

Documentación Sprint 1

PetLost

Localización de mascotas perdidas

Apellidos, Nombres	Correo electrónico	Rol
Useche Perdomo, Joshua Daniel	joshua_dup@hotmail.com	Desarrollador Front-end

Historias de Usuario

Sprint 1

Historia de usuario	
ID: 1	Nombre: Documentación del proyecto
Prioridad: Alta	Sprint: 1
Rol: Desarrollador	
Descripción: Se debe crear la documentación necesaria del proyecto para tener una guía base a seguir para el desarrollo de este proyecto.	
Criterio de aceptación: <ul style="list-style-type: none">• Product Backlog• Crear repositorio en GitHub• Gestión de configuración• Historias de usuario	

Sprint 2

Historia de usuario	
ID: H001	Nombre: Registro de usuario
Prioridad: Alta	Sprint: 2
Rol: Yo como usuario/cliente	
Funcionalidad: Deseo registrarme para poder publicar y ver mascotas perdidas	
Criterio de aceptación: <ul style="list-style-type: none">• Debe ingresar los siguientes datos: Nombre, Apellido, Teléfono y Correo.• Si los datos están completos, se mostrará mensaje de "Registro Exitoso" y se redireccionará al login• Si los datos son incompletos se mostrará un validador en los campos que faltan por diligenciar y un mensaje "Registro Fallido"	

Historia de usuario	
ID: H002	Nombre: Ingreso de usuario
Prioridad: Alta	Sprint: 2
Rol: Yo como usuario/cliente	
Funcionalidad: Deseo ingresar a la aplicación para poder hacer uso del sistema	
Criterio de aceptación: <ul style="list-style-type: none"> • Debe ingresar el correo • Debe ingresar la contraseña y poder visualizarla • Si el usuario existe se redirige a la pantalla principal de la aplicación • Si el usuario no existe se le muestra el mensaje de “Usuario no existe” • Si se da click en el botón ingresar sin llenar un campo se mostrará el mensaje “Todos los campos son obligatorios” 	

Sprint 3

Historia de usuario	
ID: H003	Nombre: Ver Perfil de usuario
Prioridad: Alta	Sprint: 3
Rol: Yo como usuario/cliente	
Funcionalidad: Quiero ver los detalles del perfil recién creado.	
Criterio de aceptación: <ul style="list-style-type: none"> • Se debe mostrar: Nombre, Email y Teléfono • Se debe poder agregar información como: Ciudad/Municipio de residencia • Solo se verá reflejada la información no se puede modificar. 	

Historia de usuario	
ID: H004	Nombre: Ver anuncios (Home)
Prioridad: Alta	Sprint: 3
Rol: Yo como usuario/cliente	
Funcionalidad: Quiero ver los anuncios de mascotas perdidas donde pueda acceder para tener más información.	
Criterio de aceptación: <ul style="list-style-type: none"> • Se debe mostrar: Tarjetas de anuncios con imagen y nombre de la mascota • Una breve descripción sobre la mascota • Poder ordenar perdidas recientes Plus • Poder ordenar perdidas más antiguas Plus • Poder ordenar por ciudades Plus • Poder mostrar fecha de perdida en la tarjeta Plus 	

Sprint 4

Historia de usuario	
ID: H005	Nombre: Publicar mascota perdida (Crear anuncio)
Prioridad: Alta	Sprint: 4
Rol: Yo como usuario/cliente	
Funcionalidad: Deseo publicar una mascota perdida para que pueda ser vista por otros usuarios	
Criterio de aceptación: <ul style="list-style-type: none"> • Se debe llenar los siguientes campos: Nombre de la mascota, descripción detallada como Color, Raza y manchas específicas en zonas del cuerpo. • Fecha de perdida • Lugar donde se perdió • Campo para cargar imágenes máx. 3 imágenes por mascota • Si se da click en publicar sin que se halla llenado todos los campos, se debe avisar que ningún campo debe quedar vacío. • Botón de publicar o crear anuncio • Botón para regresar al home o cerrar el nuevo anuncio 	

Historia de usuario	
ID: H006	Nombre: Ver Mascota Perdida
Prioridad: Alta	Sprint: 4
Rol: Yo como usuario/cliente	
Funcionalidad: Quiero ver los detalles de una mascota seleccionada para tener más información para localizarla.	
Criterio de aceptación: <ul style="list-style-type: none"> • Se debe mostrar: Imagen, nombre, descripción, si es gato o perro, fecha de perdida, lugar de perdida. • Datos de contacto del dueño como: Nombre, Teléfono y Correo. • Se debe poder conectar a las redes sociales del dueño como Wp, Fb o Instagram • Botón de contactar. Plus 	

Gestión de Configuración

Este recurso está diseñado para ayudar a encontrar una mascota perdida, así como proporcionar herramientas de información y/o contacto entre dueños y personas que encuentran mascotas, formularios para informar una pérdida con información detallada de evento de pérdida. Además, aquellos que encuentran a una mascota perdida deben encontrar formas confiables de devolver el animal a casa de manera segura.

Objetivo

Facilitar la búsqueda, localización y recuperación de mascotas perdidas por medio del uso de un aplicativo web.

Público Objetivo:

Familias y dueños de mascotas.

Impacto Esperado:

Aumento en las cifras de recuperación de mascotas extraviadas.

Link Repositorio GitHub: <https://github.com/Dev-Joshua/Proyecto-Ciclo4>

Integrantes del equipo Roles y tecnologías

- JOSHUA USECHE
Rol: DESARROLLADOR FULL-STACK
Tecnologías: HTML, CSS, JavaScript, Bootstrap, Figma, React, NodeJs, MongoDB

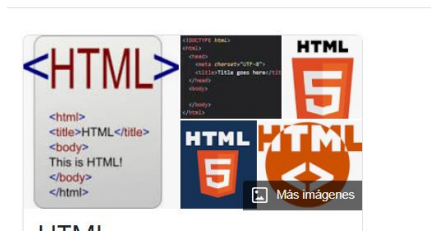
JavaScript

JavaScript es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se usará para la parte Frontend.



HTML

(HyperText Markup Language) hace referencia al lenguaje de marcado para la elaboración de páginas web. Mediante HTML le damos estructura a nuestro proyecto del lado del Frontend.



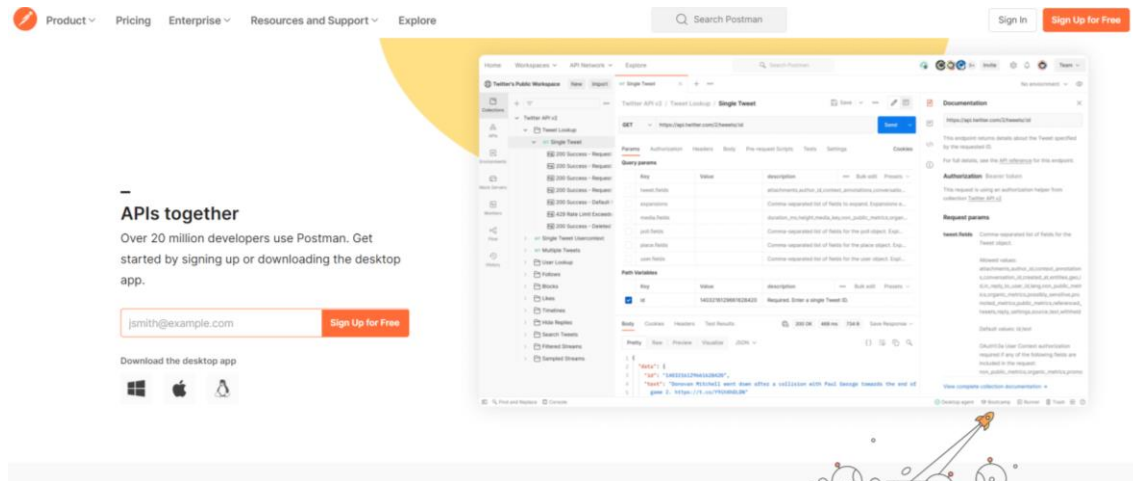
CSS

(Cascadin Style Sheets) hace referencia a hojas de estilo en cascada, es un lenguaje de estilos utilizado para definir y describir la presentación de un documento estructurado escrito en HTML.



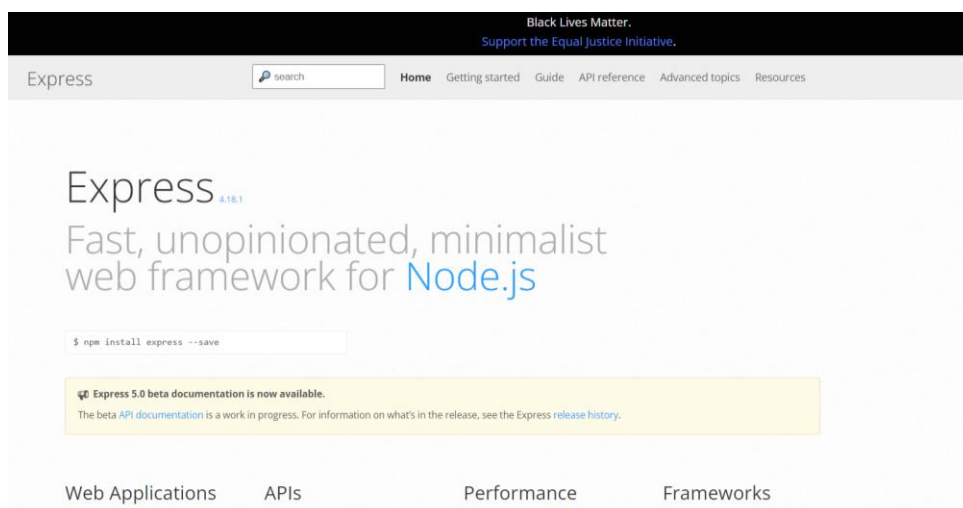
Postman

Postman es una herramienta que se utiliza, sobre todo, para el testing de API REST, aunque también admite otras funcionalidades que se salen de lo que engloba el testing de este tipo de sistemas. La utilizaremos para probar todas las peticiones a nuestro servidor, muy utilizada para Backend.



Express

Express es un entorno de trabajo para aplicaciones web para la programación con node.js, es de código abierto y con licencia MIT. Se utiliza para desarrollar aplicaciones web y APIs.



Editor de código: Visual Studio Code

Extensiones:

Bracket pair colorizer



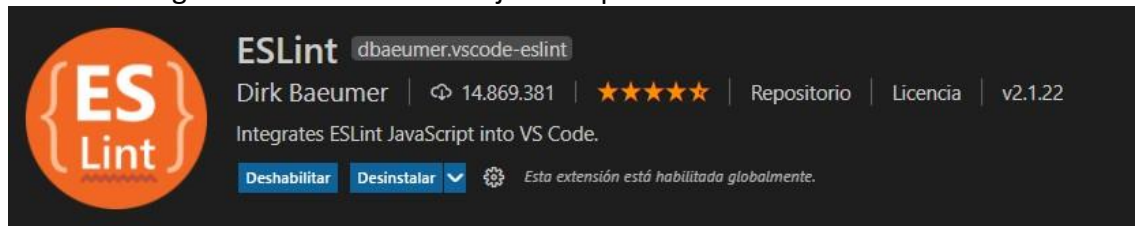
Bracket Pair Colorizer `coenraads.bracket-pair-colorizer`
CoenraadS | 5.421.333 | ★★★★★ | Repositorio | Licencia | v1.0.61
A customizable extension for coloring matching brackets
[Deshabilitar](#) [Desinstalar](#) [⚙️](#) Esta extensión está habilitada globalmente.

ES7 React/Redux/GraphQL/React-Native snippets: snippets para react



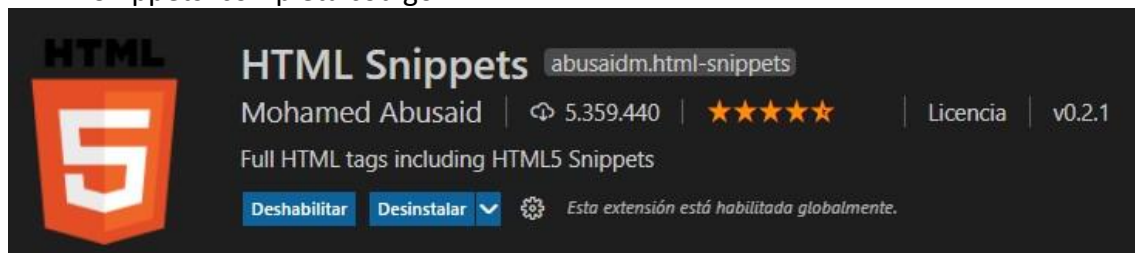
ES7 React/Redux/GraphQL/React-Native snippets `dsznajder.es7-react-js-snippets`
dsznajder | 2.940.632 | ★★★★★ | Repositorio | v3.1.1
Simple extensions for React, Redux and GraphQL in JS/TS with ES7 syntax
[Deshabilitar](#) [Desinstalar](#) [⚙️](#) Esta extensión está habilitada globalmente.

ESLint: corregir errores de sintaxis en java script



ESLint `dbaeumer.vscode-eslint`
Dirk Baeumer | 14.869.381 | ★★★★★ | Repositorio | Licencia | v2.1.22
Integrates ESLint JavaScript into VS Code.
[Deshabilitar](#) [Desinstalar](#) [⚙️](#) Esta extensión está habilitada globalmente.

HTML Snippets: completa código HTML



HTML Snippets `abusaidm.html-snippets`
Mohamed Abusaid | 5.359.440 | ★★★★★ | Licencia | v0.2.1
Full HTML tags including HTML5 Snippets
[Deshabilitar](#) [Desinstalar](#) [⚙️](#) Esta extensión está habilitada globalmente.

Intellisense for CSS class names in HTML: autocompletar o auto rellenar las clases CSS



IntelliSense for CSS class names in HTML `zignd.html-css-class-completion`
Zignd | 3.459.309 | ★★★★★ | Repositorio | Licencia | v1.20.0
CSS class name completion for the HTML class attribute based on the definitions found in your workspace.
[Deshabilitar](#) [Desinstalar](#) [⚙️](#) Esta extensión está habilitada globalmente.

JavaScript (ES6) code snippets: autorellenar




JavaScript (ES6) code snippets xabikos.javascriptsnippets

charalampos karypidis | 5.570.231 | ★★★★★ | Repositorio | v1.8.0

Code snippets for JavaScript in ES6 syntax

[Deshabilitar](#) [Desinstalar](#) ⚙️ Esta extensión está habilitada globalmente.

Material Icon Theme: Iconos para todos los archivos




Material Icon Theme pkief.material-icon-theme

Philipp Kief | 7.896.659 | ★★★★★ | Repositorio | Licencia | v4.7.0

Material Design Icons for Visual Studio Code

[Instalar](#) ⚙️

Path intellisense: completa un path a la hora de integrarlo




Path Intellisense christian-kohler.path-intellisense

Christian Kohler | 4.846.211 | ★★★★★ | Repositorio | Licencia | v2.3.0

Visual Studio Code plugin that autocompletes filenames

[Instalar](#) ⚙️

Prettier – Code formatter: Indenta código




Prettier - Code formatter esbenp.prettier-vscode

Prettier | 13.111.496 | ★★★★★☆ | Repositorio | Licencia | v6.4.0

Code formatter using prettier

[Instalar](#) ⚙️

Reactjs code snippets: Disparadores de código React JS



Reactjs code snippets


xabikos.reactsnippets

charalampos karypidis | 883.330 | ★★★★★ | Repositorio | v2.4.0

Code snippets for Reactjs development in ES6 syntax

[Instalar](#) ⚙️

Vscode-pdf: Lector de PDF



vscode-pdf


tomoki1207.pdf

tomoki1207 | 803.861 | ★★★★★ | Repositorio | Licencia | v1.1.0

Display pdf file in VSCode.

[Instalar](#) ⚙️

Convertidor de Markdown a PDF



Markdown PDF


yzane.markdown-pdf

yzane | 804.300 | ★★★★★ | Repositorio | Licencia | v1.4.4

Convert Markdown to PDF

[Instalar](#) ⚙️

Creador de Markdown



Markdown All in One

yzhang.markdown-all-in-one

Yu Zhang | 2.686.433 | ★★★★★ | Repositorio | Licencia | v3.4.0

All you need to write Markdown (keyboard shortcuts, table of contents, auto preview and more)

[Instalar](#) ⚙️

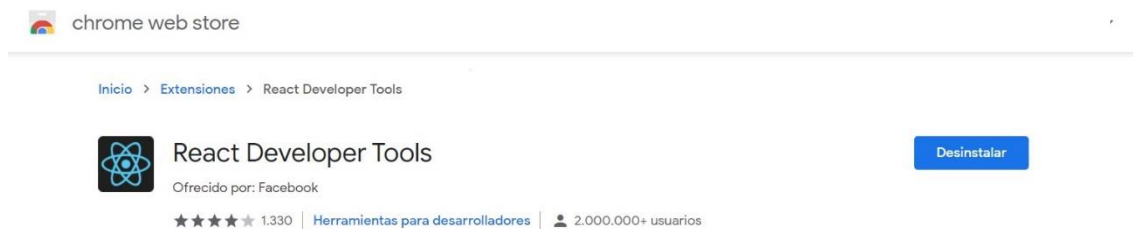
Node JS

Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web.⁴

Descargar siempre la versión en LTS

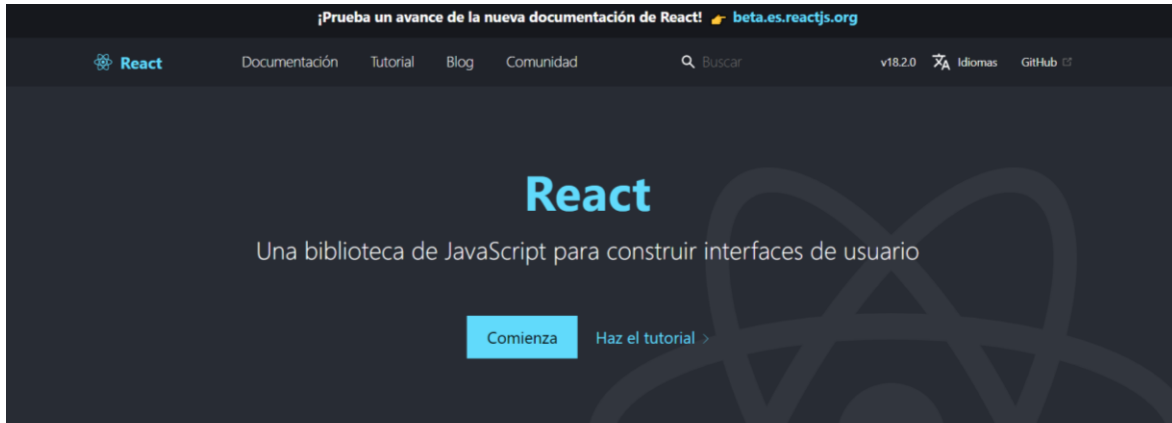


Extensión de Google Chrome: React Developer Tools



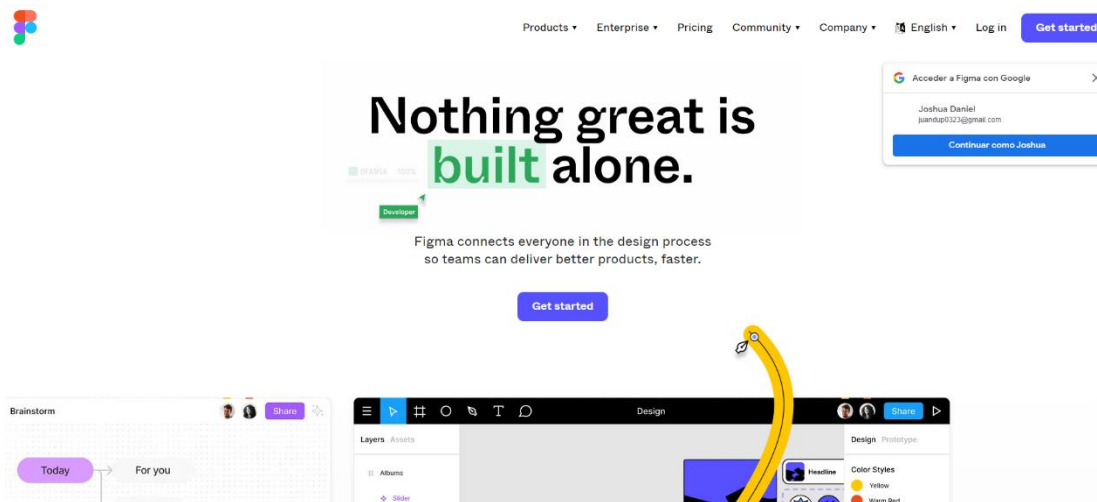
React Js

React es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Te ayuda a crear interfaces interactivas de forma sencilla, diseña vistas simples para cada estado de la aplicación, también se encargará de actualizar y renderizar de manera eficiente los componentes correctos cuando los datos cambien.



Figma

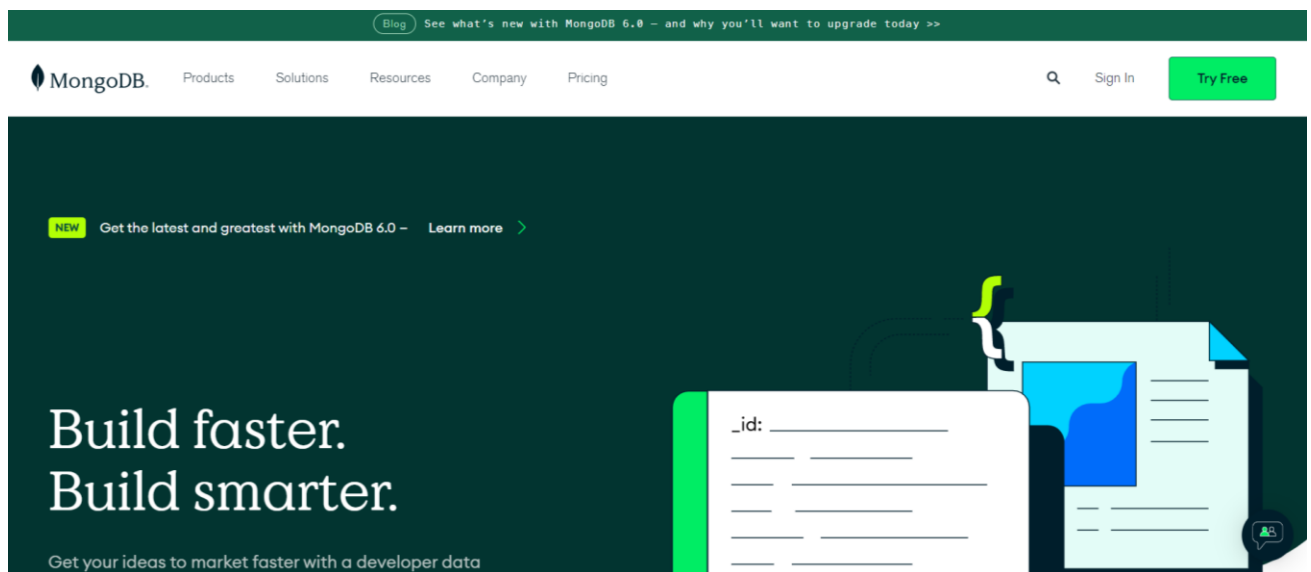
Es un editor de gráficos vectorial y una herramienta de generación de prototipos, wireframes, interfaces... Principalmente basada en la web.



MongoDB

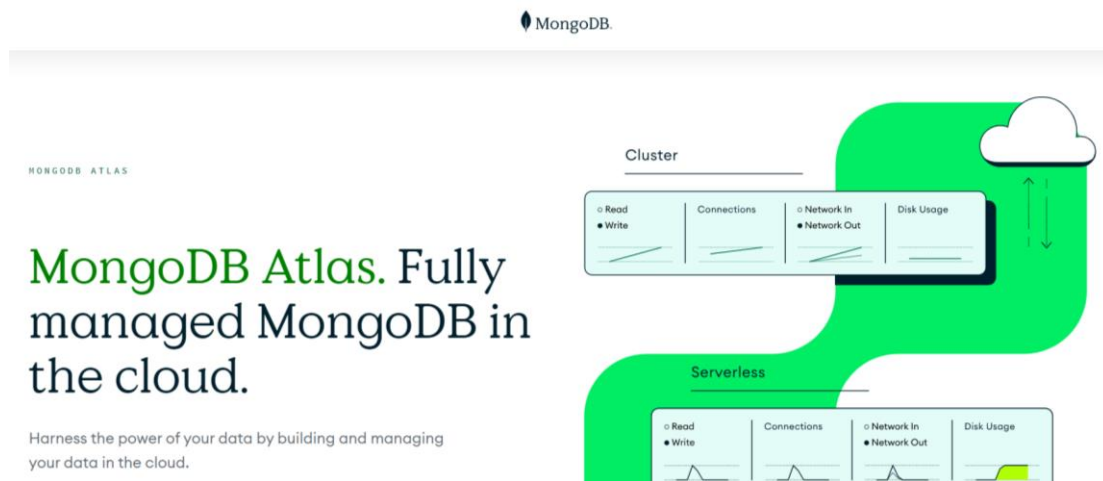
Es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

MongoDB es una base de datos adecuada para su uso en producción y con múltiples funcionalidades. Esta base de datos se utiliza mucho en la industria, contando con implantaciones en empresas como MTV Network, Craigslist, Foursquare.



Mongo Atlas

Servicio MongoDB alojado en la nube AWS, Azure y Google Cloud para implementar, operar y escalar una base de datos MongoDB



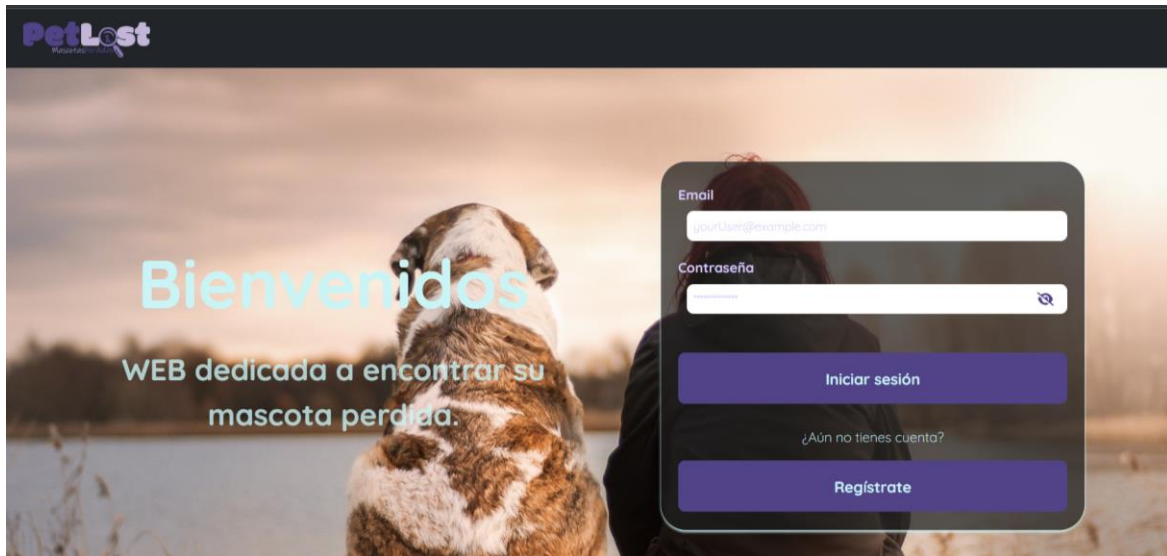
INFORME RETROSPECTIVA SEMANAL

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

1. Empecé a estructurar el backend, creando los modelos para la base de datos en Mongo. Así como también se hicieron las debidas pruebas en postman de las solicitudes planteadas. (registrar, loguear, eliminar, crear y actualizar)
2. Mejorar en los tiempos de entrega
3. Se me han ocurrido ideas para la estructuración del proyecto, diseño y tecnologías que usaremos.
4. Sin acuerdos pactados. Grupo de un solo integrante
5. Ya que se me dificulta un poco realizar todo el proyecto solo, me toca entregar lo que tengo hecho y seguir avanzando poco a poco con el backend y la conexión.

MOCKUPS

Login



The login mockup features a dark header with the 'PetLost' logo. The background is a warm-toned image of a dog's head in profile, looking out over a body of water at sunset. A semi-transparent dark modal box is positioned on the right side of the screen. Inside the modal, there are two input fields: 'Email' with the placeholder 'your.user@example.com' and 'Contraseña' with a toggle icon for password visibility. Below these fields are two purple buttons: 'Iniciar sesión' and 'Regístrate'. A link '¿Aún no tienes cuenta?' is located between the two buttons.

PetLost

Bienvenidos

WEB dedicada a encontrar su mascota perdida.

Email
your.user@example.com

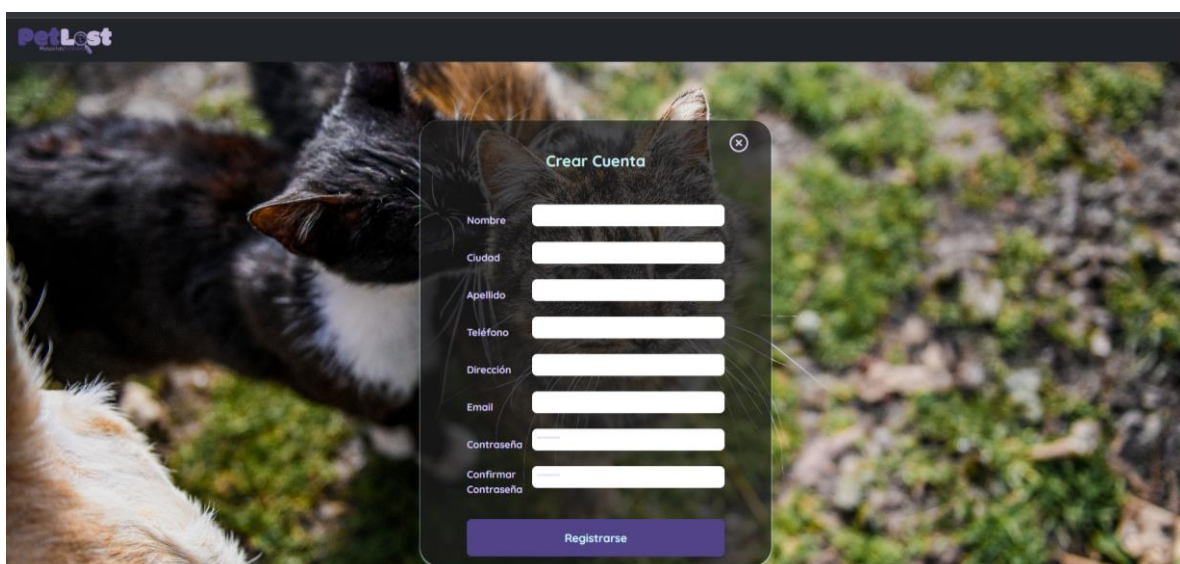
Contraseña

Iniciar sesión

¿Aún no tienes cuenta?

Regístrate

Registro



The registration mockup features a dark header with the 'PetLost' logo. The background is a close-up image of a cat's face. A semi-transparent dark modal box is centered on the screen. The modal has a title 'Crear Cuenta' with a close button in the top right corner. It contains seven input fields: 'Nombre', 'Ciudad', 'Apellido', 'Teléfono', 'Dirección', 'Email', and 'Contraseña'. Below the password field is a 'Confirmar Contraseña' field. At the bottom of the modal is a purple button labeled 'Registrarse'.

PetLost

Crear Cuenta

Nombre

Ciudad

Apellido

Teléfono

Dirección

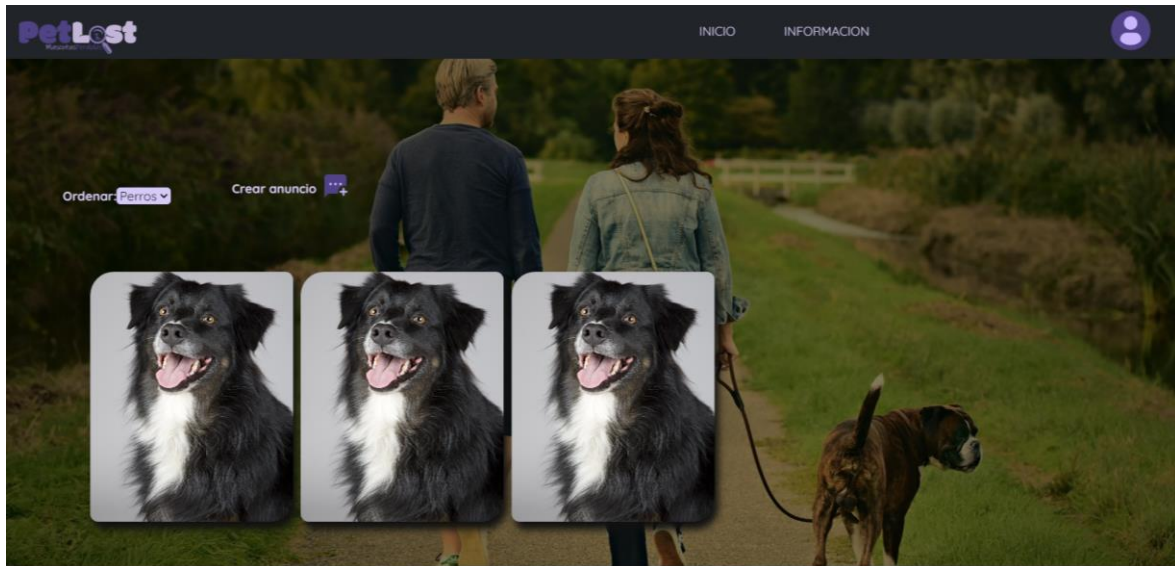
Email

Contraseña

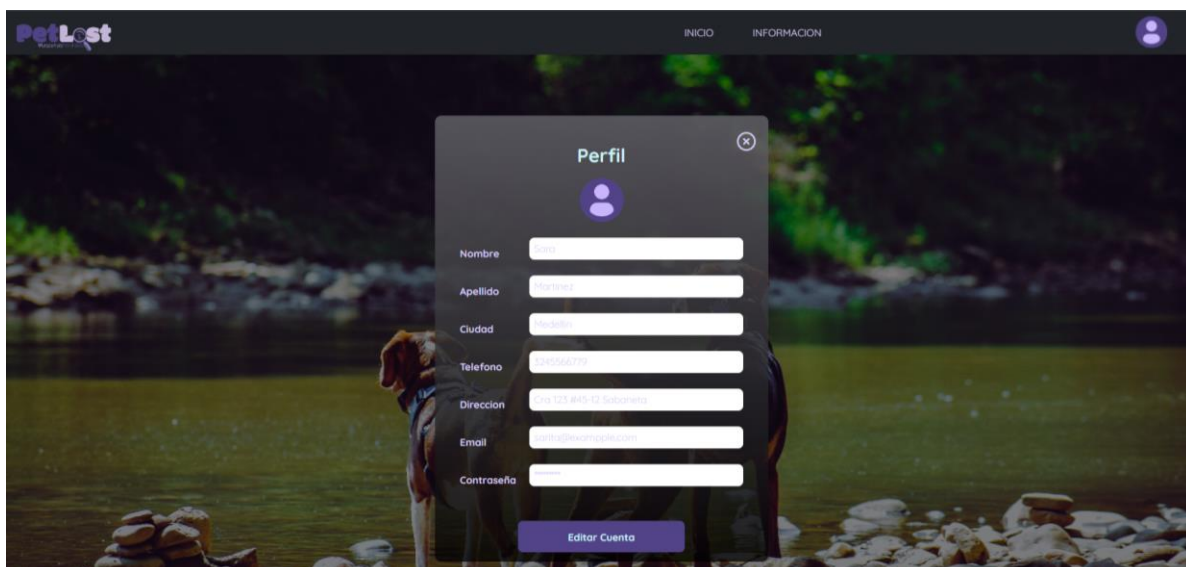
Confirmar Contraseña

Registrarse

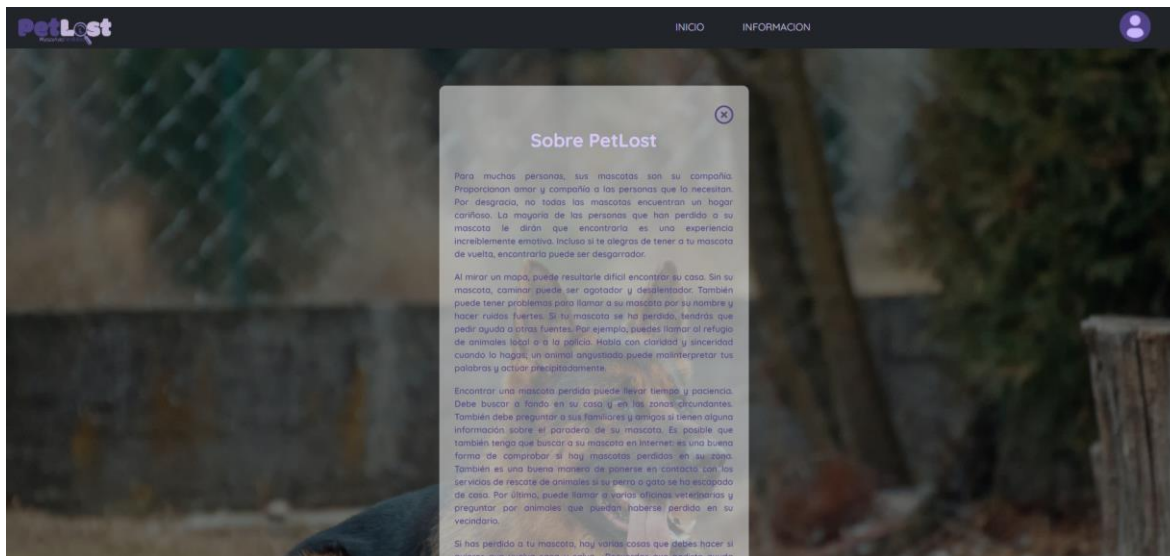
Home(ver mascotas publicadas)



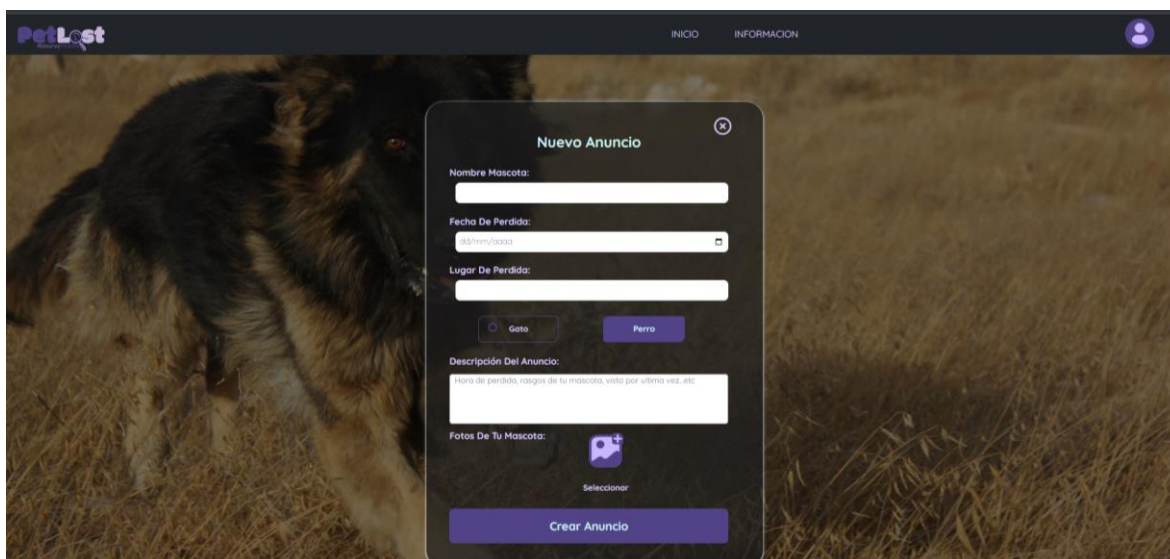
Perfil



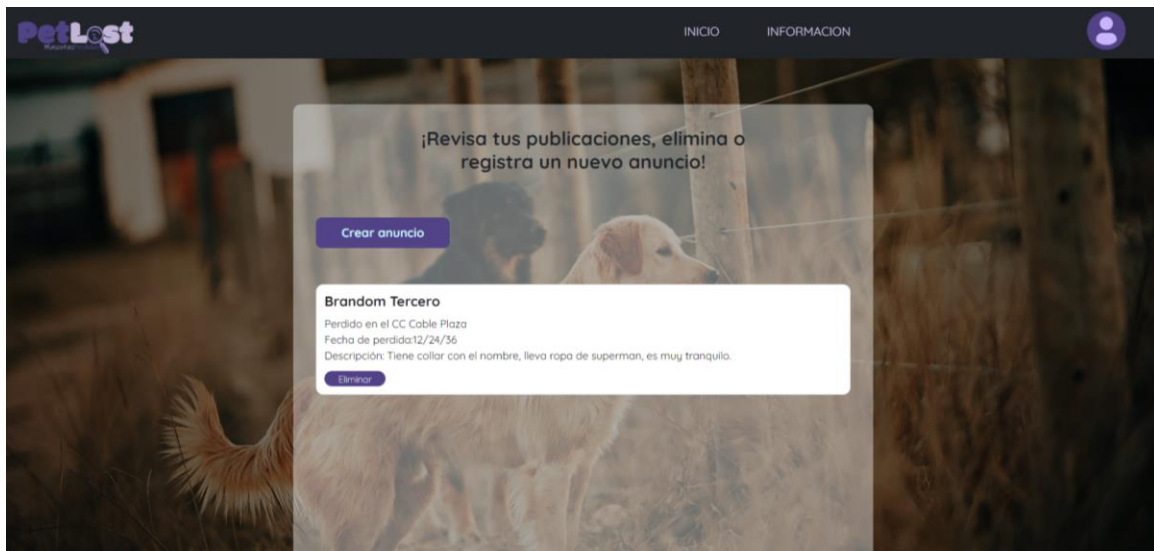
Información



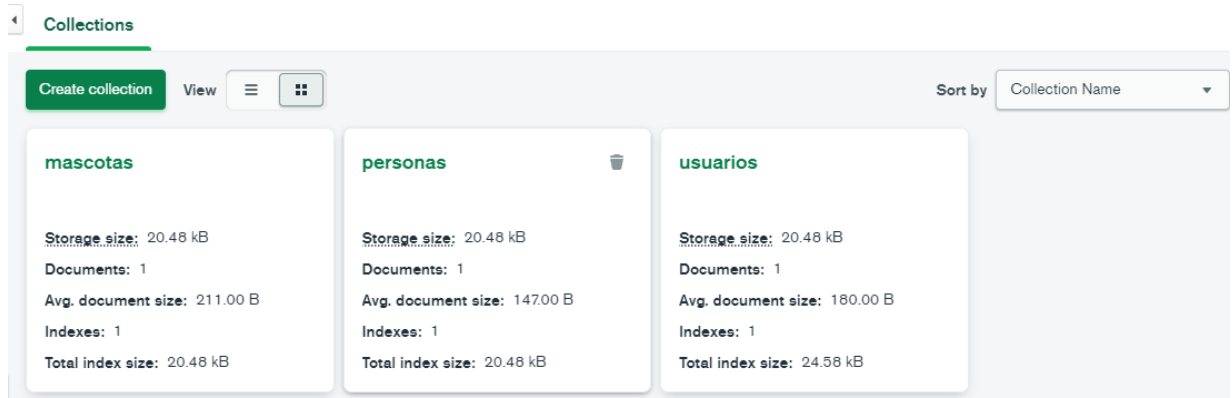
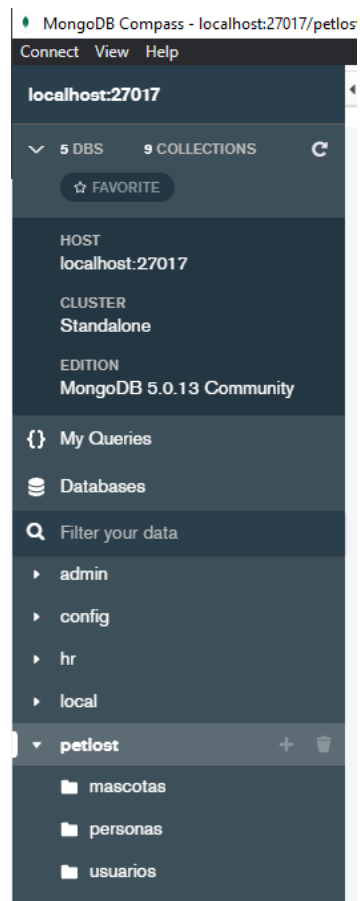
Añadir Anuncio



Anuncios Usuario



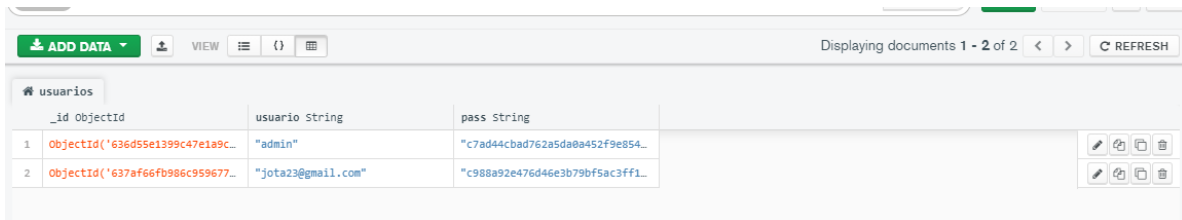
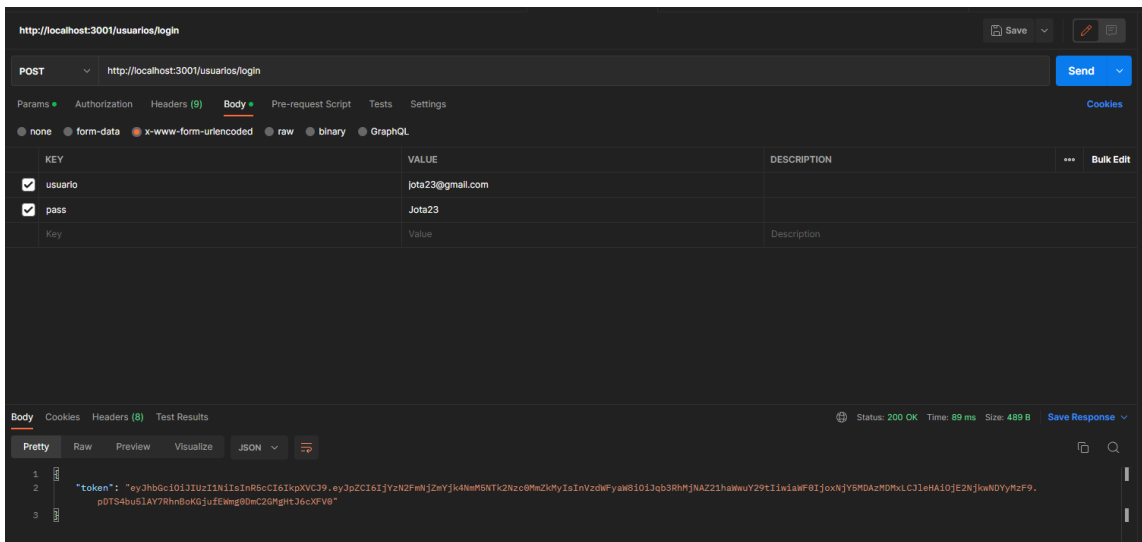
Base de datos en MongoDB Compass



TESTER(Postman)

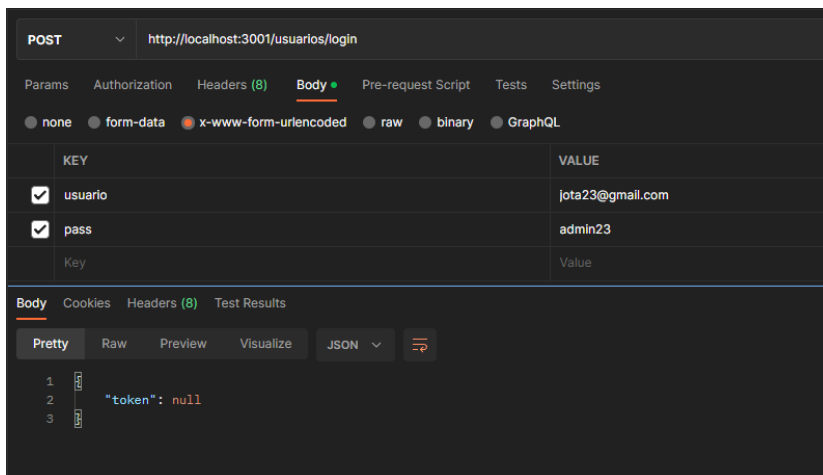
Pruebas de funcionalidades (Loguear, registrar, eliminar, crear, actualizar)

Prueba Funcional N° 1	
Nombre	Login Exitoso
Descripción	Se verifica la funcionalidad del método login del backend cuando el usuario y la contraseña son correctos
Parámetros de Entrada	*Usuario: jota23@gmail.com *Contraseña: Jota23
Resultado Esperado	Se debe recibir un token como validación del usuario y contraseña para iniciar sesion
Resultado Obtenido:	

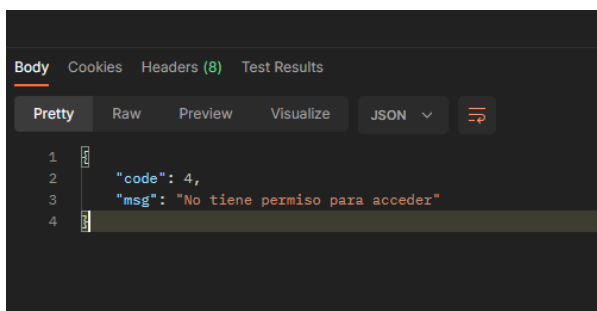


NOTA: Para poder realizar las peticiones debo estar logeado. En el postman una recibido el token del login con mi usuario registrado puedo hacer las demas consultas a traves del token bajo una autoruizacion en Headers.

Prueba Funcional N° 2	
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: jota23@gmail.com *Contraseña: admin23
Resultado Esperado	Se debe recibir un token: “null”, haciendo alusión a que el usuario y/o contraseña son incorrectos. Por lo tanto, no genera el token para iniciar sesión
Resultado Obtenido:	



NOTA: Si se realizan las peticiones en el postman sin haber hecho el login correctamente con el token muestra un mensaje code: “4” “No tiene permiso para acceder”



Prueba Funcional N° 3	
Nombre	Registrar Persona
Descripción	Se verifica la funcionalidad del método Registrar la persona en el backend
Parámetros de Entrada	* Nombres: Juan Daniel * Apellidos: Perdomo * Direccion: Calle 9ª 23 30 * Celular: 3112223344 * Ciudad: Bogota
Resultado Esperado	Se debe recibir un mensaje indicando que la persona se guardó correctamente
Resultado Obtenido:	Mensaje de éxito: true, además se evidencia en base de datos que existe el registro

POST http://localhost:3001/personas Send

Params Authorization Headers (0) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> nombres	Juan Daniel			
<input checked="" type="checkbox"/> apellidos	Perdomo			
<input checked="" type="checkbox"/> direccion	Calle 9A 23 30			
<input checked="" type="checkbox"/> celular	3112223344			
<input checked="" type="checkbox"/> ciudad	Bogota			
Key	Value	Description		

Body Cookies Headers (0) Test Results Status: 200 OK Time: 50 ms Size: 324 B Save Response

Pretty Raw Preview Visualize JSON Copy Search

```

1
2
3
4
[{"msg": "La persona se guardo correctamente",
  "exito": true}]

```



```

_id: ObjectId('637afa491b661877cc688cc5')
nombres: "Juan Daniel"
apellidos: "Perdomo"
direccion: "Calle 9A 23 30"
celular: "3112223344"
ciudad: "Bogota"
__v: 0

```

Prueba Funcional N° 4	
Nombre	Crear Mascota Perdida
Descripción	Se verifica la funcionalidad del método Crear Mascota del backend
Parámetros de Entrada	* Mascota_id: "pet_2" * Nombre: "Roko" * Foto: "roko.jpg" * Fecha Perdida: "22/11/22" * Lugar Perdida: "Parque salitre" * Descripción: "Perro pitbull Stanford gris con ojos azules"
Resultado Esperado	Se debe recibir un mensaje indicando que la mascota se guardó correctamente
Resultado Obtenido:	Mensaje de éxito: true, además se evidencia en base de datos que existe el registro.

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

- Method:** POST
- URL:** http://localhost:3001/mascotas
- Body Type:** x-www-form-urlencoded
- Form Data:**

KEY	VALUE
mascota_id	pet_2
nombre	Roko
foto	roko.jpg
fecha_perdida	22/11/22z
lugar_perdida	Parque salitre
descripcion	Perro pitbull Stanford gris con ojos azules
- Response Body (JSON):**

```

{
  "msg": "La mascota se guardo correctamente",
  "exito": true
}

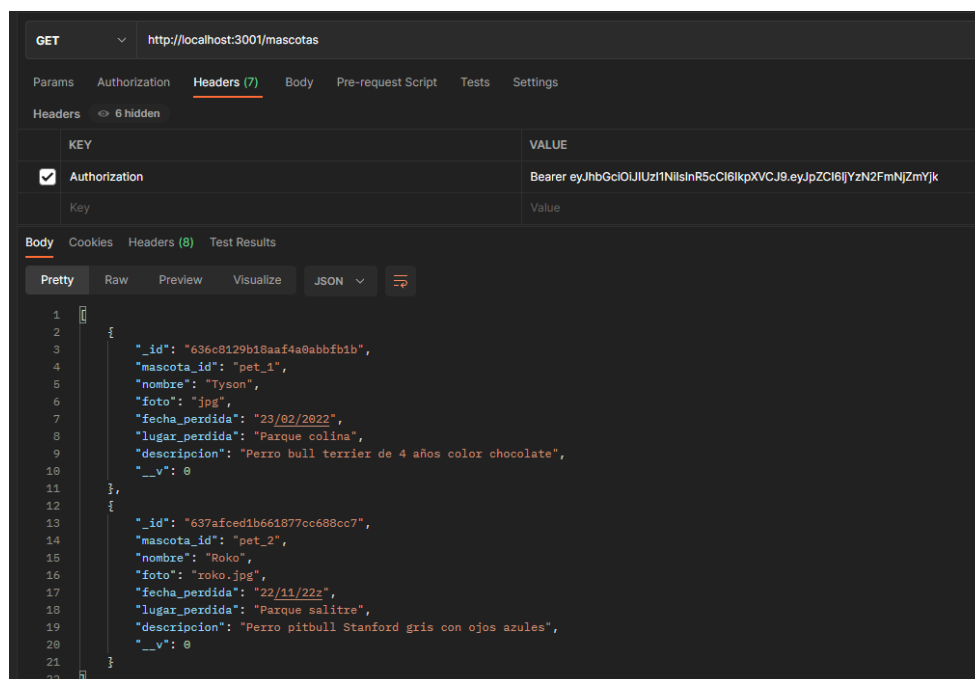
```

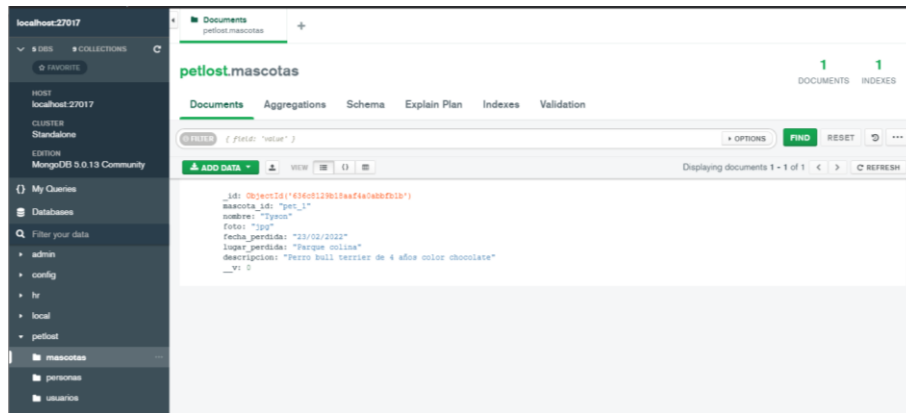
```

_id: ObjectId('637afced1b661877cc688cc7')
mascota_id: "pet_2"
nombre: "Roko"
foto: "roko.jpg"
fecha_perdida: "22/11/22z"
lugar_perdida: "Parque salitre"
descripcion: "Perro pitbull Stanford gris con ojos azules"
__v: 0

```

Prueba Funcional N° 5	
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:	Lista de mascotas perdidas





Prueba Funcional N° 7	
Nombre	Actualizar Persona
Descripción	Se verifica la funcionalidad del método update de la persona en el backend
Parámetros de Entrada	* Nombres: Daniel * Apellidos: Perdomo * Direccion: Carrera 7 23 30 * Celular: 123 * Ciudad: Bogota
Resultado Esperado	Se debe recibir un mensaje indicando que el usuario se modificó correctamente
Resultado Obtenido:	Mensaje de éxito: true, además se evidencia en base de datos la modificación de la persona

PUT ⌵ http://localhost:3001/personas/637afa491b661877cc688cc5

Params Authorization Headers (9) **Body** • Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE
<input checked="" type="checkbox"/>	nombres	Daniel
<input checked="" type="checkbox"/>	apellidos	Perdomo
<input checked="" type="checkbox"/>	direccion	Carrera 7 23 30
<input checked="" type="checkbox"/>	celular	123
<input checked="" type="checkbox"/>	ciudad	Bogota
	Key	Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ⌵ ↻

```
1
2  "msg": "El usuario se modifico correctamente",
3  "exito": true
4
```

```
_id: ObjectId('637afa491b661877cc688cc5')
nombres: "Daniel"
apellidos: "Perdomo"
direccion: "Carrera 7 23 30"
celular: "123"
ciudad: "Bogota"
__v: 0
```