

Taller Unidad 3 Frontend

Taller Unidad 3 Frontend	1
Creación del crud de Mascotas	1
Vista Lista de Mascotas	3
Adopción de Mascotas	5
Vista de Administración de Solicitudes	8
Header	14
Resultados	17

Creación del crud de Mascotas

1. Crear las rutas en /src/App.js, esto se hace mediante el uso de la biblioteca react-router-dom, obtenemos algo como los siguiente:

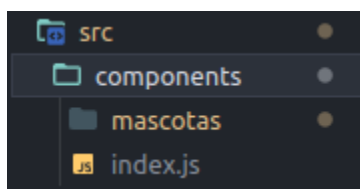
```
import './App.css';
import {BrowserRouter,Routes,Route} from 'react-router-dom';
import { MascotasComponent } from './components';

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

export default App;
```

Podremos agregar más rutas a posteriori.

2. Posteriormente crearemos la carpeta components y la subcarpeta mascotas.



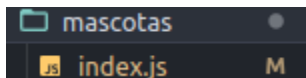
3. dentro del archivo src/components.js agregamos lo siguiente:

```
import MascotasComponent from "../mascotas";
```

```
export { MascotasComponent };
```

Este archivo nos permite exportar cada componente de la carpeta components y de esa manera podremos organizar mejor cada componente en una subcarpeta, pero podrá ser importado desde components directamente.

- Ahora, dentro la la carpeta src/components/mascotas crearemos el archivo index.js



- Dentro del archivo src/components/mascotas/index.js colocaremos toda la lógica necesaria para poder crear el crud de mascotas.

Este proceso ya se hizo en clase y consta de usar la biblioteca de axios para establecer conexión con el backed, guardamos los datos obtenidos de la consulta en una variable y la paramos para poder mostrar los datos, con un ciclo, en este caso array.map, recorremos el array de datos y los vamos colocando en una tabla, también colocamos 2 botones que permiten enviar las solicitudes editar, de tipo put y eliminar de tipo delete, además de un botón que permite enviar la solicitud de tipo post agregar elemento. Téngase en cuenta que además se usan conceptos de HTML, css Y bootstrap para permitir hacer uso de alertas y modales.

Como elemento adicional, no visto en clase tenemos la configuración que permite mostrar datos diferentes de acuerdo al valor de un campo, de la siguiente forma:

```
<td>
  {mascota.tipo_mascota === "P"?
    (<p>Perro</p>) :
    (<p>Gato</p>)
  }
</td>
<td>
  {mascota.estado === 1?
    (<p>Adoptado</p>) :
    (<p>Lista para Adoptar</p>)
  }</td>
<td>
```

En este caso se puede ver que si el tipo de mascota es P entonces se colocará perro en una etiqueta de párrafo, de lo contrario será gato, caso similar con la propiedad estado, donde

1 significa que la mascota fue adoptada y otro caso significa que está disponible para ser adoptada.

Vista Lista de Mascotas

La vista anterior permite listar las mascotas, editarlas, eliminarlas y agregar nuevas mascotas, sin embargo es una vista mas para un administrador del sitio, ahora se pretende desarrollar una vista para la vista de un usuario que quiere adoptar una mascota.

El primer paso es crear el componente mascota:

1. Crear la carpeta /src/components/mascota y dentro el index.js



El contenido inicial del archivo index.js en este contexto es:

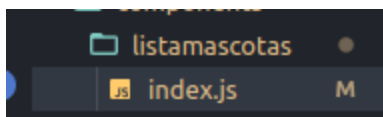
```
1  import React from "react";
2
3  const MascotaComponent = ({pk, nombre, edad, tipo_mascota}) => {
4    return (
5      <div className="card">
6        <div className="card-body">
7          <h5 className="card-title text-center">{nombre}</h5>
8          <p className="card-text text-center">{tipo_mascota}</p>
9          <hr />
10         <p className="card-text text-center">Edad: {edad} años.</p>
11         <button className="btn btn-success w-100">
12           Adoptar {pk}
13         </button>
14       </div>
15     </div>
16   );
17 }
18
19 export default MascotaComponent;
```

2. El siguiente paso es añadir al archivo src/components/index.js, el nuevo componente:

```
import MascotasComponent from "../mascotas";
import MascotaComponent from "../mascota";

export { MascotasComponent, MascotaComponent };
```

3. Crear el componente lista mascotas



Contenido del index en este contexto:

```
import React, { useEffect, useState } from "react";
import MascotaComponent from "../mascota";
import axios from "axios";

const ListaMascotasComponent = () => {

  const url = 'http://localhost:9000/mascotas';

  const [mascotas, setMascotas] = useState([]);

  useEffect(() => {
    getMascotas();
  }, []);

  const getMascotas = async () => {
    const res = await axios.get(`${url}`);
    setMascotas(res.data);
  }

  return (
    <div className="container mt-5">
      <div className="row row-cols-1 row-cols-sm-1
row-cols-md-3" >
        {
          mascotas.map((mascota, index)=>{
            const data = {
              pk:mascota.pk,
              nombre: mascota.nombre,
              edad: mascota.edad,
              tipo_mascota: mascota.tipo_mascota
            }
            return (
              <div className="col mb-5" key={index +1}>
                <MascotaComponent {...data} />
              </div>
            )
          })
        }
      </div>
    </div>
  )
}
```

```

    );
  })
}
</div>

</div>
);
}

export default ListaMascotasComponent;

```

Adopción de Mascotas

Ahora desarrollaremos la lógica para que el Botón de adoptar genere una solicitud de adopción:

```

import React, { useState } from "react";
import axios from "axios";
import { mostrarAlerta } from "../functions";

const MascotaComponent = ({pk, nombre, edad, tipo_mascota}) => {

  const url = "http://localhost:9000/solicitudes/crear/";
  const [adoptante, setAdoptante] = useState('');

  const validar = () => {

    if(adoptante === ''){
      mostrarAlerta("Debe escribir su nombre");
    }else{
      console.log(adoptante);
      const hoy = new Date();
      const fecha_inicio = hoy.toISOString().split('T')[0];
      const parametros = {
        mascotaPK:pk,
        adoptante:adoptante,
        estado:"A",
        fecha_inicio:fecha_inicio
      }
      enviarSolicitud('POST', url, parametros);
    }
  }
}

```

```

    }

    const enviarSolicitud = async (metodo, url, parametros) => {
      await axios({method: metodo, url: url, data: parametros}).then((res) => {
        {
          mostrarAlerta(res.data.mensaje, 'success');
        }}.catch((error) => {
          console.log(parametros);
          mostrarAlerta(`Error en la solicitud ${error}`, 'error');
        });
      }

      return (
        <div className="card">
          <div className="card-body">
            <h5 className="card-title text-center">{nombre}</h5>
            <p className="card-text text-center">{
              tipo_mascota === "P"?("Perro"):( "Gato")
            }</p>
            <hr />
            <p className="card-text text-center">Edad: {edad} años.</p>
            <button
              className="btn btn-success w-100"
              data-bs-toggle = "modal"
              data-bs-target = `{`#mascotasModal${pk}`}`
            >
              Adoptar
            </button>
          </div>

          <div className="modal fade"
            id={`mascotasModal${pk}`}
            tabIndex="-1"
            aria-labelledby="modal"
            aria-hidden="true"
          >
            <div className="modal-dialog">
              <div className="modal-content">
                <div className="modal-header">

```

```

        <h1 className="modal-title fs-5"
id={`titulomascotasModal${pk}`}>Proceso de Adopcion de {nombre}</h1>
        <button type="button" className="btn-close"
data-bs-dismiss="modal" aria-label="Close"></button>
    </div>
    <div className="modal-body">
        <input
            type="text"
            id={`mascota${pk}adopcion`}
            className="form-control"
            placeholder="Ingrese su Nombre"
            value={adoptante}
            onChange={ (e) =>
setAdoptante(e.target.value) }
            >
        </input>
    </div>
    <div className="modal-footer">
        <button
            type="button"
            className="btn btn-secondary"
            data-bs-dismiss="modal"
        >Cancelar</button>
        <button
            type="button"
            className="btn btn-success"
            onClick={ (e) => validar() }
        >Adoptar</button>
    </div>
</div>
</div>
</div>
</div>
);
}

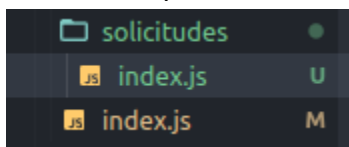
export default MascotaComponent;

```

Vista de Administración de Solicitudes

Ahora se hace la vista para la administración de las solicitudes:

1. crear el componente:



2. Exportar desde componentes:

```
1 import MascotasComponent from "../mascotas";
2 import MascotaComponent from "../mascota";
3 import ListaMascotasComponent from "../listamascotas"
4 import HeaderComponent from "../header";
5 import SolicitudesComponent from "../solicitudes";
6
7 export { MascotasComponent, MascotaComponent, ListaMascotasComponent, HeaderComponent, SolicitudesComponent};
```

3. Añadir la ruta

```
import './App.css';
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import { HeaderComponent, ListaMascotasComponent, MascotasComponent, SolicitudesComponent } from './components'

function App() {
  return (
    <BrowserRouter>
      <HeaderComponent />
      <Routes>
        <Route path="/" element={ <MascotasComponent /> } />
        <Route path="/mascotas" element={ <ListaMascotasComponent /> } />
        <Route path="/solicitudes" element={ <SolicitudesComponent /> } />
      </Routes>
    </BrowserRouter>
  );
}

export default App;
```

4. Código para administración de solicitudes: src/components/solicitudes/index.js

```
import React, { useEffect, useState } from "react";
import axios from "axios";
import { mostrarAlerta } from "../../functions";

const SolicitudesComponent = () => {

  const url = 'http://localhost:9000/solicitudes';
  const urlM = 'http://localhost:9000/mascotas';
  const [solicitudes, setSolicitudes] = useState([]);
  const [mascotas, setMascotas] = useState([]);
```



```

useEffect(() => {
    getSolicitudes();
    getMascotas();
}, []);

const getSolicitudes = async () => {
    const res = await axios.get(`${url}`);
    setSolicitudes(res.data);
}

const getMascotas = async () => {
    const res = await axios.get(`${urlM}`);
    setMascotas(res.data);
}

const getMascota = (pk) => {
    const mascota = mascotas.find((m) => m.pk === pk);
    if(mascota !== null){
        return mascota;
    }

    return null;
}

const limpiarSolicitudes = (mascotaPK) => {
    solicitudes.map(async (sol) => {
        if(sol.mascotaPK === mascotaPK){
            const urlExtS = `${url}/actualizar/${sol.pk}`;
            const hoy = new Date();
            const fecha_fin = hoy.toISOString().split('T')[0];
            const params = {
                estado: "R",
                fecha_fin: fecha_fin
            }

            //rechazar todas las demás solicitudes
            await axios({method: 'PUT', url: urlExtS,
data:params});
        }
    });
};

```

```

}

const acpetarSolicitud = async (mascotaPK, solicitudPK) =>{
  if(mascotaPK !== '' && solicitudPK !== ''){
    const urlExt = `${urlM}/actualizar/${mascotaPK}`;
    const parametros = {
      estado:1
    }

    //cambiar el estado de la mascota
    await axios({method:'PUT', url:urlExt,
data:parametros}).then(async (res)=>{
      const urlExtS = `${url}/actualizar/${solicitudPK}`;
      const hoy = new Date();
      const fecha_fin = hoy.toISOString().split('T')[0];
      const params = {
        estado:"A",
        fecha_fin:fecha_fin
      }

      //primero se rechaza todas
      limpiarSolicitudes(mascotaPK);

      //cambiar el estado de la solicitud para aceptar solo
esa
      await axios({method:'PUT', url:urlExtS,
data:params}).then((res)=>{
        mostrarAlerta(`Adopcion Hecha con exito`,
'success');

        getSolicitudes();
        getMascotas();
      }).catch((error)=> {
        mostrarAlerta(`Error en la solicitud ${error}`,
'error');
      });
    }).catch((error)=> {
      mostrarAlerta(`Error en la solicitud ${error}`,
'error');
    });
  }
}

```

```

    }else{
        mostrarAlerta("No se puede encontrar la mascota o al
adoptante", "error");
    }
}

const rechazarSolicitud = async (pk) => {
    const urlExtS = `${url}/actualizar/${pk}`;
    const hoy = new Date();
    const fecha_fin = hoy.toISOString().split('T')[0];
    const params = {
        estado:"R",
        fecha_fin:fecha_fin
    }

    //actualizar los datos de la solicitud
    await axios({method:'PUT', url:urlExtS,
data:params}).then((res)=>{
        mostrarAlerta(`Solicitud Rechazada Exitosamente`,
'success');

        getSolicitudes();
    }).catch((error)=> {
        mostrarAlerta(`Error en la solicitud ${error}`, 'error');
    });
}

const eliminarSolicitud = async(pk) => {
    const urlExtS = `${url}/eliminar/${pk}`;

    //actualizar los datos de la solicitud
    await axios({method:'DELETE', url:urlExtS}).then((res)=>{
        mostrarAlerta(`Solicitud Eliminada Exitosamente`,
'success');

        getSolicitudes();
    }).catch((error)=> {
        mostrarAlerta(`Error en la solicitud ${error}`, 'error');
    });
}

```

```

return (
  <div className="container">
    <table className="table table-hover">
      <thead>
        <tr>
          <th >#</th>
          <th >Adoptante</th>
          <th >Mascota</th>
          <th >Fecha de Solicitud</th>
          <th colSpan={3}> Acciones </th>
        </tr>
      </thead>
      <tbody className="table-group-divider">
        {
          solicitudes.map((solicitud, index) => (
            <tr key={solicitud.pk}>
              <td>{index+1}</td>
              <td>{solicitud.adoptante}</td>
              <td>{
                getMascota(solicitud.mascotaPK) !== null?
                getMascota(solicitud.mascotaPK).nombre:
                "No Encontrada"
              }</td>
              <td>{solicitud.fecha_inicio}</td>
              {
                solicitud.estado==="P"? (
                  <td>
                    <button
                      className="btn btn-success w-100"
                      onClick={ (e)
=> acpetarSolicitud(solicitud.mascotaPK, solicitud.pk) }
                    >
                      <i
                        className="fa-solid fa-check"></i>
                      </button>

```

```

        </td>
    ): ""
    }
    {
        solicitud.estado=="P"? (
            <td>
                <button
                    className="btn btn-danger w-100"
                    onClick={ (e)
=> rechazarSolicitud(solicitud.pk) }
                    >
                        <i
                            className="fa-solid fa-xmark"></i>
                        </button>
                    </td>
                ): ""
            }
            {
                solicitud.estado=="A"? (
                    <td colspan={2}>
                        <div
                            className="alert alert-success text-center" role="alert">
                                Mascota
                                Adoptada en {solicitud.fecha_fin}
                                </div>
                            </td>
                        ): ""
                    }
                    {
                        solicitud.estado=="R"? (
                            <td colspan={2}>
                                <div
                                    className="alert alert-danger text-center" role="alert">
                                        Esta
                                        Solicitud fue Rechazada en {solicitud.fecha_fin}
                                    </div>
                                </td>
                            ): ""
                        }
                    }
                }
            }
        }
    }

```

```

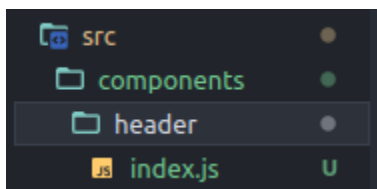
                                <td>
                                    <button
                                        className="btn
btn-danger w-100"
                                        onClick={ (e) =>
eliminarSolicitud(solicitud.pk) }
                                    >
                                        <i
className="fa-solid fa-trash"></i>
                                    </button>
                                </td>
                            </tr>
                        )
                    )
                }
            </tbody>
        </table>
    </div>
);
}

export default SolicitudesComponent;

```

Header

1. Creamos el componente



2. Adjuntar el componente en el src/components/index.js

```

import MascotasComponent from "../mascotas";
import MascotaComponent from "../mascota";
import ListaMascotasComponent from "../listamascotas";
import HeaderComponent from "../header";

export { MascotasComponent, MascotaComponent,
ListaMascotasComponent, HeaderComponent };

```

3. Código a renderizar:

```

import React from "react";
import "../index.css";

const HeaderComponent = () => {
  return (
    <div className="sticky-top">
      <nav className="navbar navbar-expand-lg">
        <div className="container-fluid">
          <button
            className="nav-link navbar-brand"
          >Mascotas</button>
          <button
            className="navbar-toggler"
            type="button" data-bs-toggle="collapse"
            data-bs-target="#navbarTogglerDemo02"
            aria-controls="navbarTogglerDemo02"
            aria-expanded="false"
            aria-label="Toggle navigation"
          >
            <span className="navbar-toggler-icon"></span>
          </button>
          <div
            className="collapse navbar-collapse"
            id="navbarTogglerDemo02"
          >
            <ul className="navbar-nav me-auto mb-2
mb-lg-0">
              <li className="nav-item">
                <button
                  className="nav-link"
                  disabled={window.location.pathname === "/"}
                  onClick={(e) => {
                    window.location =
"http://localhost:3000/";
                  }}
                >Administrar Mascotas</button>
              </li>
              <li className="nav-item">

```

```

        <button
          className="nav-link"
          disabled={window.location.pathname
=== "/mascotas"}

          onClick={ (e) => {
            window.location =
"http://localhost:3000/mascotas";
          }}
        >Adoptar Mascotas</button>
      </li>
      <li className="nav-item">
        <button
          className="nav-link"

          disabled={window.location.pathname === "/solicitudes"}
          onClick={ (e) => {
            window.location =
"http://localhost:3000/solicitudes";
          }}
        >Administrar Solicitudes</button>
      </li>
    </ul>
  </div>
</div>
</nav>
</div>
);
}

export default HeaderComponent;

```

EL archivo index.css contiene algunos estilos básicos.

4. Finalmente integramos el elemento de navegación al app.js debido a que es común para todas las vistas:

app.js:

```
import './App.css';
```



```
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import { HeaderComponent, ListaMascotasComponent, MascotasComponent
} from './components';

function App() {
  return (
    <BrowserRouter>
      <HeaderComponent />
      <Routes>
        <Route path="/" element={ <MascotasComponent /> } />
        <Route path="/mascotas" element={<ListaMascotasComponent />}
      />
      </Routes>
    </BrowserRouter>
  );
}

export default App;
```

Resultados

Se obtiene la vista de administración de mascotas de la siguiente forma:

Mascotas Administrar Mascotas Adoptar Mascotas Administrar Solicitudes					
● Añadir					
#	Nombre	Edad	Tipo de Mascota	Estado	Acciones
1	Avejon	3	Perro	Adoptado	 
2	Lucas	2	Perro	Adoptado	 
3	Pelusa	2	Gato	Lista para Adoptar	 
4	Mirringo	3	Gato	Adoptado	 
5	Manchas	5	Perro	Lista para Adoptar	 
6	Coco	1	Perro	Lista para Adoptar	 
7	Agi	8	Perro	Lista para Adoptar	 

Aquí el usuario podrá registrar, editar y eliminar mascotas.

Header:

El header le permite al usuario navegar por el sitio, el link del sitio donde se encuentre en el momento se deshabilitará.

Vista Móvil del header:



Vista móvil de Administrar mascotas:



Se cuenta con un scroll en la parte inferior que permite desplazarse horizontalmente en la tabla.

Vista adoptar Mascotas:

Mascotas
Administrar Mascotas
Adoptar Mascotas
Administrar Solicitudes

Avejon
Perro

Edad: 3 años.

Esta mascota ya fue Adoptada

Lucas
Perro

Edad: 2 años.

Esta mascota ya fue Adoptada

Pelusa
Gato

Edad: 2 años.

Adoptar

Mirringo
Gato

Edad: 3 años.

Esta mascota ya fue Adoptada

Manchas
Perro

Edad: 5 años.

Adoptar

Coco
Perro

Edad: 1 años.

Adoptar

Las mascotas cuya solicitud de adopción ya fue aceptada, ya no tendrán botón adoptar y aparecerá el letrero donde dice que ya fueron adoptadas.

Vista Administrar solicitudes:

Mascotas Administrar Mascotas Adoptar Mascotas Administrar Solicitudes				
#	Adoptante	Mascota	Fecha de Solicitud	Acciones
1	Mabel Costain	Lucas	2023-12-25	Mascota Adoptada en 2023-12-26 
2	Lucia Ortiz	Toby	2023-12-26	Mascota Adoptada en 2023-12-26 
3	Juan Gallardo	Agi	2023-12-26	  
4	Segundo Lasso	Agi	2023-12-26	  
5	Maria Paladines	Coco	2023-12-26	  
6	Juan Gallardo	Pelusa	2023-12-26	Esta Solicitud fue Rechazada en 2023-12-26 

En esta vista se administran las solicitudes, al aceptar una solicitud, todas las demás que estaban en espera por la misma mascota son rechazadas, si se elimina una solicitud solamente se saca de la tabla, al rechazarla se muestra como rechazada.