



Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

Programación para Internet – Sección D03 – Ciclo 2025B

Reporte de Proyecto. Página de reconocimiento de perros y gatos.



Profesor: Michel Emanuel López Franco

Bautista Chavira Alan Gabriel

Rodríguez Macías Andrea

Gutiérrez Hernández José Rafael

3/11/2025

Este proyecto se compone de 3 servidores separados

Tenemos:

- 1.- El servidor que tiene el frontend
- 2.- El servidor que tiene backend que maneja el api y la conexión con otras apis como la de la IA
- 3.- El servidor de la base de datos.

¿Qué hace?

La utilidad principal de la página es escanear fotos de animales y distinguir entre perros y gatos. Adicionalmente como los requerimientos del proyecto incluían inicio de sesión, optamos por incluir un listado de mascotas personales, que básicamente guarda mascotas en la BD con el id del usuario que las registró como llave foránea.

El sistema tiene algunas pocas funcionalidades que no dependen del inicio de sesión, pero la consulta de imágenes con la IA y el listado de mascotas registrados requieren inicio de sesión para funcionar.

El componente de login se accede haciendo click en el botón con el logo usuario o en el texto justo enseguida de dicho botón que dice Iniciar Sesión. Al pulsarlo se renderiza el componente de login. En dicho componente hay dos funciones, Registrarse y hacer Login

Los apartados de Home, Catálogo y Curiosidades son meramente informativos y no contienen funcionalidades que reaccionen con el usuario en el sentido de que procesen información, simplemente muestran los ejemplos que hay en la BD.

Las pestañas de “Mis Mascotas” y “Consulta por Foto” son las que realmente le permiten al usuario interactuar.

Consulta por foto permite al usuario subir una imagen, al cargarla al navegador este la manda al api express que a su vez la manda por medio del api de hugging face al modelo de inteligencia artificial Microsoft Resnet-50, que es un modelo de inteligencia artificial de reconocimiento de imágenes. No esta específicamente entrenado para el reconocimiento de razas de perros o gatos, pero si cumple medianamente bien su propósito si lo acompañamos de un script para discriminar el resultado de lo que arroja.

La IA simplemente nos regresa la etiqueta con más peso dentro de lo que rescató de la imagen, es decir, si lo más reconocible de una imagen es un gato, nos retornará una etiqueta que diga que es un gato o un gato de una raza en particular.

Un script en el backend toma la etiqueta de la respuesta de la IA y la compara con un listado de razas o palabras clave más comunes entre razas de perros y gatos y hace una búsqueda parcial. Si un término de la respuesta coincide con cualquier de los de esta lista se considera mascota. Si coincide específicamente con los del arreglo de perros o gatos se le asigna si es gato o perro. Para ampliar el catálogo de razas de perros que se reconozcan como perro habría que ampliar este listado. Esto se debe a que la IA no retorna una etiqueta deseada en concreto, pero nos da la etiqueta con más peso.

```
18 }
19 |
20 const dogBreeds = [
21   'doberman pinscher', 'doberman', 'golden retriever', 'german shepherd', 'labrador retriever',
22   'bulldog', 'poodle', 'chihuahua', 'beagle', 'boxer', 'dachshund', 'rottweiler',
23   'shih tzu', 'husky', 'great dane', 'border collie', 'cocker spaniel', 'basset hound',
24   'akita', 'malinois', 'samoyed', 'terrier', 'greyhound', 'whippet',
25   'australian shepherd', 'bernese mountain dog', 'boston terrier', 'cane corso',
26   'cavalier king charles spaniel', 'dalmatian', 'french bulldog', 'havanese',
27   'irish setter', 'jack russell terrier', 'miniature schnauzer', 'newfoundland',
28   'papillon', 'pembroke welsh corgi', 'pomeranian', 'saint bernard', 'shar pei',
29   'shetland sheepdog', 'shiba inu', 'weimaraner'
30 ]
31 |
32 const catBreeds = [
33   'siamese cat', 'persian cat', 'maine coon', 'bengal cat', 'sphynx',
34   'ragdoll', 'british shorthair', 'russian blue', 'norwegian forest cat',
35   'abyssinian', 'savannah cat', 'scottish fold', 'oriental shorthair',
36   'american shorthair', 'balinese', 'birman', 'bombay', 'burmese', 'chartreux',
37   'cornish rex', 'devon rex', 'egyptian mau', 'exotic shorthair', 'havana brown',
38   'japanese bobtail', 'korat', 'manx', 'munchkin', 'ocicat', 'peterbald',
39   'selkirk rex', 'snowshoe', 'turkish angora'
40 ]
41 |
42 const generalSpecies = [
43   // Mamíferos comunes
44   'dog', 'cat', 'rabbit', 'bunny', 'hamster', 'guinea pig', 'ferret', 'mouse', 'rat', 'chinchilla', 'gerbil', 'hedgehog', 'sugar glider',
45   // Aves
46   'parrot', 'budgie', 'canary', 'cockatiel', 'lovebird', 'finch', 'macaw', 'conure', 'parakeet',
47   // Reptiles
48   'turtle', 'tortoise', 'lizard', 'gecko', 'iguana', 'snake', 'chameleon', 'bearded dragon',
49   //
50 ]
```

¿Cómo esta implementado?

La página web esta dividida en componentes, cada menú es un componente separado, el componente App siempre se esta renderizando, renderiza una barra de navegación arriba y va renderizando los demás componentes en lo que resta de la pantalla conforme se van necesitando. Los datos del usuario se los van pasando como props conforme sea necesario. Solo se usó React + Vite, y por ende node.js.

El backend está corriendo express con estas dependencias
bcrypt para hashear contraseñas de forma segura. Se usa para proteger credenciales antes de guardarlas en la base de datos.

cors para que la API sea accesible desde otros dominios como el entorno de prueba local.

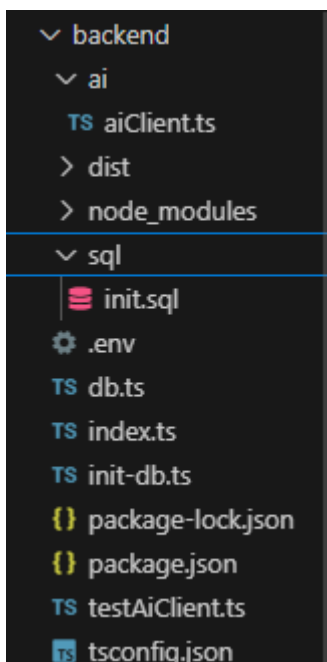
dotenv para usar claves secretas desde las variables de entorno.

express como la api rest. Maneja rutas, middleware, peticiones y respuestas.

node-fetch como cliente http para hacer peticiones desde el backend (como enviar imágenes a Hugging Face o Resend, aunque todos los demás proveedores se fresearon pero bueno).

Y pg com cliente oficial de PostgreSQL para Node.js

El backend tiene esta estructura



Donde lo más importante es el index.ts que contiene la lógica de todos los endpoints y llamadas a la base de datos, el init.sql que crea la base de datos de forma automática al desplegarse, ya que render te cobra para ver un panel de control de la BD y aiClient.ts que es el que hace las peticiones al api de la IA que usamos.

Usamos como inteligencia artificial el modelo de reconocimiento de y clasificación de imágenes Resnet-50 de Microsoft del proveedor HuggingFace. De momento la estamos usando con el plan de prueba que solo deja hacer menos de 0.1 dólares

```
frontend > src > TS config.ts > [⌘] API_BASE_URL
1  // API configuration
2  const API_BASE_URL = import.meta.env.VITE_API_URL || 'https:
3  ⚠
4  export const API_URLS = {
5    // Catalogo endpoints
6    categories: `${API_BASE_URL}/categories`,
7    breeds: `${API_BASE_URL}/breeds`,
8
9    // Auth endpoints
10   login: `${API_BASE_URL}/login`,
11   register: `${API_BASE_URL}/register`,
12   recover: `${API_BASE_URL}/recover`,
13
14   // User endpoints
15   profile: `${API_BASE_URL}/profile`,
16
17   // MisMascotas endpoints
18   pets: `${API_BASE_URL}/pets`,
19
20   // GuardarMascota
21   savepet: `${API_BASE_URL}/save-pet`,
22
23   //AnalizadorIA endpoint
24   analyzePhoto: `${API_BASE_URL}/analyze-photo`,
25
26   // Curiosidades endpoint
27   curiosidades: `${API_BASE_URL}/curiosidades`,
28 } as const;
29
```

¿Cómo se despliega?


Primero tenemos que hacer un servidor para base de datos postgres. En mi caso particular para render lo tuve que hacer así porque render no deja alojar base de datos y servicios web varios a la vez.

Una vez creamos la Base de datos deberíamos tener una URL de acceso y credenciales de usuario, estas credenciales de usuario deben ser puestas en el servidor Backend (que recordemos es el que recibe peticiones de la página del front y consulta la BD).

Todos los servicios a los que se conecta el backend deberían de estar conectados por medio de las variables de entorno para no exponer nuestras conexiones en el código, sin embargo, también puede modificar el código en líneas como ésta. No es recomendable pero también se puede

```
sync function main() {  
  const defaultDbUrl = process.env.DATABASE_URL || 'ejemplourl.com'; //Esta url hay que cambiarla  
  const targetUrl = new URL(defaultDbUrl);
```

El panel de entorno de Render nos deja poner varias variables de entorno en él. Así se debería de ver nuestro panel de entorno



The screenshot shows the 'Environment Variables' panel in Render. It includes a title, a description, an 'Export' button, and a table of variables.

KEY	VALUE
BCRYPT_ROUND	*****
DATABASE_URL	*****
HF_TOKEN	*****
PEPPER_SECRET	*****

Con la Api Key para conectarnos con el api de la IA de HuggingFace, La URL para la BD (que ya debe incluir en ella las credenciales de usuario), y las variables para el uso de hasheo y seguridad.

Cabe destacar que cambiar de modelo de inteligencia artificial implicaría potencialmente cambiar la forma en la que se maneja la respuesta en el “aiClient.ts” y/o el archivo con los endpoints de expres “index.ts”

Una vez creado el servidor de la BD y obtenidas las URL y Api Keys, procedemos a desplegar el servidor Backend, para desplegarlo simplemente despliega desde Render con las cinco variables de entorno ya puestas. El servidor arranca solo gracias a los archivos de configuración y dependencias. Solo tienes que desplegar a partir del directorio Backend del repositorio de github en vez de desplegar desde la raíz. Backend, no root.

El servidor Backend tiene un archivo “init.sql” que contiene las sentencias para crear la estructura de las tablas en la base de datos suficiente para que le proyecto

funcione. Si no es la primera vez que despliegas el back, verás un mensaje en consola con algo así como “Las tablas ya están creadas” dando a entender que se intentaron volver a crear, pero no fue necesario.

Una vez dado de alta el back ya solo falta desplegar el servidor frontend con la página web de react. De igual forma debería poder desplegarse con los comandos de siempre, “npm install” y “npm start” de forma automática, solamente tienes que cambiar el directorio de despliegue de root a Frontend en la configuración de despliegue sobre repositorios de github.

El frontend también puede hacer uso de una variable de entorno, o también puedes poner la url después del || , (no recomendable)

```
// API configuration
const API_BASE_URL = import.meta.env.VITE_API_URL ||
```

En resumidas cuentas, el despliegue consiste en
Crear una BD en render.

Crear un servidor para servicios web en render, poner las variables de entorno que requiere el back (key de la ia, salt, bcrypt rounds, db url).

Desplegar el backend.

Crear un servidor para página web en render, poner la url de la api como variable de entorno.

Desplegar el front.

Pruebas de la página en funcionamiento

Iniciar sesión

Nombre de usuario

alan

Contraseña

.....

Entrar

¿No tienes cuenta?

Regístrate

Descripción: Gato mamañuevo

Iniciar sesión

Usuario o contraseña incorrectos.

Nombre de usuario

alan

Contraseña

.....

Entrar

¿No tienes cuenta?

Regístrate

El...

Registro



El correo electrónico no tiene un formato válido.

Nombre de usuario

Lisa

Correo electrónico

monalisa@gmail.com

Contraseña

Crear cuenta

[Volver a entrar](#)

Mis Mascotas

Carlos

Raza: persian cat

Descripción: No se me ocurre nada la verdad

Eliminar

Otto

Raza: eskimo dog, husky

Descripción: Gato mamagüevo

Eliminar

AlanJr

Raza: doberman, doberman pinscher

Descripción: es un perro

Eliminar





Resultado de la IA

Tu mascota es: doberman, doberman pinscher

Confianza: 100.0%

Especie detectada: Perro

Guardar en Mis Mascotas

[illegible]

Subir

Vista previa



Resultado de la IA

Tu mascota es: siamese cat, siamese

Clasificación: Sí es una mascota

Confianza: 100.0%

Especie detectada: Gato

Guardar en Mis Mascotas

Siames

Gato siames color blanco con marrón oscuro y negro

Guardar mascota

Mascota guardada correctamente

Mis Mascotas

Consulta por Foto

Mis Mascotas

Siames

Raza: siamese cat, siamese

Descripción: Gato siames color blanco con marrón oscuro y negro

Eliminar

Carlos

Raza: persian cat

Descripción: No se me ocurre nada la verdad

Eliminar

Otto

Raza: eskimo dog, husky

Descripción: Gato mamagüevo

Eliminar

AlanJr

Raza: doberman, doberman pinscher

Descripción: es un perro

Eliminar

Ir a Consulta por Foto