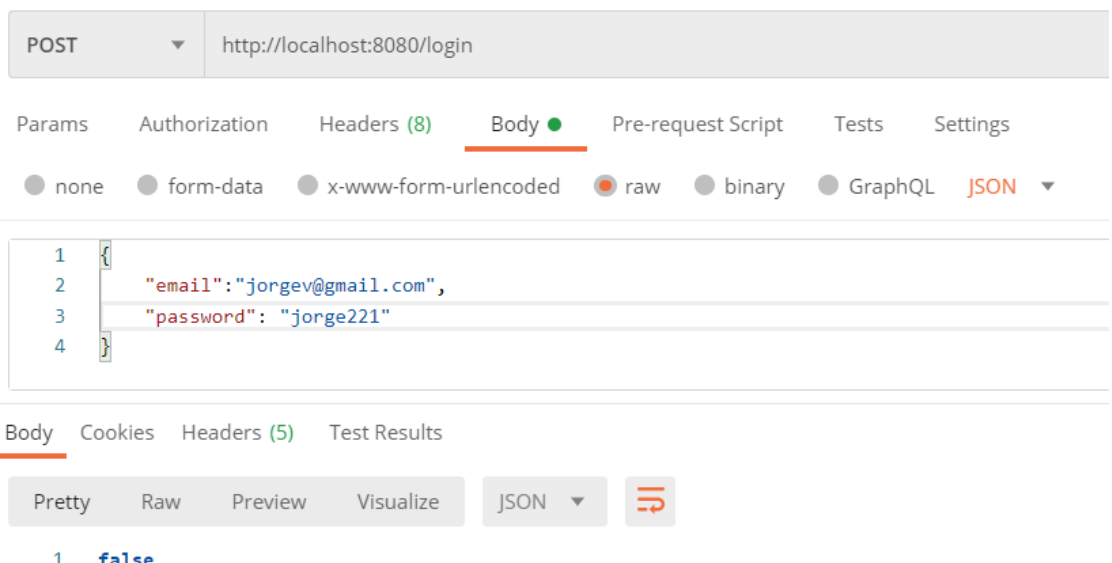
Prueba Funcional (Test Case) N° 2	
Nombre	Registrar Persona
Descripción	Se verifica la funcionalidad del método Registrar Persona del backend
Parámetros de Entrada	* Nombres: Jesús David * Apellidos: Montero Corzo
Resultado Esperado	Se debe recibir el id (3) de la persona recién registrada
Resultado Obtenido	El número 3 fue obtenido, además se evidencia en base de datos que existe el registro.

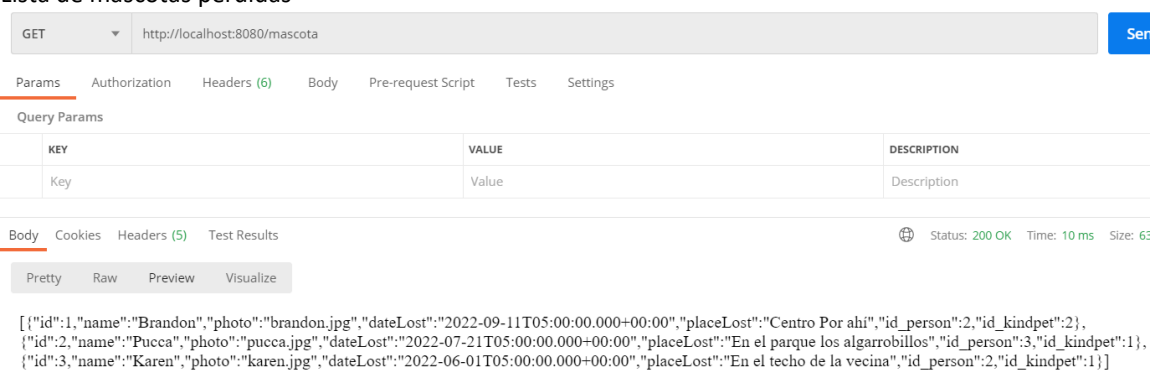


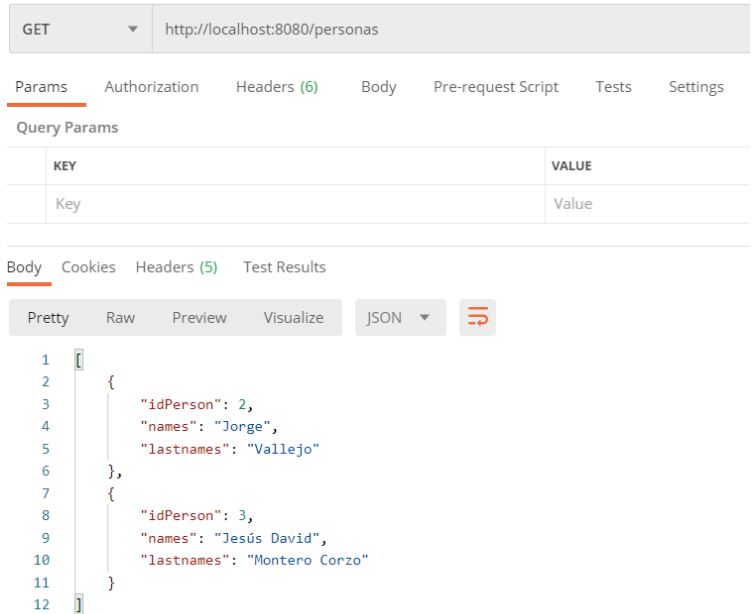
Prueba Funcional (Test Case) N° 3	
Nombre	Agregar Contacto Persona
Descripción	Se verifica la funcionalidad del método Crear Contacto del backend
Parámetros de Entrada	* Dirección: Calle 90 Norte * Celular: 3228864925 * Ciudad: Bogotá * Id Persona: 2
Resultado Esperado	Se debe recibir un mensaje indicando el contacto se ha creado
Resultado Obtenido	Mensaje de “Contacto Creado”, además se evidencia en base de datos que existe el registro. 

Prueba Funcional (Test Case) N° 4	
Nombre	Agregar Mascota Perdida
Descripción	Se verifica la funcionalidad del método Crear Mascota del backend
Parámetros de Entrada	<ul style="list-style-type: none"> <li>* Nombre: "Brandon"</li> <li>* Foto: "brandon.jpg"</li> <li>* Fecha Perdida: "2022-09-12"</li> <li>* Lugar Perdida: "Centro Por ahí"</li> <li>* Id Persona: 2</li> <li>* Id Tipo Mascota: 2</li> </ul>
Resultado Esperado	Se debe recibir un mensaje indicando el contacto se ha creado
Resultado Obtenido	<p>Mensaje de "Mascota Publicada", además se evidencia en base de datos que existe el registro.</p>  <pre> POST http://localhost:8080/mascota  {   "name": "Brandon",   "photo": "brandon.jpg",   "dateLost": "2022-09-12",   "placeLost": "Centro Por ahí",   "id_person": 2,   "id_kindpet": 2 }  Mascota Publicada </pre>

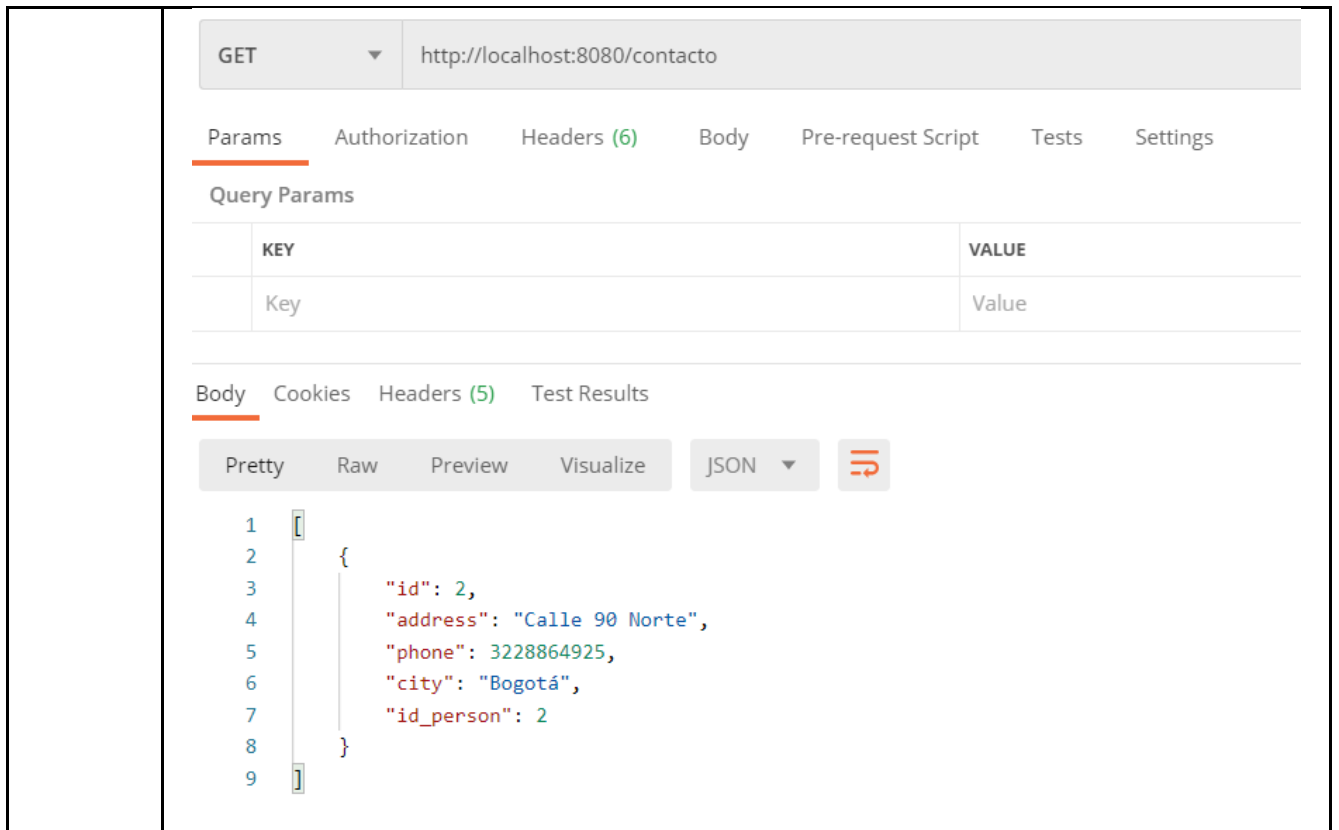
Prueba Funcional (Test Case) N° 5	
Nombre	Login Exitoso
Descripción	Se verifica la funcionalidad del método login del backend cuando el usuario y la contraseña son correctos
Parámetros de Entrada	<ul style="list-style-type: none"> <li>*Correo: jorgev@gmail.com</li> <li>*Contraseña: jorge22</li> </ul>
Resultado Esperado	Se debe recibir un mensaje de true, haciendo alusión que existe el usuario
Resultado Obtenido	<p>true</p>  <pre> POST http://localhost:8080/login  {   "email": "jorgev@gmail.com",   "password": "jorge22" }  true </pre>

Prueba Funcional (Test Case) N° 6	
Nombre	Login Fallido
Descripción	Se verifica la funcionalidad del método login del backend cuando el usuario o la contraseña son incorrectos
Parámetros de Entrada	*Correo: jorgev@gmail.com *Contraseña: jorge221
Resultado Esperado	Se debe recibir un mensaje de false, haciendo alusión que el usuario o la contraseña son incorrectos
Resultado Obtenido	<p>false</p>  <p>The screenshot shows a REST client interface. The request method is POST, and the URL is http://localhost:8080/login. The body is a JSON object: {"email": "jorgev@gmail.com", "password": "jorge221"}. The response is a single line: false.</p>

Prueba Funcional (Test Case) N° 7	
Nombre	Obtener Mascotas Perdidas
Descripción	Se verifica la funcionalidad del método Obtener Mascotas del backend
Parámetros de Entrada	Ninguno
Resultado Esperado	Se debe recibir una lista con todas las mascotas perdidas que están registradas en la base de datos y su respectiva información
Resultado Obtenido	<p>Lista de mascotas perdidas</p>  <p>The screenshot shows a REST client interface. The request method is GET, and the URL is http://localhost:8080/mascota. The response is a JSON array of three lost pets: Brandon, Pucca, and Karen.</p>

Prueba Funcional (Test Case) N° 8	
Nombre	Obtener Personas
Descripción	Se verifica la funcionalidad del método Obtener Personas del backend
Parámetros de Entrada	Ninguno
Resultado Esperado	Se debe recibir una lista con todas las personas que están registradas en la base de datos y su respectiva información
Resultado Obtenido	<p>Lista de personas</p>  <pre> 1  [ 2    { 3      "idPerson": 2, 4      "names": "Jorge", 5      "lastnames": "Vallejo" 6    }, 7    { 8      "idPerson": 3, 9      "names": "Jesús David", 10     "lastnames": "Montero Corzo" 11   } 12 ] </pre>

Prueba Funcional (Test Case) N° 9	
Nombre	Obtener Contactos
Descripción	Se verifica la funcionalidad del método Obtener Contactos del backend
Parámetros de Entrada	Ninguno
Resultado Esperado	Se debe recibir una lista con todos los contactos que están registradas en la base de datos y su respectiva información
Resultado Obtenido	Lista de contactos



The screenshot shows a REST client interface with the following components:

- Method and URL:** GET `http://localhost:8080/contacto`
- Tabs:** Params, Authorization, Headers (6), Body, Pre-request Script, Tests, Settings. The 'Params' tab is selected.
- Query Params Table:**

KEY	VALUE
Key	Value
- Body Section:** Body, Cookies, Headers (5), Test Results. The 'Body' tab is selected.
- Body Format:** Pretty, Raw, Preview, Visualize. The 'Pretty' format is selected, and the content is shown as JSON.
- Response Body (JSON):**

```
[
  {
    "id": 2,
    "address": "Calle 90 Norte",
    "phone": 3228864925,
    "city": "Bogotá",
    "id_person": 2
  }
]
```

## LINK DE REPOSITORIOS

Link Repositorio Backend

<https://github.com/jmontero22c/PetLost-Backend>

Link Repositorio Frontend

<https://github.com/Dev-Joshua/PetLost-Frontend>

Link repositorio documentación y principal

<https://github.com/onfevs/PetLost>



## INFORME RETROSPECTIVA SEMANAL.

1. Entregables
2. Plan de acciones de mejora
3. Mejorar practicas
4. Acuerdos de equipo actualizados
5. Impedimentos a escalar.

1. Se han ido realizando las entregas de lo que se ha solicitado, a medida que transcurre el tiempo, cada integrante del grupo esa cumpliendo a cabalidad su rol.
  2. Se debe mejorar un poco la comunicación, ya que varios integrantes no cuentan con un buen acceso a la red, o tienen otros compromisos, no significa que esto sea un impedimento para lograr las tareas asignadas.
  3. Se han ido proponiendo ideas las cuales han mejorado las practicas que estamos llevando a cabo.
  4. Como grupo cada cambio se somete a una votación en el cual 3 votos es la mayoría.
- Eliminar campo hora de perdida y colocar en la descripción detallada un placeholder.
  - Pestaña anuncios se debe mostrar la imagen con la descripción y botón eliminar o actualizar información
  - Insertar x para volver al home
  - Pestaña registrar se debe agregar x para volver al home.
  - Organizar header en todas las pantallas
  - Definir margin para logo uniforme en todas las pantallas
  - Botón editar cuenta debe enviar al home
  - Cambiar títulos a la paleta de colores definida
  - Cambiar color de párrafos a la paleta de colores definida
  - Botón cerrar sesión debe volver al home
  - Se deben ajustar detalles mínimos en el front como ajustar el header en todas las pantallas.
  - Se debe llegar a un acuerdo con las imágenes de fondo de la aplicación.
  - Se debe empezar a pensar en la parte responsiva.
  - Ya todas las pantallas tienen navegación, solo es ajustar detalles menores, cómo centrar textos, agrandar títulos, cambiar fondos a imágenes. Etc.

### Plus

- ----- CONTACTAR -----
- Se debe crear una tabla en la BD
- Se debe crear la funcionalidad en el BACK

- Pantalla contactar debe tener un formulario con la información de Nombre de dueño, Teléfono y correo personal

## Historias de Usuario Sprint III

Historia de usuario	
<b>ID:</b> H005	<b>Nombre:</b> Publicar mascota perdida (Crear anuncio)
<b>Prioridad:</b> Alta	<b>Iteración:</b> 3
<b>rol:</b> Yo como usuario/cliente	
<b>Funcionalidad:</b> Deseo publicar una mascota perdida para que pueda ser vista por otros usuarios	
<b>Criterio de aceptación:</b> <ul style="list-style-type: none"> <li>• Se debe llenar los siguientes campos: Nombre de la mascota, descripción detallada como Color, Raza y manchas específicas en zonas del cuerpo.</li> <li>• Fecha de perdida</li> <li>• Lugar donde se perdió</li> <li>• Campo para cargar imágenes máx. 3 imágenes por mascota</li> <li>• Si se da click en publicar sin que se halla llenado todos los campos, se debe avisar que ningún campo debe quedar vacío.</li> <li>• Botón de publicar o crear anuncio</li> <li>• Botón para regresar al home o cerrar el nuevo anuncio</li> </ul>	

Historia de usuario	
<b>ID:</b> H006	<b>Nombre:</b> Ver Mascota Perdida
<b>Prioridad:</b> Alta	<b>Iteración:</b> 3
<b>rol:</b> Yo como usuario/cliente	
<b>Funcionalidad:</b> Quiero ver los detalles de una mascota seleccionada para tener más información para localizarla.	
<b>Criterio de aceptación:</b> <ul style="list-style-type: none"> <li>• Se debe mostrar: Imagen, nombre, descripción, si es gato o perro, fecha de perdida, lugar de perdida.</li> <li>• Datos de contacto del dueño como: Nombre, Teléfono y Correo.</li> <li>• Se debe poder conectar a las redes sociales del dueño como Wp, Fb o Instagram</li> <li>• Botón de contactar. Plus</li> </ul>	