

DOGOBOOK

INTEGRANTES:

- Nayalem Karim Arune Chaua
- Marccelo Farid Tito Lezano
- Amara Barrera
- Camila Andrea Yanela Huaila Luján

AGENDA

01 app.py

02 templates

03 static

04 Base de datos

05 GraphQL



APP.PY

Inicializa Flask y la configura para conectarse a una base de datos MySQL, además se crea la base de datos conectando con index.html y pedidos.html

```
import logging
import os
from flask import Flask, render_template, request, redirect, url_for
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy.exc import SQLAlchemyError
from werkzeug.utils import secure_filename
from flask_graphql import GraphQLView
import graphene
from graphene_sqlalchemy import SQLAlchemyObjectType, SQLAlchemyConnection

ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg', 'gif'}

def allowed_file(filename):
    return '.' in filename and \
           filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

# Configuración del registro (logging)
logging.basicConfig(level=logging.DEBUG)

# Configuración de Flask y SQLAlchemy para MySQL
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql+pymysql://root:@localhost:3306'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

# Configuración para la carga de archivos
app.config['UPLOAD_FOLDER'] = 'static/uploads'
app.config['MAX_CONTENT_LENGTH'] = 16 * 1024 * 1024 # 16 MB

# Inicialización de SQLAlchemy
db = SQLAlchemy(app)

# Definición del modelo SQLAlchemy para Registro y Perro
class Registro(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    nombre = db.Column(db.String(100), nullable=False)
    telefono = db.Column(db.String(20), nullable=False)

class Perro(db.Model):
    __tablename__ = 'perritos'
    id = db.Column(db.Integer, primary_key=True)
    nombre = db.Column(db.String(100), nullable=False)
    raza = db.Column(db.String(100), nullable=False)
    edad = db.Column(db.Integer, nullable=False)
    descripcion = db.Column(db.Text, nullable=False)
    imagen = db.Column(db.String(255), nullable=False)

    def __repr__(self):
        return f'<Perro {self.id>: {self.nombre}>'

# Definición de tipos de objeto GraphQL para SQLAlchemy
class RegistroObject(SQLAlchemyObjectType):
    class Meta:
        model = Registro
        interfaces = (graphene.relay.Node, )

class PerroObject(SQLAlchemyObjectType):
    class Meta:
        model = Perro
        interfaces = (graphene.relay.Node, )

# Definición de esquema GraphQL
class Query(graphene.ObjectType):
    registros = SQLAlchemyConnectionField(RegistroObject)
    perros = SQLAlchemyConnectionField(PerroObject)

schema = graphene.Schema(query=Query)

# Crear la tabla si no existe
with app.app_context():
    db.create_all()

# Rutas y lógica de la aplicación
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/adopcion')
def adopcion():
    perros = Perro.query.all()
    return render_template('adopcion.html', perros=perros)

@app.route('/agregar_perro', methods=['GET', 'POST'])
def agregar_perro():
    if request.method == 'POST':
        nombre = request.form['nombre']
        raza = request.form['raza']
        edad = request.form['edad']
        descripcion = request.form['descripcion']
        imagen = request.files['imagen'] # Acceder al archivo de imagen

        if nombre and raza and edad and descripcion and imagen:
            try:
                # Guardar la imagen en la carpeta de uploads
                if imagen and allowed_file(imagen.filename):
                    filename = secure_filename(imagen.filename)
                    imagen.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))

                # Crear un nuevo perro con la información
                nuevo_perro = Perro(nombre=nombre, raza=raza, edad=edad, descripcion=descripcion, imagen=filename)
                db.session.add(nuevo_perro)
                db.session.commit()
                logging.debug("Nuevo perro añadido correctamente")
                return redirect(url_for('adopcion'))
            except SQLAlchemyError as e:
                db.session.rollback()
                logging.error(f"Error al añadir un nuevo perro: {e}")
                return render_template('agregar_perro.html', error="Ocurrió un error al añadir el perro. Intentalo de nuevo más tarde.")
            except Exception as e:
                logging.error(f"Error inesperado al añadir perro: {e}")
                return render_template('agregar_perro.html', error="Ocurrió un error inesperado. Intenta de nuevo más tarde.")
        else:
            return render_template('agregar_perro.html', error="Por favor, completa todos los campos.")
    else:
        return render_template('agregar_perro.html')

#-----
@app.route('/eliminar_perro/<int:perro_id>', methods=['POST'])
def eliminar_perro(perro_id):
    perro = Perro.query.get_or_404(perro_id)
    try:
        db.session.delete(perro)
        db.session.commit()
        logging.debug(f"Perro con ID {perro_id} eliminado correctamente")
        perros_actualizados = Perro.query.all()
        return render_template('adopcion.html', perros=perros_actualizados)
    except SQLAlchemyError as e:
        db.session.rollback()
        logging.error(f"Error al eliminar el perro con ID {perro_id}: {e}")
        return render_template('adopcion.html', perros=Perro.query.all(), error="Ocurrió un error al eliminar el perro.")
    except Exception as e:
        logging.error(f"Error inesperado al eliminar perro: {e}")
        return render_template('adopcion.html', perros=Perro.query.all(), error="Ocurrió un error inesperado.")

# Configuración de GraphQL
app.add_url_rule('/graphql', view_func=GraphQLView.as_view('graphql', schema=schema, graphiql=True))

# Punto de entrada para ejecutar la aplicación Flask
if __name__ == '__main__':
    app.run(debug=True, port=8080)
```



Index.html

Este es la base del proyecto, aquí desde aquí se ramificara gran parte del código, ya sean diversas imágenes y la introducción de nuestra pagina.

```
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>DogoBook</title>
  <link href="{{ url_for('static', filename='ayudante.css') }}" rel="stylesheet" type="text/css">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='engine0/style.css') }}" />
  <script type="text/javascript" src="{{ url_for('static', filename='engine0/jquery.js') }}"></script>
</head>
<body>
  <header>
    <h1>DogoBook</h1>
  </header>

  <div class="ws_shadow"></div>
  <div style="background-color: ■antiquewhite; padding: 15px;">
    <ul>
      <li><a href="#caja02">SOBRE NOSOTROS</a></li>
      <li><a href="#caja06" name="caja06">GALERIA</a></li>
      <li><a href="#caja18">INFORMACION</a></li>
      <li><a href="{{ url_for('pedidos') }}">PEDIDOS</a></li>  <!-- Corrección del enlace a la página 1>
    </ul>
  </div>
```

```
<div id="caja10">
  <div id="caja02">
    <p>¡Bienvenidos a nuestro refugio virtual de adopción de perros! Aquí, cada latido cuenta.</p>
    <p>Nuestro objetivo es encontrar un hogar amoroso para cada uno de nuestros amigos de cuatro patas. Cada perro t.
    <p>Además, ofrecemos apoyo continuo a nuestros adoptantes para asegurarnos de que la transición sea lo más suave
    <p>Navega por nuestras páginas llenas de rostros peludos y ojos brillantes, cada uno esperando a su humano espec.
  </div>

  <div id="caja11">
    
  </div>
</div>

<div id="caja06">Galeria</div>
<div class="cajas">
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</div>

<div id="caja18">Información</div>
<div id="caja16">
  <video id="video01" style="width: 600px;" controls loop>
    <source src="{{ url_for('static', filename='video01.mp4') }}" type="video/mp4">
  </video>
</div>
<div></div>
</body>
</html>
```


Confirmacion.html

```
confirmacion.html X
Adopcion > templates > confirmacion.html > ...

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Confirmación de Pedido</title>
6   <style>
7     body {
8       font-family: Arial, sans-serif;
9       background-color: #f0f0f0;
10      margin: 0;
11      padding: 0;
12    }
13
14    .container {
15      width: 80%;
16      margin: 20px auto;
17      background-color: #fff;
18      padding: 20px;
19      border-radius: 8px;
20      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
21    }
22
23    h1 {
24      color: #333;
25      text-align: center;
26    }
27
28    p {
29      font-size: 18px;
30      line-height: 1.6;
31      margin-bottom: 20px;
32    }
33
34    a {
35      display: block;
36      text-align: center;
37      text-decoration: none;
38      color: #007bff;
39      margin-top: 20px;
40    }
41  </style>
42 </head>
43 <body>
44   <div class="container">
45     <h1>Confirmación de Pedido</h1>
46     {% if success %}
47       <p>¡Gracias, {{ nombre }}! Tu pedido de adopción para el teléfono {{ telefono }} ha sido recibido correctamente.</p>
48     {% else %}
49       <p>Hubo un problema al procesar tu pedido. Por favor, intenta de nuevo más tarde.</p>
50     {% endif %}
51     <a href="/">Volver a la página de inicio</a>
52   </div>
53 </body>
```

Se muestra la confirmación de un pedido de adopción, con mensajes de éxito o error según el resultado.

Realizar Pedido de Adopción

Nombre:

Nayalem Arune

Teléfono:

951297522

Formato: 9 dígitos (sin guiones)

958512621

9512975221

agregar_perro.html

```

agregar_perro.html X
templates > agregar_perro.html > ...
1 <!-- agregar_perro.html -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5   <meta charset="UTF-8">
6   <title>Agregar Perro</title>
7   <link href="{{ url_for('static', filename='estilo.css') }}" rel="stylesheet" type="text/css">
8 </head>
9 <body>
10   <header>
11     <h1>Agregar Nuevo Perro</h1>
12     <a href="{{ url_for('adopcion') }}">Volver a la Lista de Perros en Adopción</a>
13   </header>
14
15   <div class="cajas">
16     <form action="{{ url_for('agregar_perro') }}" method="post" enctype="multipart/form-data">
17       <label for="nombre">Nombre:</label>
18       <input type="text" id="nombre" name="nombre" required><br><br>
19
20       <label for="raza">Raza:</label>
21       <input type="text" id="raza" name="raza" required><br><br>
22
23       <label for="edad">Edad:</label>
24       <input type="number" id="edad" name="edad" required><br><br>
25
26       <label for="descripcion">Descripción:</label><br>
27       <textarea id="descripcion" name="descripcion" rows="4" required></textarea><br><br>
28
29       <label for="imagen">Imagen:</label>
30       <input type="file" id="imagen" name="imagen" accept="image/*" required><br><br>
31
32       <button type="submit">Agregar Perro</button>
33     </form>
34   </div>
35 </body>
36 </html>
37
38

```

Podremos agregar una nueva raza de perro con su respectiva edad, descripción e imagen con GraphQL

adopcion.html

PSe podrá adoptar un nuevo perro y/o agregar una raza de perro, además de vincularlo al archivo principal app.py

adopcion.html X

templates > adopcion.html > ...

```
1 <!-- adopcion.html -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <title>Lista de Perros en Adopción</title>
7     <link href="{{ url_for('static', filename='estilos1.css') }}" rel="stylesheet" type="text/css">
8 </head>
9 <body>
10     <header>
11         <h1>Perros en Adopción</h1>
12         <a href="{{ url_for('agregar_perro') }}">Agregar Nuevo Perro</a>
13     </header>
14
15     <div class="cajas">
16         {% if error %}
17             <div class="error">{{ error }}</div>
18         {% endif %}
19
20         {% for perro in perros %}
21             <div class="perro">
22                 <h2>{{ perro.nombre }}</h2>
23                 <p>Raza: {{ perro.raza }}</p>
24                 <p>Edad: {{ perro.edad }} años</p>
25                 <p>{{ perro.descripcion }}</p>
26                 
27                 <form action="{{ url_for('eliminar_perro', perro_id=perro.id) }}" method="post">
28                     <button type="submit">Eliminar</button>
29                 </form>
30             </div>
31         {% endfor %}
32     </div>
33 </body>
34 </html>
35
36
37
```

**MUCHAS
GRACIAS**