CRUD LARAVEL REACT
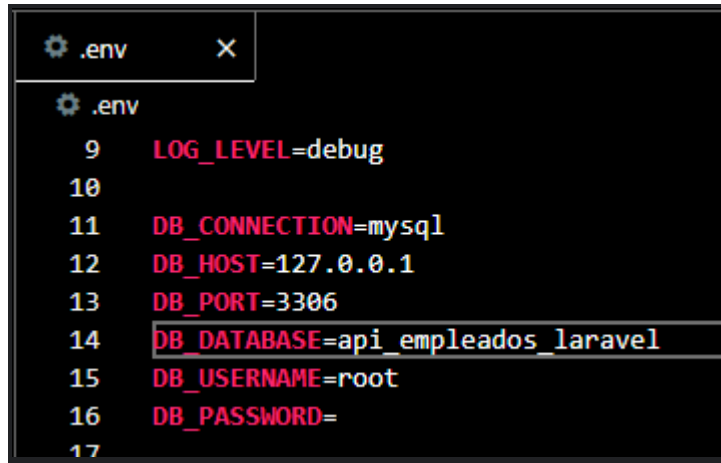
```
composer create-project laravel/laravel proyecto_name "8.*"
```

CREAR BASE DE DATOS Y AGREGAR AL   .env

```
⚙ .env          ✕
  ⚙ .env
   9    LOG_LEVEL=debug
  10
  11    DB_CONNECTION=mysql
  12    DB_HOST=127.0.0.1
  13    DB_PORT=3306
  14    DB_DATABASE=api_empleados_laravel
  15    DB_USERNAME=root
  16    DB_PASSWORD=
  17
```

```
php artisan make:model Empleado -m
```

```
php artisan make:controller Api/EmpleadoController --api
```

```
php artisan make:seeder EmpleadoSeeder
```

EMPLEADO CONTROLLER

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateEmpleadosTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('empleados', function (Blueprint $table) {
            $table->id();
            $table->id();
            $table->string('nombre');
            $table->bigInteger('celular');
            $table->string('correo');
            $table->timestamps();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
```

```php
    public function down()
    {
        Schema::dropIfExists('empleados');
    }
}
```

MODELO EMPLEADO

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Empleado extends Model
{
    use HasFactory;
    protected $fillable = ['nombre', 'celular', 'correo'];
}
```

php artisan:migrate

CONTROLADOR EMPLEADO

```php
<?php

namespace App\Http\Controllers\Api;

use Illuminate\Http\Request;
use App\Models\Empleado;
use App\Http\Controllers\Controller;

class EmpleadoController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $empleados = Empleado::all();
        return $empleados;
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $empleado = new Empleado();
        $empleado->nombre = $request->nombre;
        $empleado->celular = $request->celular;
        $empleado->correo = $request->correo;

        $empleado->save();
    }

    /**
     * Display the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
```

```php
    public function show($id)
    {
        $empleado = Empleado::find($id);
        return $empleado;
    }


    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        $empleado = Empleado::findOrFail($request->id);
        $empleado->nombre = $request->nombre;
        $empleado->celular = $request->celular;
        $empleado->correo = $request->correo;

        $empleado->save();
        return $empleado;
    }


    /**
     * Remove the specified resource from storage.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $empleado = Empleado::destroy($id);
        return $empleado;
    }
}
```

EMPLEADO SEEDER

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use App\Models\Empleado;

class EmpleadoSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $empleado1 = new Empleado();
        $empleado1->nombre = "FELIPE";
        $empleado1->correo = "felipe@mail.com";
        $empleado1->celular = 54321;
        $empleado1->save();

        $empleado2 = new Empleado();
        $empleado2->nombre = "PEDRO";
        $empleado2->correo = "pedro@mail.com";
        $empleado2->celular = 54321;
        $empleado2->save();

        $empleado3 = new Empleado();
        $empleado3->nombre = "JUAN";
        $empleado3->correo = "juan@mail.com";
        $empleado3->celular = 54321;
        $empleado3->save();
    }
}
```

DATABASE SEEDER

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     *
     * @return void
     */
    public function run()
    {
        $this->call(EmpleadoSeeder::class);
    }
}
```
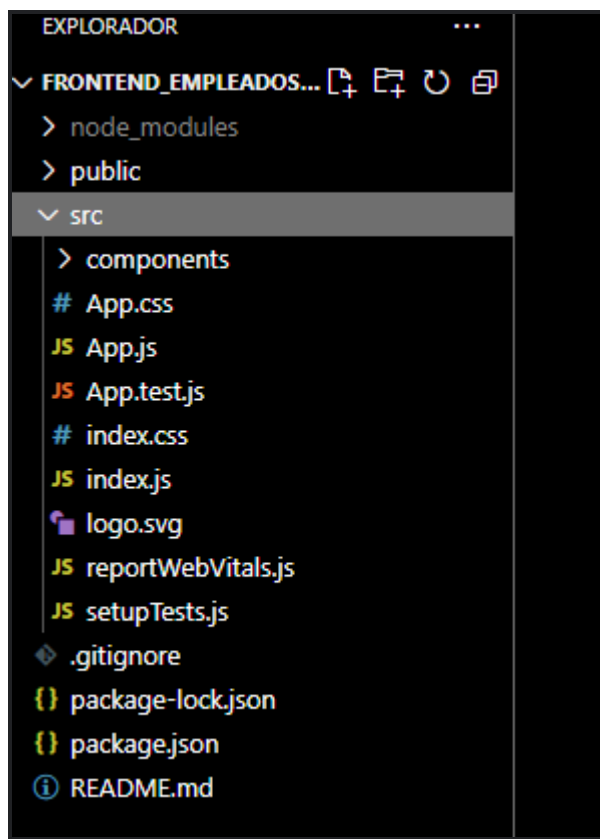
php artisan db:seed

php artisan serve

FRONT END CON REACT

npm install axios bootstrap react-router-dom@6

npm install sweetalert2 --save

index.js

```js
JS index.js    ✕

src > JS index.js > ...
  1    import React from 'react';
  2    import ReactDOM from 'react-dom/client';
  3    import './index.css';
  4    import App from './App';
  5    import reportWebVitals from './reportWebVitals';
  6    import 'bootstrap/dist/css/bootstrap.min.css';
  7
  8    const root = ReactDOM.createRoot(document.getElementById('root'));
  9    root.render(
 10      <React.StrictMode>
 11        <App />
 12      </React.StrictMode>
 13    );
 14
 15    // If you want to start measuring performance in your app, pass a function
 16    // to log results (for example: reportWebVitals(console.log))
 17    // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
 18    reportWebVitals();
 19
```

COMPONENTE MOSTRAR

```jsx
import React, {useEffect, useState} from 'react'
import Swal from 'sweetalert2';
import axios from 'axios'
import {Link} from 'react-router-dom'

const endpoint = 'http://localhost:8000/api'
const MostrarEmpleados = () => {
    const [ empleados, setEmpleados ] = useState([])
        useEffect( ()=> {
            getAllEmpleados()
        }, [])

    const getAllEmpleados = async () => {
        const response = await axios.get(`${endpoint}/empleados`)
        setEmpleados(response.data)
    }

  /*
  const deleteEmpleado = async (id) => {
        if(window.confirm("¿¿¿ Eliminar este registro ???")){
        await axios.delete(`${endpoint}/empleado/${id}`)
        getAllEmpleados()
        }
    }
  */
    const deleteEmpleado = async (id) => {
        const isConfirm = await Swal.fire({
            title: '¿ Estas seguro de borrar esto ?',
            text: "¡ Esta acción es irreversible !",
            icon: 'warning',
            showCancelButton: true,
            confirmButtonColor: '#3085d6',
            cancelButtonColor: '#d33',
            confirmButtonText: 'Si, borrar!'
        }).then((result) => {
            return result.isConfirmed
        });
        if(!isConfirm){
            return;
        }
```

```jsx
        await axios.delete(`${endpoint}/empleado/${id}`).then(({data})=>{
        Swal.fire({
            icon:"success",
            text:data.message
        })
        getAllEmpleados()
    }).catch(({response:{data}})=>{
        Swal.fire({
            text:data.message,
            icon:"error"
        })
    })
}
    return (
        <div>
            <div className='d-grid gap-2'>
                <Link to="/crear" className='btn btn-success col-2 mt-2 mb-2 text-white'>CREAR</Link>
            </div>
            <table className='table table-striped'>
                <thead className='bg-primary text-white'>
                    <tr>
                        <th>NOMBRE</th>
                        <th>CORREO Name</th>
                        <th>CELULAR</th>
                        <th>ACCIONES</th>
                    </tr>
                </thead>
                <tbody>
                    { empleados.map( (empleado) => (
                        <tr key={empleado.id}>
                            <td>{empleado.nombre}</td>
                            <td>{empleado.correo}</td>
                            <td>{empleado.celular}</td>

                            <td>
                                <Link to={`/editar/${empleado.id}`} className='btn btn-info'>Edit</Link>
                                <button onClick={
()=>deleteEmpleado(empleado.id)} className='btn btn-danger'>Delete</button>
                            </td>
                        </tr>
                    ))}
                </tbody>
            </table>
        </div>
    )
    }
```

```
    export default MostrarEmpleados
```

COMPONENTE CREAR

```jsx
import axios from 'axios'
import React, {useState} from 'react'
import Swal from 'sweetalert2';
import { useNavigate } from 'react-router-dom'

const endpoint = 'http://localhost:8000/api/empleado'
const CrearEmpleado = () => {

    const [nombre, setNombre] = useState('')
    const [celular, setCelular] = useState('')
    const [correo, setCorreo] = useState('')
    const navigate = useNavigate()
    const [setValidationError] = useState({})

    const store = async (e) => {
        e.preventDefault();
        await axios.post(endpoint, {nombre: nombre, celular: celular, correo:
correo}).then(({data})=>{
            Swal.fire({
                icon:"success",
                text:data.message
            })
            navigate("/")
        }).catch(({response})=>{
            if(response.status===422){
                setValidationError(response.data.errors)
            }else{
                Swal.fire({
                    text:response.data.message,
                    icon:"error"
                })
            }
        })

    }
  return (
    <div>
```

```jsx
        <h2>Crear Nuevo Empleado</h2>
        <form onSubmit={store}>
            <div className='mb-3'>
                <label className='form-label'>Nombre</label>
                <input
                    value={nombre}
                    onChange={ (e)=> setNombre(e.target.value)}
                    type='text'
                    className='form-control'
                />
            </div>


            <div className='mb-3'>
                <label  className='form-label'>Celular</label>
                <input
                    value={celular}
                    onChange={ (e)=> setCelular(e.target.value)}
                    type='number'
                />
            </div>


            <div className='mb-3'>
                <label className='form-label'>Correo</label>
                <input
                    value={correo}
                    onChange={ (e)=> setCorreo(e.target.value)}
                    type='text'
                />
            </div>




            <button type='submit' className='btn btn-success'>Enviar</button>
        </form>
    </div>
  )
}


export default CrearEmpleado
```

COMPONENTE EDITAR

```javascript
import axios from 'axios'
import React,{useState, useEffect} from 'react'
import Swal from 'sweetalert2';
import { useNavigate, useParams } from 'react-router-dom'

const endpoint = 'http://localhost:8000/api/empleado/'

const EditarEmpleado = () => {


    const [nombre, setNombre] = useState('')
    const [celular, setCelular] = useState('')
    const [correo, setCorreo] = useState('')
    const navigate = useNavigate()
    const [setValidationError] = useState({})
    const {id} = useParams()

    const update = async (e) => {
        e.preventDefault();
        await axios.put(`${endpoint}${id}`, {
            nombre: nombre,
            celular: celular,
            correo: correo

        }).then(({data})=>{
            Swal.fire({
                icon:"success",
                text:data.message
            })
            navigate("/")
        }).catch(({response})=>{
            if(response.status===422){
                setValidationError(response.data.errors)
            }else{
                Swal.fire({
                    text:response.data.message,
                    icon:"error"
                })
            }
        })
        navigate('/')
    }
```

```jsx
    useEffect( () =>{

        const getEmpleadoById = async () => {
            const response = await axios.get(`${endpoint}${id}`)
            setNombre(response.data.nombre)
            setCelular(response.data.celular)
            setCorreo(response.data.correo)

        }
        getEmpleadoById()

    }, [])
    return (
    <div>
        <h2>Editar Empleado</h2>
        <form onSubmit={update}>
            <div className='mb-3'>
                <label className='form-label'>Nombre</label>
                <input
                    value={nombre}
                    onChange={ (e)=> setNombre(e.target.value)}
                    type='text'
                    className='form-control'
                />
            </div>

            <div className='mb-3'>
                <label className='form-label'>Celular</label>
                <input
                    value={celular}
                    onChange={ (e)=> setCelular(e.target.value)}
                    type='number'
                    className='form-control'
                />
            </div>

            <div className='mb-3'>
                <label className='form-label'>Correo</label>
                <input
                    value={correo}
                    onChange={ (e)=> setCorreo(e.target.value)}
                    type='text'
                    className='form-control'
                />
            </div>
```

```
                <button type='submit' className='btn
btn-success'>Actualizar</button>
        </form>
    </div>
  )
}


export default EditarEmpleado
```

APP.JS

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

import { BrowserRouter, Routes, Route, Link } from 'react-router-dom';
//se importa el componente
import MostrarEmpleados from './components/MostrarEmpleados';
import CrearEmpleado from './components/CrearEmpleado';
import EditarEmpleado from './components/EditarEmpleado';



function App() {
  return (
    <BrowserRouter>
    <div className="App">
    <nav className="navbar navbar-expand navbar-light bg-warning">
        <div className="nav navbar-nav">
            <Link className="nav-item nav-link active fs-3"
to={"/"}>Empleados </Link>
            <Link className="nav-item nav-link fs-3"
to={"/crear"}>CrearEmpleado </Link>
        </div>
    </nav>

    <Routes>
      <Route path='/' element={ <MostrarEmpleados/>} />
      <Route path='/crear' element={ <CrearEmpleado/>} />
      <Route path='/editar/:id' element={ <EditarEmpleado/>} />
    </Routes>
```

```
      </div>
    </BrowserRouter>
  );
}


export default App;
```

npm start
php artisan serve