

DIPLOMADO DE ACTUALIZACION EN NUEVAS TEGNOLOGIAS PARA EL DESARROLLO DE SOFTWARE.

UNIDAD 3 TALLER FRONTEND

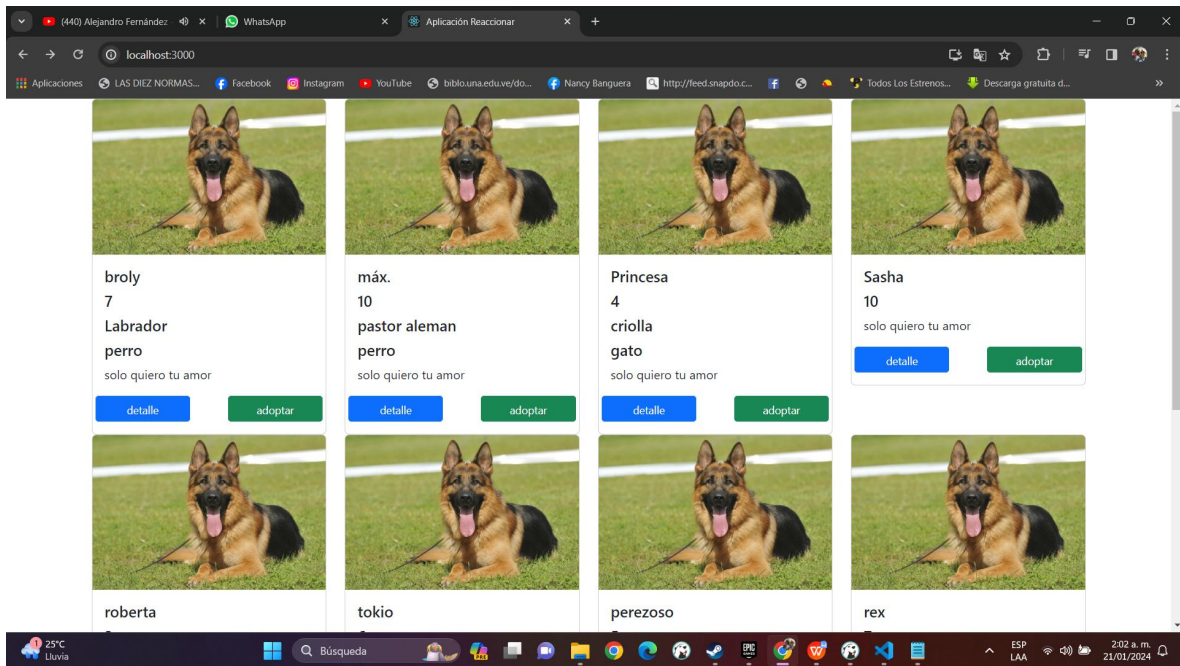
REVISADO POR: VICENTE AUX

ENTREGADO POR: HECTOR SALAZAR

UNIVERSIDAD DE NARIÑO

## Frontend

Para esta aplicación web creamos un proyecto Fron-End el cual consiste en una interfaz amigable con el cliente en esta vamos a hacer el complemento del Back-End ya trabajado el que consiste en un portal para adoptar mascotas

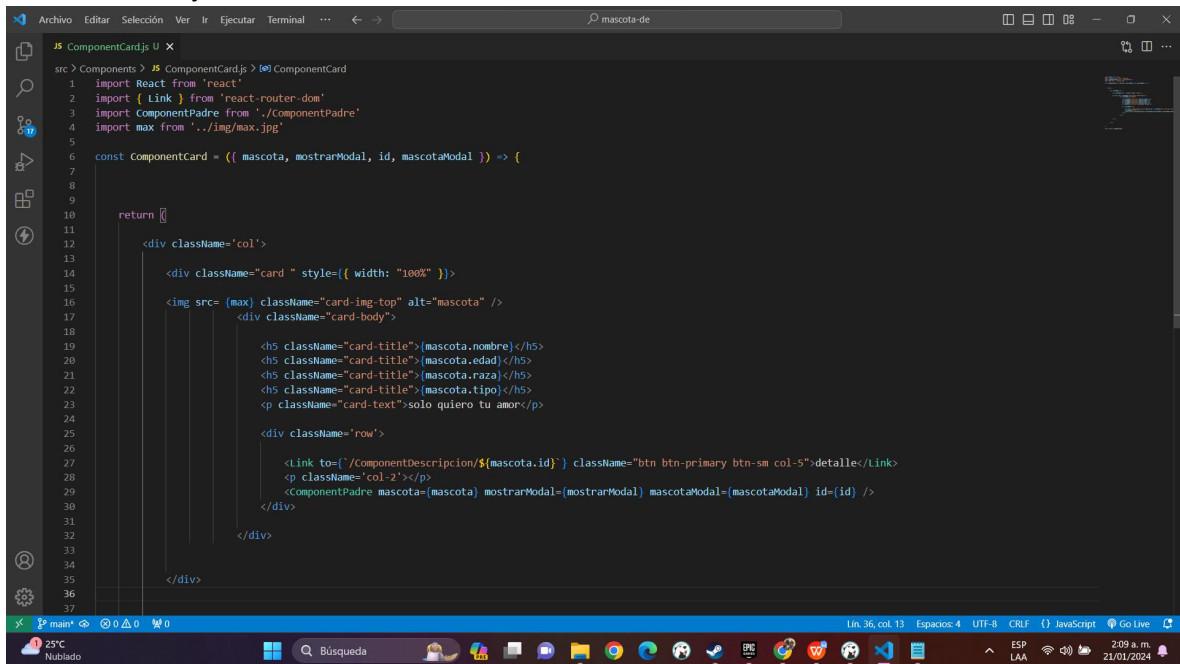


## Pantalla de inicio

Esta solo requería las tarjetas con los diferentes ejemplares para considerar una adopción para lograr este render o interfaz contamos con unos componentes los cuales se complementan entre si en especial el componente `ComponentCard`

## Componentcard

En este hacemos la creacion de la tarjeta a mostrar el cual por medio de una consulta muestra la informacion alojada en nuestra base de datos

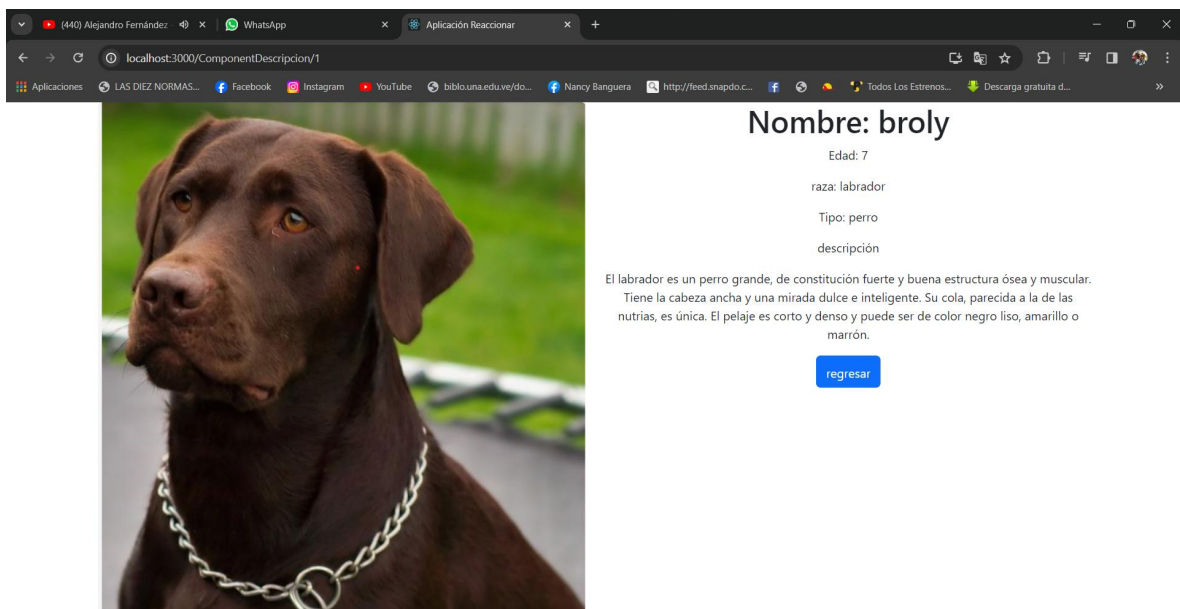


```
1 import React from 'react'
2 import { Link } from 'react-router-dom'
3 import ComponentPadre from './componentPadre'
4 import max from '../img/max.jpg'
5
6 const ComponentCard = ({ mascota, mostrarModal, id, mascotaModal }) => {
7
8
9
10  return ()
11
12    <div className="col">
13
14      <div className="card" style={{ width: "100%" }}>
15
16        <img src={max} className="card-img-top" alt="mascota" />
17        <div className="card-body">
18
19          <h5 className="card-title">{mascota.nombre}</h5>
20          <h5 className="card-title">{mascota.edad}</h5>
21          <h5 className="card-title">{mascota.raza}</h5>
22          <h5 className="card-title">{mascota.tipo}</h5>
23          <p className="card-text">solo quiero tu amor</p>
24
25          <div className="row">
26
27            <Link to={"/ComponentDescripcion/${mascota.id}"} className="btn btn-primary btn-sm col-5">detalle</Link>
28            <p className="col-2"></p>
29            <ComponentPadre mascota={mascota} mostrarModal={mostrarModal} mascotaModal={mascotaModal} id={id} />
30          </div>
31
32        </div>
33
34      </div>
35
36    </div>
37  }
```

## ComponentDescripcion

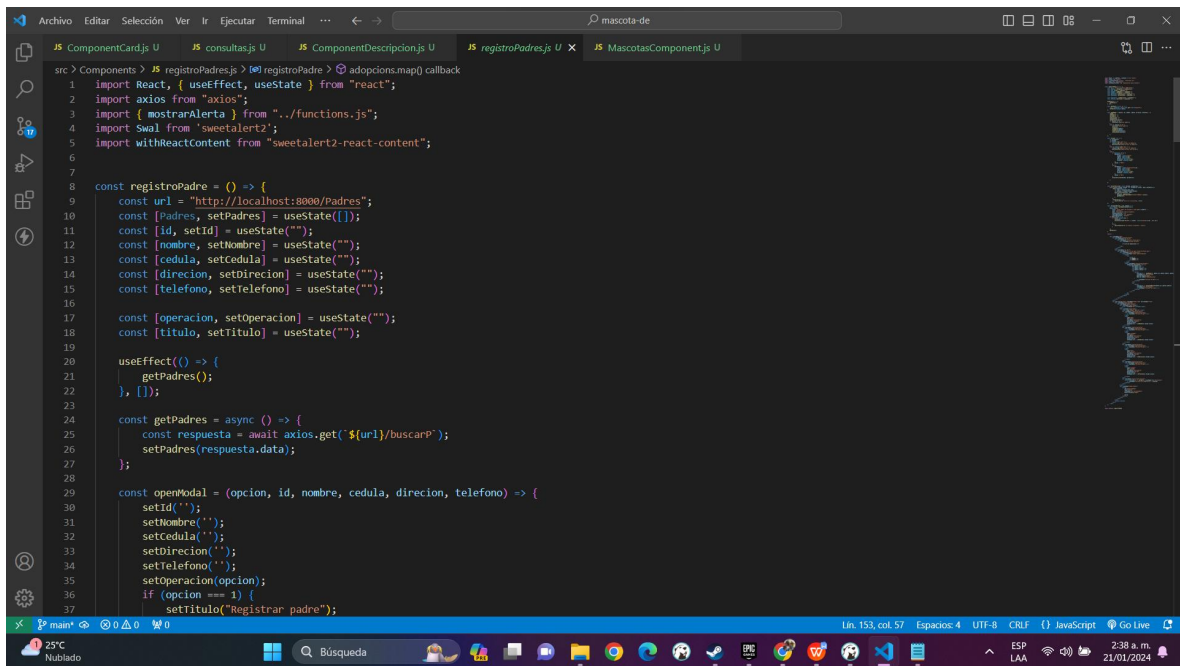
Este consta de la informacion detallada de un ejemplar por medio de su id “identificador unico” hacemos la busqueda de solo su informacion para este usamos las funciones predeterminadas de react useState y useEffect las cuales consisten en guardar el vector de la informacion requerida y de cierta manera cargarlo en forma de vector para asi facilitar su muestreo

```
Archivo Editar Selección Ver Ir Ejecutar Terminal mascota-de
ComponentCard.js U consultas.js U ComponentDescription.js U MascotasComponent.js U
src > Components > ComponentDescription.js > ComponentDescription
1 import React, { useEffect, useState } from 'react'
2 import { Link, useParams } from 'react-router-dom'
3 import { getMascotasById } from './consultas'
4 import lalimagen from '../img/broly.png'
5
6 const ComponentDescription = () => {
7
8
9   const { id } = useParams()
10
11   const [mascota, setMascota] = useState({})
12
13   const cargarMascota = async () => {
14     const res = await getMascotasById(id)
15     setMascota(res.data)
16   }
17
18   useEffect(() => {
19     cargarMascota()
20   }, [])
21
22   return (
23     <div className="container">
24       <div className="row">
25
26         <div class="text-center col-6">
27           <img src={lalimagen} width={"100%"} class="rounded" alt="..." />
28         </div>
29         <div className="text-center col-6">
30           <h1>nombre: {mascota.nombre}</h1>
31           <p>edad: {mascota.edad}</p>
32           <p>raza: {mascota.raza}</p>
33           <p>tipo: {mascota.tipo}</p>
34           <p>descripcion: {mascota.descripcion}</p>
35         </div>
36       </div>
37     </div>
38   )
39 }
```



## RegistroPadre

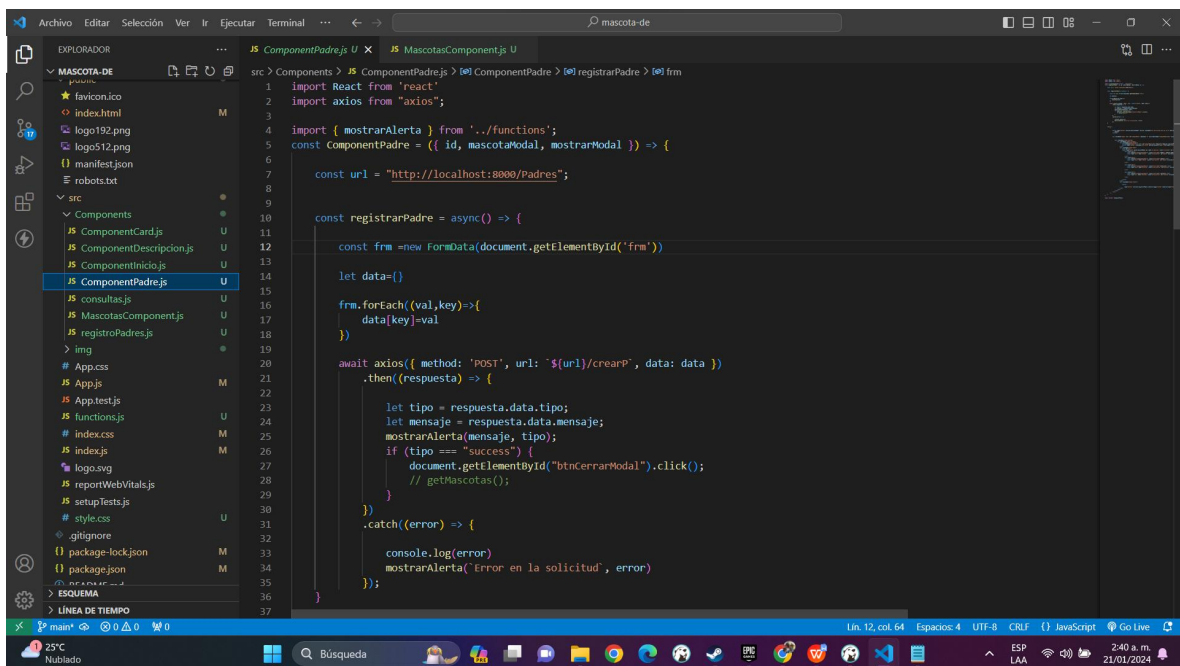
En este se hace una conexión directa a una tabla de la base de datos para hacer un registro del cliente interesado en hacer una adopción. Esta conveja los métodos necesarios para capturar la información escrita en el formulario como crear una función creada desde el backend y enrutada entre sí para cumplir con dicho proceso.



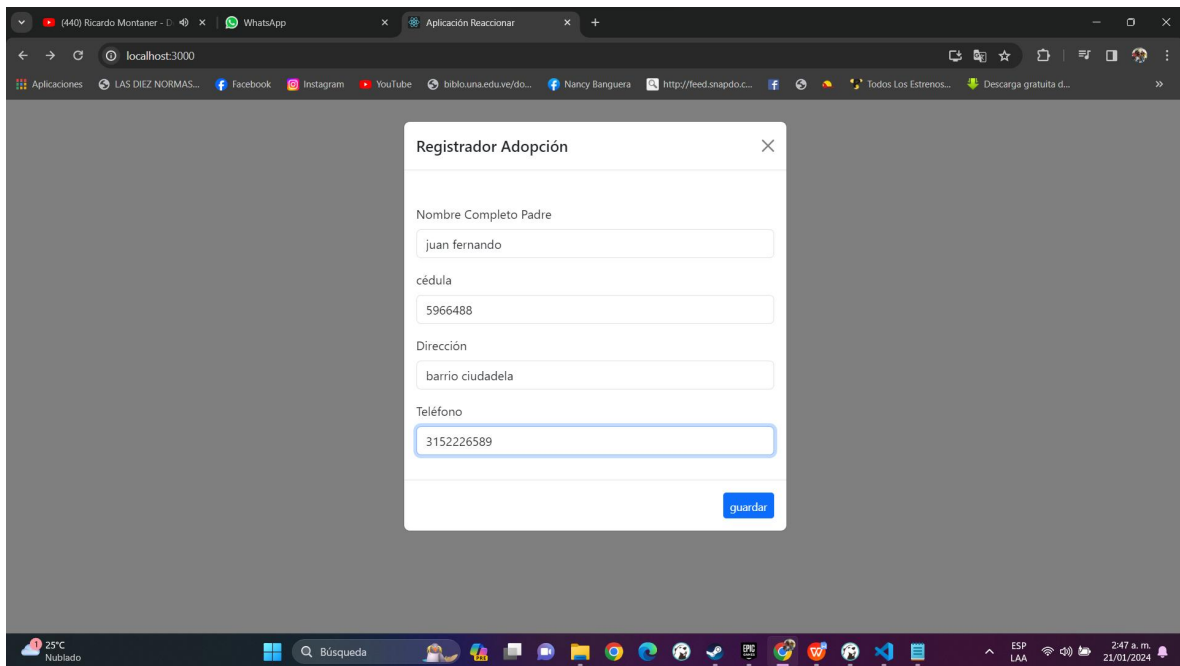
```
src > Components > JS registroPadres.js > [0] registroPadre > [0] adopciones.map() callback
1 import React, { useEffect, useState } from 'react';
2 import axios from 'axios';
3 import { mostrarAlerta } from '../functions.js';
4 import Swal from 'sweetalert2';
5 import withReactContent from 'sweetalert2-react-content';
6
7
8 const registroPadre = () => {
9   const url = "http://localhost:8000/Padres";
10  const [Padres, setPadres] = useState([]);
11  const [id, setId] = useState("");
12  const [nombre, setNombre] = useState("");
13  const [cedula, setCedula] = useState("");
14  const [direccion, setDireccion] = useState("");
15  const [telefono, setTelefono] = useState("");
16
17  const [operacion, setOperacion] = useState("");
18  const [titulo, setTitulo] = useState("");
19
20  useEffect(() => {
21    getPadres();
22  }, []);
23
24  const getPadres = async () => {
25    const respuesta = await axios.get(`${url}/buscarP`);
26    setPadres(respuesta.data);
27  };
28
29  const openModal = (opcion, id, nombre, cedula, direccion, telefono) => {
30    setId("");
31    setNombre("");
32    setCedula("");
33    setDireccion("");
34    setTelefono("");
35    setOperacion(opcion);
36    if (opcion === 1) {
37      setTitulo("Registrar padre");
38    }
39  };
40 }
```

## ComponentPadre

Este es el complemento directo de registroPadre ya que este despliega el formulario que se requiere para dicho registro



```
src > Components > JS ComponentPadre.js > [0] ComponentPadre > [0] registrarPadre > [0] frm
1 import React from 'react';
2 import axios from 'axios';
3
4 import { mostrarAlerta } from '../functions.js';
5 const ComponentPadre = ({ id, mascotaModal, mostrarModal }) => {
6
7   const url = "http://localhost:8000/Padres";
8
9
10  const registrarPadre = async () => {
11
12    const frm = new FormData(document.getElementById('frm'))
13
14    let data = {}
15
16    frm.forEach((val, key) => {
17      data[key] = val
18    })
19
20    await axios({ method: 'POST', url: `${url}/crearP`, data: data })
21      .then((respuesta) => {
22
23        let tipo = respuesta.data.tipo;
24        let mensaje = respuesta.data.mensaje;
25        mostrarAlerta(mensaje, tipo);
26        if (tipo === "success") {
27          document.getElementById("btnCerrarModal").click();
28          // getMascotas();
29        }
30      })
31      .catch((error) => {
32        console.log(error)
33        mostrarAlerta("Error en la solicitud", error)
34      });
35    });
36  }
37 }
```



## Consultas

En este componente se puede apreciar las consultas requeridas para el proceso de observar los atributos guardados en la base de datos

