

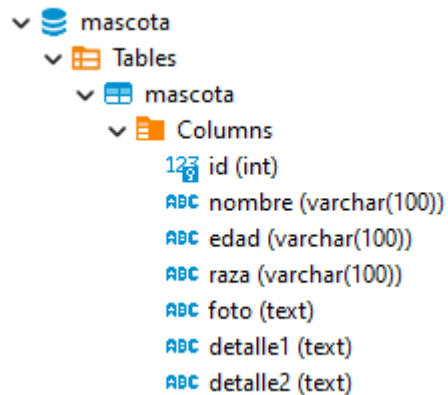
UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
DIPLOMADO DE ACTUALIZACIÓN EN NUEVAS TECNOLOGÍAS PARA EL
DESARROLLO DE SOFTWARE
VICENTE AUX REVELO

YASMANI ALEXANDER TAPIA CÓDIGO: 215036059

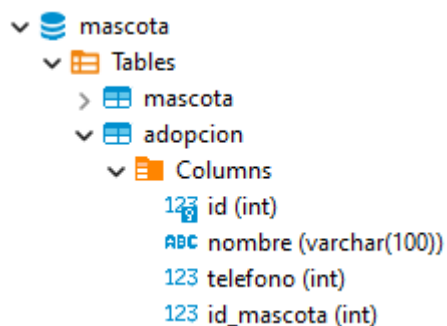
Taller Unidad 3 FontEnd.

1. Creación de componentes y métodos asociados a los mismos.

Utilizamos y modificamos las tablas dentro de la BD **mascotas** en el BackEnd.



De igual manera la tabla **adopción**.



Creamos el Modelo adopcionMascotas

```
1  import Sequelize from "sequelize";
2  import { db } from "../database/conexion.js";
3
4  // Creamos el modelo adopcionMascotas
5  const adopcionMascotas = db.define("adopcion", {
6      id: {
7          type: Sequelize.INTEGER,
8          primaryKey: true,
9          autoIncrement: true,
10         allowNull: false,
11     },
12     nombre: {
13         type: Sequelize.STRING,
14         allowNull: true
15     },
16     telefono: {
17         type: Sequelize.INTEGER,
18         allowNull: true
19     },
20     id_mascota: {
21         type: Sequelize.INTEGER,
22         allowNull: true
23     },
24 },
25 {
26     timestamps: false,
27     createdAt: false,
28     updatedAt: false
29 }
30 );
31 // Exportamos el modelo adopcion
32 export { adopcionMascotas };
```

Dentro de **mascotasController** importamos el modelo

```
import { adopcionMascotas } from "../modelos/adopcionModelo.js";
```

Luego creamos los recursos para el proceso de adopción.

```
34 // Crear recurso para la adopción
35 const crearAdopcion = (req, res) => {
36   if (!req.body.nombre) {
37     res.status(400).json({
38       mensaje: "El nombre no puede estar vacío."
39     });
40     return;
41   }
42   const dataset = {
43     nombre: req.body.nombre,
44     telefono: req.body.telefono,
45     id_mascota: req.body.id_mascota,
46   };
47
48   adopcionMascotas.create(dataset)
49     .then((resultado) => {
50       res.status(200).json({
51         tipo: "success",
52         mensaje: "Registro creado correctamente"
53       });
54     })
55     .catch((err) => {
56       res.status(500).json({
57         tipo: "error",
58         mensaje: `Error al crear el registro ::: ${err}`
59       });
60     });
61 };
62
```

Exportamos las funciones del controlador faltante **crearAdopcion**

```
// Exportamos las diferentes funciones del controlador
export { crear, buscarId, buscar, actualizar, eliminar, crearAdopcion };
```

En el Proyecto FrontEnd **mascotas-fe** modificamos MascotasComponent.js para las solicitudes que más adelante se requieran.

```
//CUERPO COMPONENTE
const MascotasComponent = () =>
{
  const url = "http://localhost:8000/mascotas";
  const [mascotas, setMascotas] = useState([]);
  const [id, setId] = useState("");
  const [nombre, setNombre] = useState("");
  const [edad, setEdad] = useState("");
  const [raza, setRaza] = useState("");
  const [foto, setFoto] = useState("");
  const [detalle1, setDetalle1] = useState("");
  const [detalle2, setDetalle2] = useState("");
  const [operacion, setOperacion] = useState("");
  const [titulo, setTitulo] = useState("");
  const [pageNumber, setPageNumber] = useState(0);
  const mascotasPerPage = 3;
}
```

```
////////////////////////////////////
//ABRE EL MODAL DE MASCOTAS
const openModal =(opcion, id, nombre, edad, raza, foto, detalle1, detalle2)=>
{
  setId('');
  setNombre('');
  setEdad('');
  setRaza('');
  setFoto('');
  setDetalle1('');
  setDetalle2('');
  setOperacion(opcion);
  if(opcion === 1)
  {
    setTitulo("Registrar Mascota");
  }
  else if(opcion===2)
  {
    setTitulo("Editar Mascota");
    setId(id);
    setNombre(nombre);
    setEdad(edad);
    setRaza(raza);
    setFoto(foto);
    setDetalle1(detalle1);
    setDetalle2(detalle2);
  }
}
```

```
////////////////////////////////////  
//VALIDAR INFORMACION PARA CREAR Y ACTUALIZAR DATOS DE MASCOTA
```

```
const validar = ()=>  
{  
  let parametros;  
  let metodo;  
  if(nombre.trim()=== '')  
  {  
    console.log("Debe escribir un Nombre");  
    mostrarAlerta("Debe escribir un Nombre");  
  }  
  else if(edad.trim()=== '')  
  {  
    console.log("Debe escribir una Edad");  
    mostrarAlerta("Debe escribir una Edad");  
  }  
  else  
  {  
    if(operacion===1)  
    {  
      parametros=  
      {  
        urlExt: `${url}/crear`,  
        nombre: nombre.trim(),  
        edad: edad.trim(),  
        raza: raza.trim(),  
        foto: foto.trim(),  
        detalle1: detalle1.trim(),  
        detalle1: detalle1.trim()  
      };  
      metodo="POST";  
    }  
    else  
    {  
      parametros=  
      {  
        urlExt: `${url}/actualizar/${id}`,  
        nombre: nombre.trim(),
```

```

////////////////////////////////////
//ELIMINAR MASCOTA
const eliminarMascota=(id,nombre)=>
{
  const MySwal = withReactContent(Swal);
  MySwal.fire({
    title: `Estas seguro de eliminar la mascota ${nombre} ?`,
    icon: 'question',
    text: 'Se eliminará Definitivamente',
    showCancelButton: true,
    confirmButtonText: 'Si, eliminar',
    cancelButtonText: 'Cancelar'
  }).then((result)=>{
    if(result.isConfirmed){
      setId(id);
      enviarSolicitud("DELETE",{urlExt: `${url}/eliminar/${id}`,id:id})
    }
    else{
      mostrarAlerta("No se elimino la mascota","info");
    }
  })
}

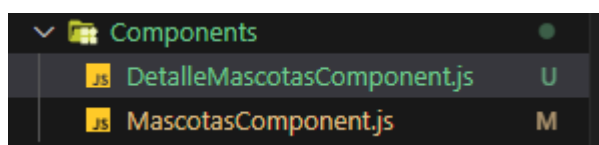
```

```

////////////////////////////////////
//CAMBIAR PAGINA
const handlePageChange = ({ selected }) => {
  setPageNumber(selected);
};
const pageCount = Math.ceil(mascotas.length / mascotasPerPage);
const offset = pageNumber * mascotasPerPage;

```

Dentro de la carpeta **Components** creamos el archivo **DetalleMascotasComponent.js** para poder visualizar la descripción o el detalle de cada mascota.



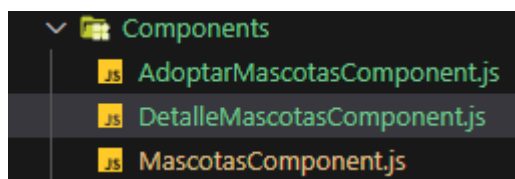
Creamos el cuerpo del componente con la conexión entre rutas, también la función para poder obtener la descripción de cada mascota por ID.

```
// CUERPO DEL COMPONENTE
const AdoptarMascotasComponent = () => {
  const { id } = useParams();
  const url = "http://localhost:8000/mascotas";
  const [mascotas, setMascotas] = useState([null]);

  useEffect(() => {
    getMascotaById(id);
  }, [id]);

  ////////////////////////////////////////
  //INFORMACION DE MASCOTAS
  const getMascotaById = async (id) => {
    try {
      const respuesta = await axios.get(`${url}/buscar/${id}`);
      console.log(respuesta.data);
      setMascotas(respuesta.data);
    } catch (error) {
      console.error("Error al obtener la mascota:", error);
    }
  };
};
```

Dentro de la carpeta **Components** creamos el archivo **AdoptarMascotasComponent.js** para poder realizar la solicitud de adopción.



```
// CUERPO DEL COMPONENTE
const AdoptarMascotasComponent = () => {

  const url = "http://localhost:8000/mascotas";
  const [mascotas, setMascotas] = useState([]);
  const [id, setId] = useState("");
  const [nombre, setNombre] = useState("");
  const [telefono, setTelefono] = useState("");
  const [titulo, setTitulo] = useState("");
```

```

////////////////////////////////////
//CAMBIAR PAGINA
const handlePageChange = ({ selected }) => {
  setPageNumber(selected);
};

//REALIZA LA OPERACION PARA OBTENER EL NUMERO DE PAGINAS
const pageCount = Math.ceil(mascotas.length / mascotasPerPage);
const offset = pageNumber * mascotasPerPage;

////////////////////////////////////
//MODAL PARA EL PROCESO DE ADOPTAR UNA MASCOTA
const openModal = (mascotaId) => {
  setTitulo("Datos Adopción Mascota");
  setId(mascotaId);
  setNombre('');
  setTelefono('');
};

```

```

////////////////////////////////////
//SOLICITUD DE ADOPCION AL SERVIDOR
const enviarSolicitud = async (metodo, parametros) => {
  try {
    const respuesta = await axios({ method: metodo, url: parametros.urlExt
    let tipo = respuesta.data.tipo;
    let mensaje = respuesta.data.mensaje;
    mostrarAlerta(mensaje, tipo);
    if (tipo === "success") {
      document.getElementById("btnCerrarModal").click();
      getMascotas();
    }
  } catch (error) {
    mostrarAlerta(`Error en la solicitud`, error);
  }
};

```


2. Uso de HTML 5 y JavaScript (Se debe desarrollar una estructura ordenada, con código legible y documentado).

Uso de HTML para crear la interfaz de usuario para las diferentes solicitudes

```
return (
  <div className="App">
    {/* Mostrar LOGO */}
    <div class="d-flex align-items-center justify-content-between">
      
        <input
          class="form-control me-2"
          type="search"
          placeholder="Buscar por nombre o raza"
          aria-label="Search"
          style={{ width: '100%' }}
          value={busqueda}
          onChange={(e) => setBusqueda(e.target.value)}
        />
        <button class="btn btn-success" type="button" onClick={getMa
          Buscar
        </button>
      </form>

      <Link to={`/${}`} className="btn btn-danger ms-2">
        <i className="fas fa-sign-out-alt"></i> Cerrar Sesión
      </Link>
    </div>
    <div className="container-fluid">
      {/* AÑADIR MASCOTA */}
      <div className="row mt-3">
        <div className="col-md-4 offset-md-4">
          <div className="d-grid mx-auto">
            <button
              onClick={() => openModal(1)}
              className="btn btn-success"
            />
          </div>
        </div>
      </div>
    </div>
  </div>
)
```

```

    { /* AÑADIR MASCOTA */ }
    <div className="row mt-3">
      <div className="col-md-4 offset-md-4">
        <div className="d-grid mx-auto">
          <button
            onClick={() => openModal(1)}
            className="btn btn-success"
            data-bs-toggle="modal"
            data-bs-target="#modalMascotas">
            <i className="fa-solid fa-circle-plus"></i>Añadir
          </button>
        </div>
      </div>
    </div>
  </div>
  { /* MOSTRAR MASCOTA */ }
  <div className="row mt-3">
    <div className="col-12 col-lg-8 offset-0 offset-lg-2">
      <div className="row">
        {mascotas.slice(offset, offset + mascotasPerPage).map((m
          <div key={mascota.id} className="col-12 col-md-6 col-1
            <div className="card" style={{ width: '18rem' }}>
              <img src={mascota.imagen} className="card-img-top">
              <div className="card-body">
                <h5 className="card-title">{mascota.nombre}</h5>
                <p className="card-text">
                  <strong>Edad:</strong> {mascota.edad}
                  <br/>
                  <strong>Raza:</strong> {mascota.raza}
                  <br/>
                  {mascota.descripcion1}
                </p>
                <button
                  onClick={() => openModal(2, mascota.id, mascot
                  className="btn btn-warning"
                  data-bs-toggle="modal"

```

```

    { /* CAMBIAR DE PAGINA */ }
    <div className="row mt-3">
      <div className="col-12 text-center">
        <ReactPaginate
          previousLabel={"Anterior"}
          nextLabel={"Siguiente"}
          breakLabel={"..."}
          pageCount={pageCount}
          onPageChange={handlePageChange}
          containerClassName={"custom-pagination"}
          pageClassName={"custom-pagination-item"}

```

Uso de HTML para el archivo **DetalleMascotaComponent.js**

Mostramos el logo de la página en la descripción de cada mascota.

```
////////////////////////////////////  
// INTERFAZ DEL COMPONENTE  
return (  
  <div className="App">  
    <br />  
    <div>  
        
    </div>  
  <br />  
)
```

Verificación de información de la mascota y botón regresar para volver al estado de lista de mascotas.

```
{mascotas && (  
  <div className="container">  
    <br />  
    <div>  
      <h2>{mascotas.nombre} ({mascotas.raza})</h2>  
      <p style={{ textAlign: 'justify' }}>{mascotas.descripcion2}</p>  
    </div>  
    <div className="d-flex justify-content-between mt-3">  
      <Link to={`/${}`} className="btn btn-dark ms-auto">  
        <i className="fas fa-reply"></i> Regresar  
      </Link>  
    </div>  
  <br />  
</div>  
)
```

Uso de HTML para el archivo **AdoptarMascotaComponent.js**

```
//////////////////////////////////////////
// INTERFAZ DEL COMPONENTE
return (
  <div className="App">
    <br/>
    <div class="d-flex align-items-center justify-content-between">
      
      <form class="d-flex" role="search" style={{ flex: '1' }}>
        <input
          class="form-control me-2"
          type="search"
          placeholder="Buscar por nombre o raza"
          aria-label="Search"
          style={{ width: '100%' }}
          value={busqueda}
          onChange={(e) => setBusqueda(e.target.value)}
        />
        <button class="btn btn-success" type="button" onClick={getMascotas}>
          Buscar
        </button>
      </form>
      <Link to="/login">
        
      </Link>
    </div>
  </div>
)
```

Contenedor principal donde se visualiza las mascotas ingresadas

```
<div className="container-fluid">
  <div className="row mt-3">
    <div className="col-12 col-lg-8 offset-0 offset-lg-2">
      <div className="row">
        {mascotas.slice(offset, offset + mascotasPerPage).map((mascota) => (
          <div key={mascota.id} className="col-12 col-md-6 col-lg-4 mb-4">
            <div className="card" style={{ width: '18rem' }}>
              <img
                src={mascota.imagen}
                className="card-img-top"
                alt=""
                style={{ objectFit: 'cover', height: '220px' }}
              />
              <div className="card-body">
                <h5 className="card-title">{mascota.nombre}</h5>
                <p className="card-text">
                  <strong>Edad:</strong> {mascota.edad}
                  <br />
                  <strong>Raza:</strong> {mascota.raza}
                  <br />
                  {mascota.descripcion1}
                </p>
              </div>
            </div>
          </div>
        )}
      </div>
    </div>
  </div>
</div>
```

Información de la mascota, enlace para la descripción de la misma en el botón **Detalles** y botón para el proceso de **Adoptar**.

```
<div className="card-body">
  <h5 className="card-title">{mascota.nombre}</h5>
  <p className="card-text">
    <strong>Edad:</strong> {mascota.edad}
    <br />
    <strong>Raza:</strong> {mascota.raza}
    <br />
    {mascota.descripcion1}
  </p>
  <Link to={`/detalles/${mascota.id}`} className="btn btn-primary">
    <i className="fas fa-info-circle"></i> Detalles
  </Link>
  { ' ' }
  <button
    onClick={() => openModal(mascota.id)}
    className="btn btn-primary"
    data-bs-toggle="modal"
    data-bs-target="#modalAdopcion"
  >
    <i className="fas fa-paw"></i> Adoptar
```

Modal para adoptar una mascota, con un requerimiento de datos.

```
<div id="modalAdopcion" className="modal fade" aria-hidden="true">
  <div className="modal-dialog">
    <div className="modal-content">
      <div className="modal-header">
        <label className="h5">Por favor ingrese sus datos</label>
      </div>
      <div className="modal-body">
        <div className="input-group mb-3">
          <span className="input-group-text">
            <i className="fa-solid fa-dog"></i>
          </span>
          <input
            type="text"
            id="id_mascota"
            className="form-control"
            value={id}
            onChange={(e) => setId(e.target.value)}
            readOnly
          ></input>
        </div>
      </div>
    </div>
  </div>
```

```

<div className="input-group mb-3">
  <span className="input-group-text">
    <i className="fa-solid fa-user"></i>
  </span>
  <input
    type="text"
    id="nombre"
    className="form-control"
    placeholder="Nombre"
    value={nombre}
    onChange={(e) => setNombre(e.target.value)}
  ></input>
</div>
<div className="input-group mb-3">
  <span className="input-group-text">
    <i className="fa-solid fa-phone"></i>
  </span>
  <input
    type="number"
    id="telefono"
    className="form-control"
    placeholder="Telefono"
    value={telefono}
    onChange={(e) => setTelefono(e.target.value)}
  ></input>

```

Botón para enviar datos del solicitante para Adopción

```

<div className="d-grid col-6 mx-auto">
  <button onClick={() => validar()} className="btn btn-success">
    <i className="fa-solid fa-floppy-disk"></i> Enviar
  </button>
</div>
</div>
<div className="modal-footer">
  <button
    id="btnCerrarModal"
    type="button"
    className="btn btn-secondary"
    data-bs-dismiss="modal"
  >
    Cerrar

```

Actualizamos el archivo **App.js** importando los componentes y definiendo las rutas de cada uno.

```
//IMPORT
import {BrowserRouter, Routes, Route} from 'react-router-dom';
import MascotasComponent from './Components/MascotasComponent.js';
import AdoptarMascotasComponent from './Components/AdoptarMascotasComponent.js';
import DetalleMascotasComponent from './Components/DetalleMascotasComponent.js';

import './App.css';

function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<AdoptarMascotasComponent></AdoptarMascotasComponent>}></Route>

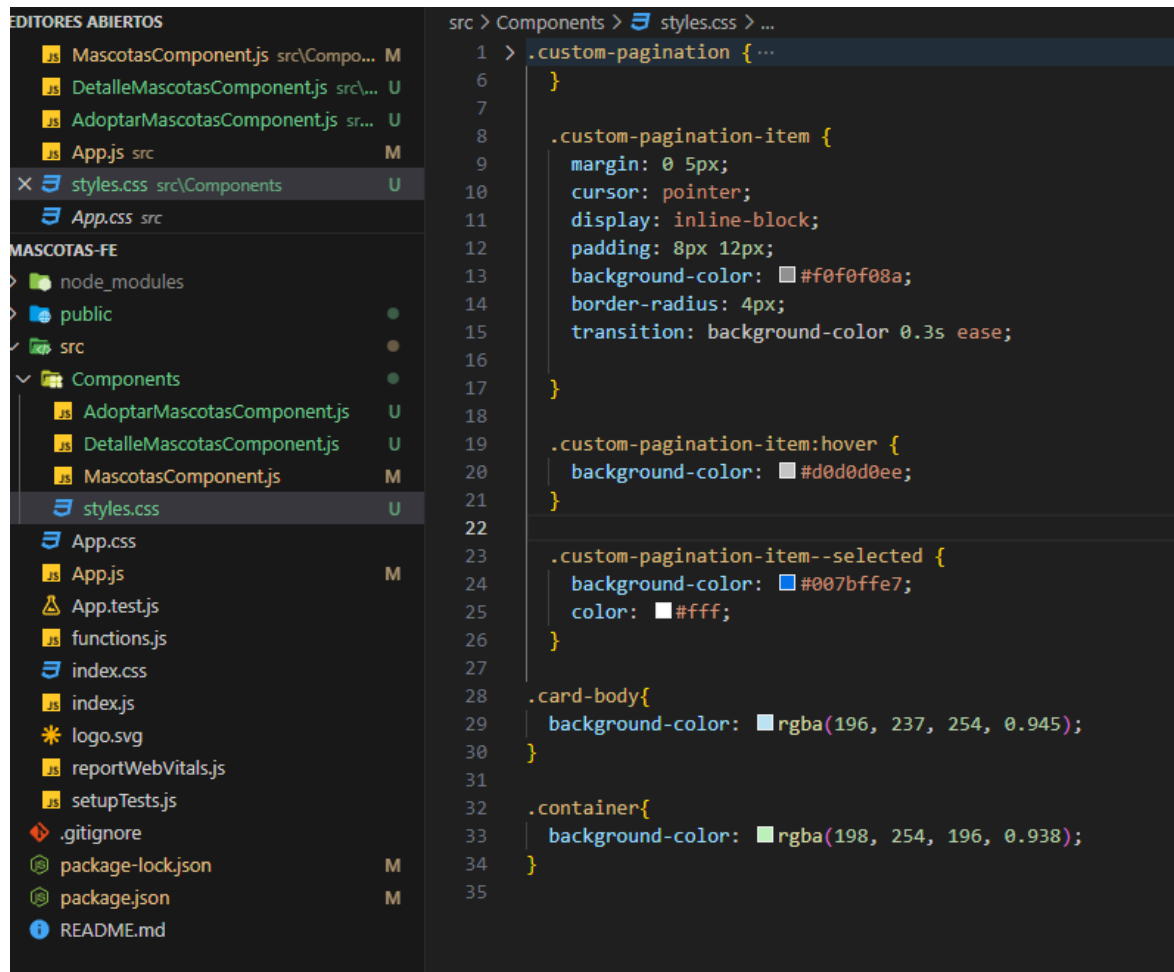
        <Route path="/crear" element={<MascotasComponent></MascotasComponent>}></Route>

        <Route path="/detalles/:id" element={<DetalleMascotasComponent></DetalleMascotasComponent>}></Route>
      </Routes>
    </BrowserRouter>
  );
}

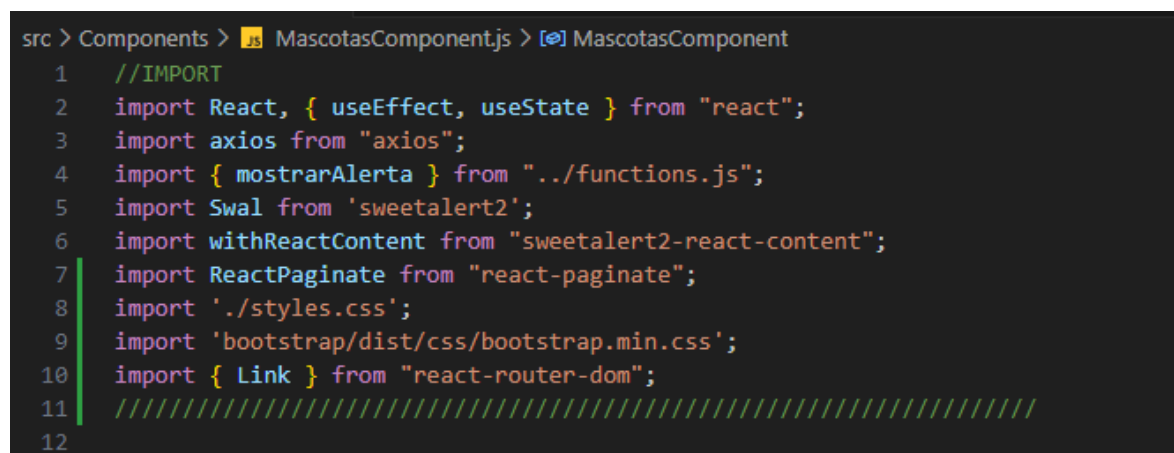
//EXPORT
export default App;
```

3. Estilos CSS (Uso de Bootstrap), se debe generar una interface ordenada estructurada y agradable para el usuario final.

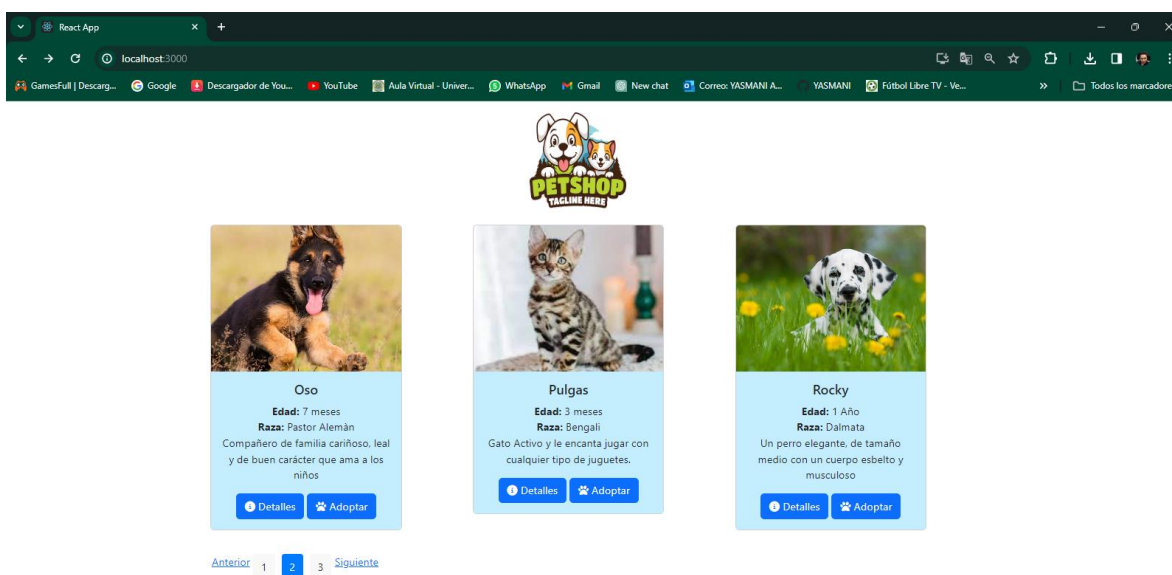
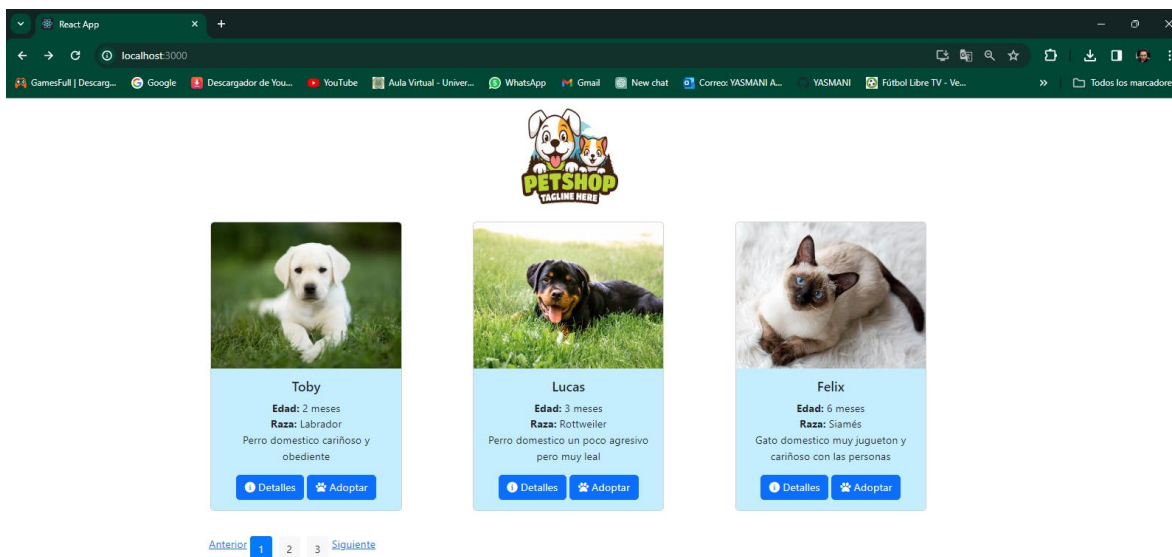
Uso de estilos CSS para cada componente

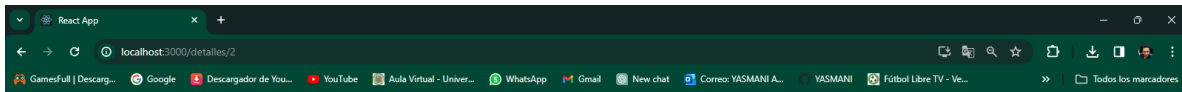


Imports para cada componente en el uso de **Styles.css** y **Bootstrap**



Algunos pantallasos del resultado final, añadiendo varios datos desde la interfaz del navegador y comprobandolos en la base de datos **mascotas**.





Toby (Labrador)

El labrador es un perro grande, de constitución fuerte y buena estructura ósea y muscular. Tiene la cabeza ancha y una mirada dulce e inteligente. Su cola, parecida a la de las nutrias, es única. El pelaje es corto y denso y puede ser de color negro liso, amarillo o marrón

[Regresar](#)

1	2	Toby	2 meses	Labrador	https://t1.uc.ltmcdn.com/es/pos	Perro domestico cariñoso y obedien	El labrador es un perro
2	3	Lucas	3 meses	Rottweiler	https://media.istockphoto.com/i	Perro domestico un poco agresivo p	El rottweiler es un per
3	4	Felix	6 meses	Siamés	https://clinicaveterinarianum.es/wp	Gato domestico muy jugueton y car	Les encanta jugar y a
4	5	Oso	7 meses	Pastor Alemán	https://www.bunko.pet/_export/	Compañero de familia cariñoso, leal	Son perros ágiles y bi
5	6	Pulgas	3 meses	Bengali	data:image/jpeg;base64,/9j/4AAQSk	Gato Activo y le encanta jugar con c	El aspecto salvaje del
6	7	Rocky	1 Año	Dalmata	https://t2.ea.ltmcdn.com/es/post	Un perro elegante, de tamaño medi	Esta raza destaca por
7	8	Paco	2 Años	Persa	https://images.hola.com/imagen	Gato que procede de dos variantes c	El persa suele ser un
8	9	Zeus	8 meses	Pitbull	https://www.santevet.es/uploads	Perro seleccionado por su fuerza, va	Aunque en su origen

Cuando hacemos Click en Adoptar

Toby
Edad: 2 meses
Raza: Labrador
Perro domestico cariñoso y obediente

[Detalles](#) [Adoptar](#)

Lucas
Edad: 3 meses
Raza: Rottweiler
Perro domestico un poco agresivo pero muy leal

[Detalles](#) [Adoptar](#)

Felix
Edad: 6 meses
Raza: Siamés
Gato domestico muy jugueton y cariñoso con las personas

[Detalles](#) [Adoptar](#)

Por favor ingrese sus datos

3

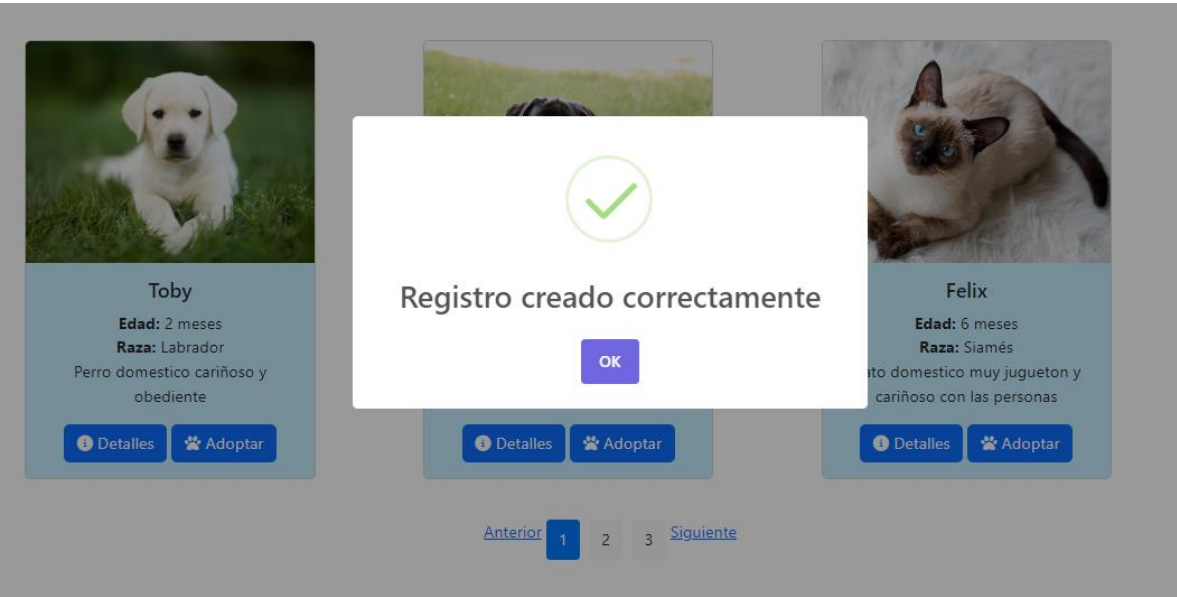
Yasmani

3152568418

[Enviar](#)

[Cerrar](#)

[Anterior](#) [1](#) [2](#) [3](#) [Siguiente](#)



	123 id	ABC nombre	123 telefono	123 id_mascota
1	1	Alejandro	2.147.483.647	2
2	2	Yasmani	3.152.568.418	3