



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

CARRERA:

Ingeniería De Software 5to Nivel

ASIGNATURA:

Aplicaciones Web

DOCENTE:

Ing. Guerrero Ulloa Gleiston Ciceron

TEMA:

React

GRUPO:

- Castro Lopez Pedro Leonardo
- Cruz Perez Justyn Keith
- Mendoza Bermello Angello Agustin
- Urbina Romero Isaias Abraham

INFORME DEL MÓDULO DE REGISTRO Y LISTADO DE DOCENTES

1. Introducción

Este proyecto es un sistema de gestión de docentes desarrollado en Java con Spring Boot en el backend y React en el frontend. En esta primera etapa se implementaron las funcionalidades de **registro de docentes y listado de los docentes registrados**. Todo el código está organizado siguiendo una estructura de paquetes por capas (controlador, entidad, repositorio, servicio, dto y configuración).

2. Backend

Estructura de paquetes:

Puedes tomar la primera captura desde tu IDE (ya lo hiciste antes). Debe mostrar la estructura `/src/main/java/com/universidad/contratacion/`.

2.1 Entidad Docente

Ubicada en el paquete entity.

Contiene los campos id, nombre, apellido, email, telefono y especialidad, todos con sus respectivos getters, setters, anotaciones de JPA y un constructor.

```
package com.universidad.contratacion.entity;

import jakarta.persistence.*;

@Entity
@Table(name = "docentes")
public class Docente {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String nombres;
    private String apellidos;
    private String email;
    private String telefono;
    private String especialidad;

    // Constructor vacío
    public Docente() {}

    // Constructor con parámetros
    public Docente(String nombres, String apellidos, String email,
String telefono, String especialidad) {
        this.nombres = nombres;
        this.apellidos = apellidos;
        this.email = email;
        this.telefono = telefono;
        this.especialidad = especialidad;
    }

    // Getters y Setters
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }

    public String getNombres() { return nombres; }
    public void setNombres(String nombres) { this.nombres = nombres; }

    public String getApellidos() { return apellidos; }
```

```

    public void setApellidos(String apellidos) { this.apellidos = apellidos; }

    public String getEmail() { return email; }
    public void setEmail(String email) { this.email = email; }

    public String getTelefono() { return telefono; }
    public void setTelefono(String telefono) { this.telefono = telefono; }

    public String getEspecialidad() { return especialidad; }
    public void setEspecialidad(String especialidad) {
this.especialidad = especialidad; }
}

```

2.2 Repositorio DocenteRepository

Ubicado en el paquete repository, extiende de JpaRepository<Docente, Long> para proveer operaciones CRUD automáticas.

```

package com.universidad.contratacion.repository;

import com.universidad.contratacion.entity.Docente;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface DocenteRepository extends JpaRepository<Docente, Long> {
    // Spring Data JPA ya nos da: save, findAll, findById, deleteById
}

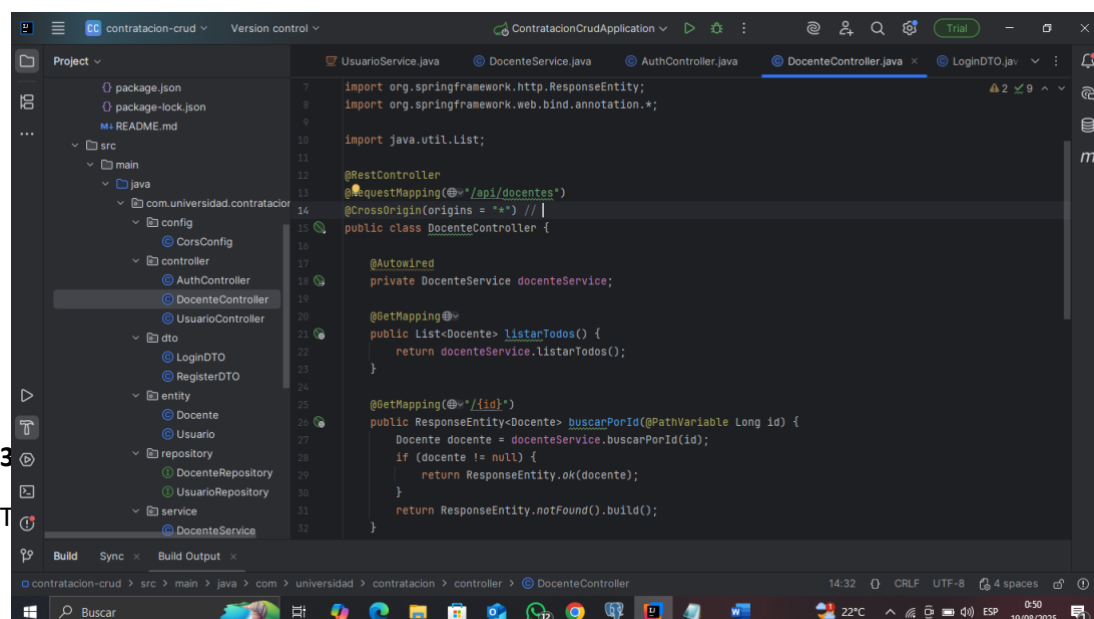
```

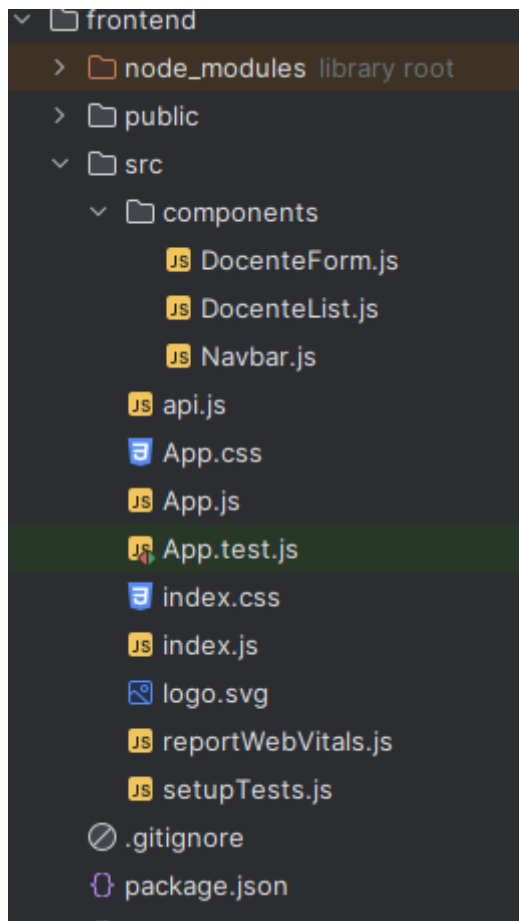
2.3 Controlador DocenteController

Se encuentra en el paquete controller.

Tiene los endpoints:

- @PostMapping("/docentes") para registrar docentes.
- @GetMapping("/docentes") para listar los docentes registrados.



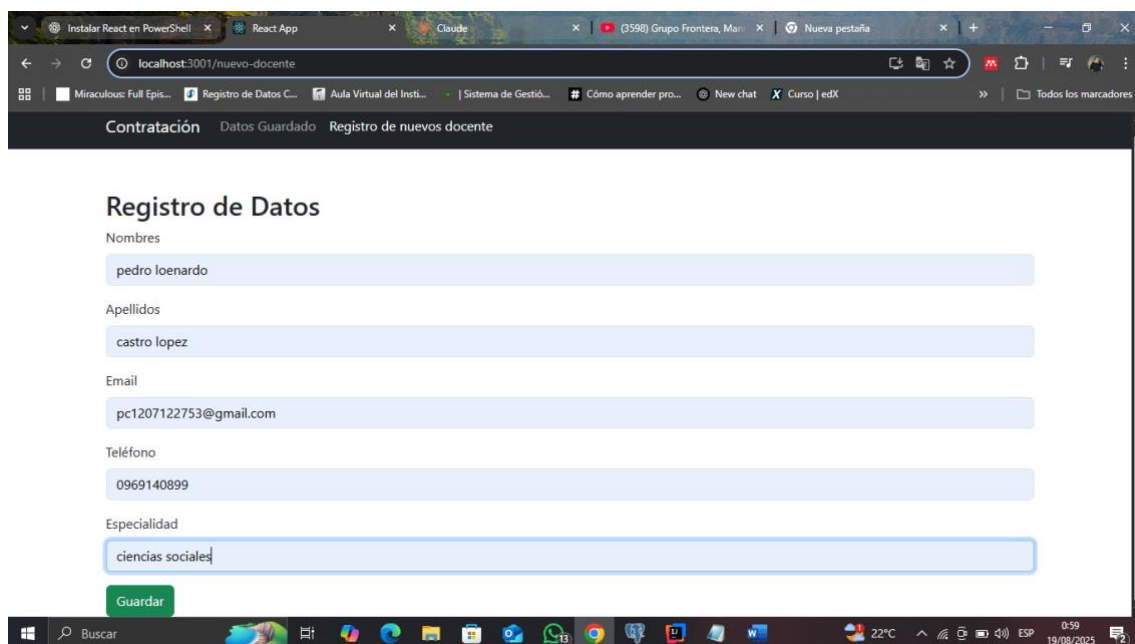


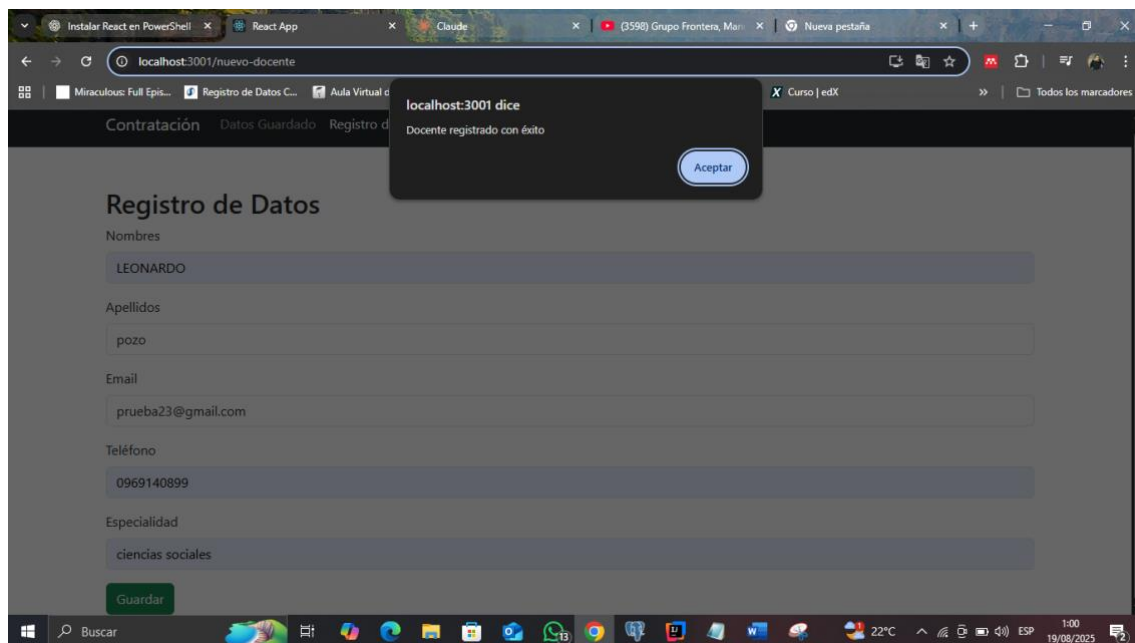
3.1 RegistroDatos.js

Este componente se utiliza para registrar docentes.

Contiene un formulario controlado con campos para nombre, apellido, email, teléfono y especialidad.

Cuando se envía, hace una petición POST al backend para guardar los datos.





Showing rows: 1 to 3 Page No: 1 of 1

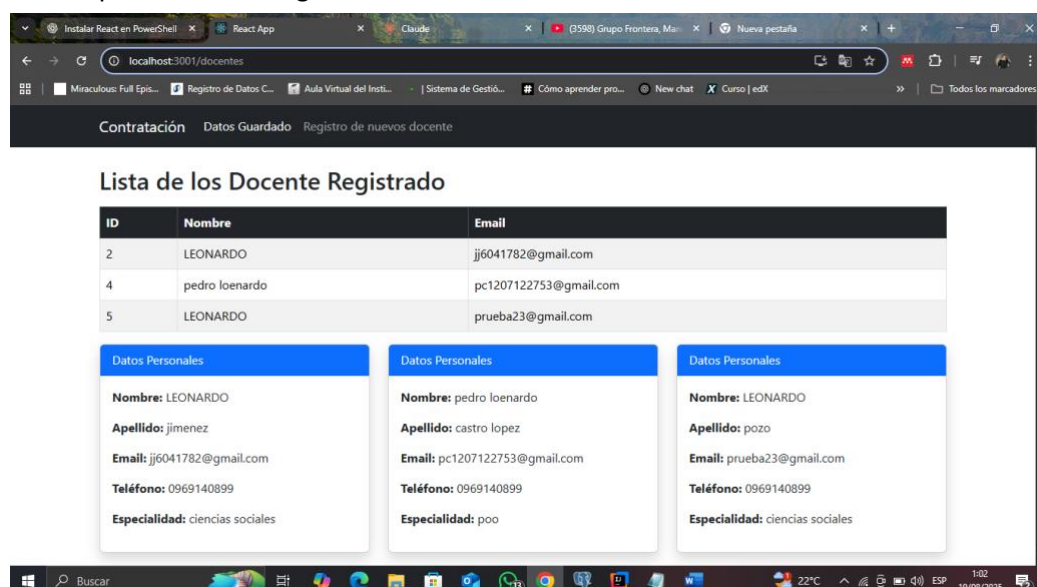
	id [PK] bigint	apellidos character varying (255)	email character varying (255)	especialidad character varying (255)	nombres character varying (255)	telefono character varying (255)
1	2	jimenez	jj6041782@gmail.com	ciencias sociales	LEONARDO	0969140899
2	4	castro lopez	pc1207122753@gmail.com	poo	pedro loenardo	0969140899
3	5	pozo	prueba23@gmail.com	ciencias sociales	LEONARDO	0969140899

Total rows: 3 Query complete 00:00:00.287 CRLF

3.2 ListaDocentes.js

Este componente se encarga de consumir el endpoint GET para listar los docentes y mostrarlos en una tabla.

Usa useEffect para hacer la carga inicial de datos.



3.3 Pasos para conectar React con Spring Boot

A continuación, se detallan los pasos realizados para lograr la conexión entre el frontend desarrollado en React y el backend en Spring Boot:

Paso 1: Crear el proyecto React dentro del proyecto principal

Desde la raíz del proyecto Spring Boot, se creó el frontend ejecutando:

```
npx create-react-app frontend
```

```
PS C:\Users\LEONARDO CASTRO\IdeaProjects\contratacion-crud> cd frontend
PS C:\Users\LEONARDO CASTRO\IdeaProjects\contratacion-crud\frontend> npx create-react-app . --use-npm

Creating a new React app in C:\Users\LEONARDO CASTRO\IdeaProjects\contratacion-crud\frontend.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

Esto genera la carpeta frontend/ con todos los archivos necesarios para una aplicación React.

Esta carpeta está dentro del mismo proyecto, pero separada del backend (no dentro de src/java).

Paso 2: Instalar Axios para hacer peticiones HTTP

Desde la terminal dentro de la carpeta frontend, se ejecutó:

```
npm install axios
```

Esto permite a React enviar y recibir datos desde el backend mediante llamadas HTTP.

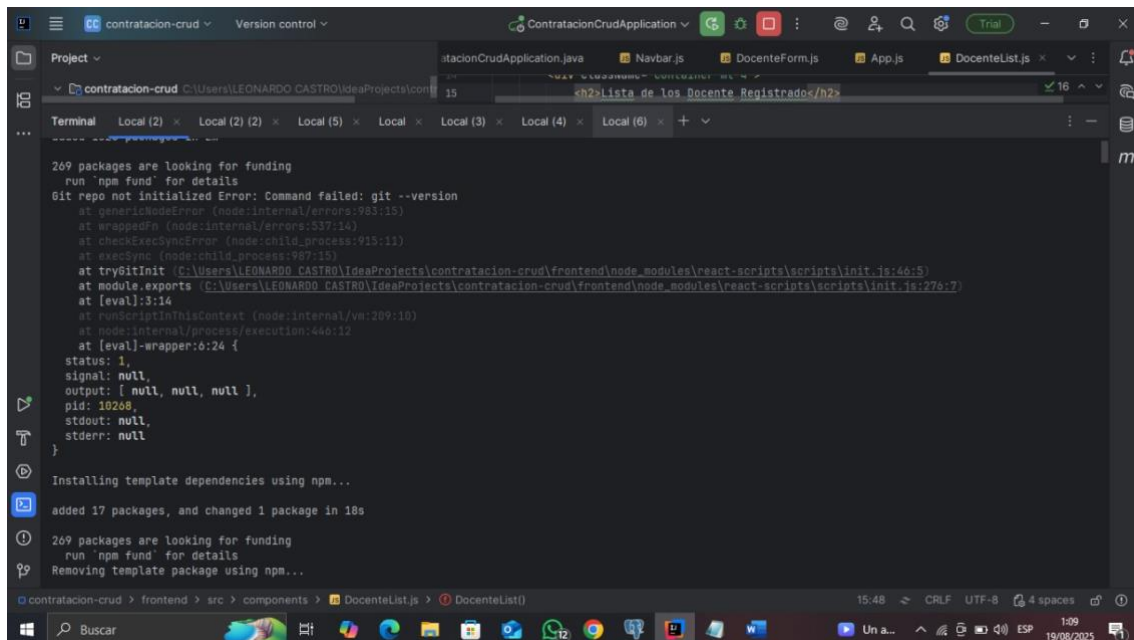
Paso 3: Crear los componentes en React

Se crearon dos componentes dentro de frontend/src/components/

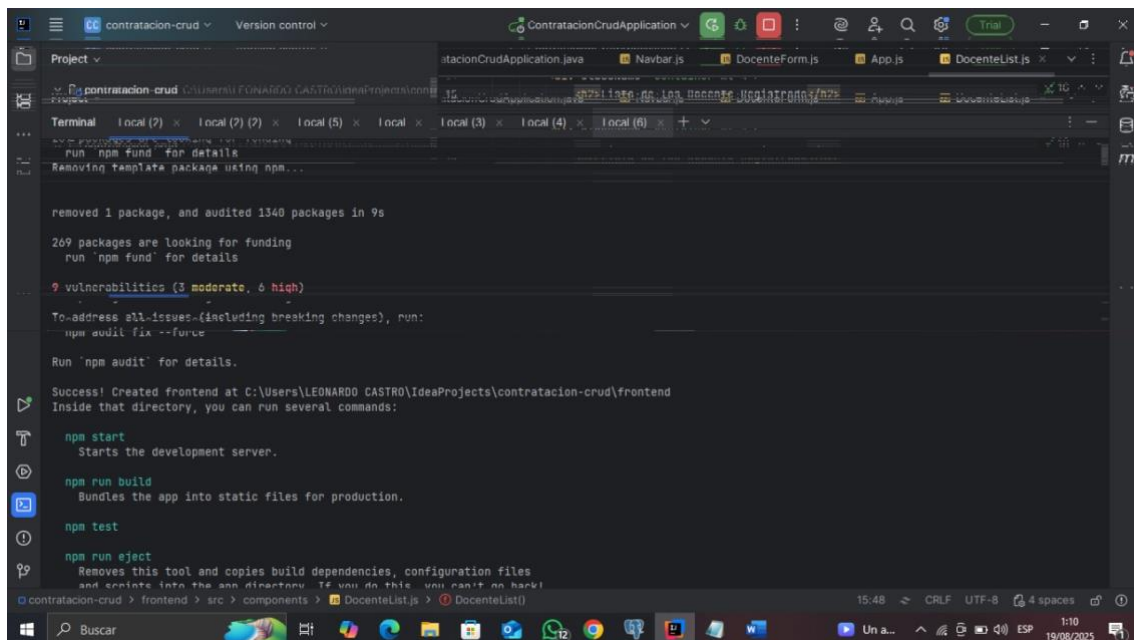
RegistroDatos.js – contiene el formulario de registro.

ListaDocentes.js – muestra la tabla con los datos.

Ambos componentes usan axios para comunicarse con el backend.



```
ContratacionCrudApplication
Project
  ContratacionCrudApplication.java
  Navbar.js
  DocenteForm.js
  App.js
  DocenteList.js
  ...
Terminal
  Local (2) x Local (2) x Local (5) x Local x Local (3) x Local (4) x Local (6) x +
  -----
  269 packages are looking for funding
  run 'npm fund' for details
  Git repo not initialized Error: Command failed: git --version
  at genericNodeError (node:internal/errors:983:15)
  at wrappedFn (node:internal/errors:537:14)
  at checkExecSyncError (node:child_process:915:11)
  at execSync (node:child_process:987:15)
  at tryGInit (C:\Users\LEONARDO CASTRO\IdeaProjects\contratacion-crud\frontend\node_modules\react-scripts\scripts\init.js:46:5)
  at module.exports (C:\Users\LEONARDO CASTRO\IdeaProjects\contratacion-crud\frontend\node_modules\react-scripts\scripts\init.js:276:7)
  at [eval]:3:14
  at node:internal/process/context: (node:internal/vm:209:10)
  at node:internal/process/execution:440:12
  at [eval]-wrapper:6:24 {
    status: 1,
    signal: null,
    output: [ null, null, null ],
    pid: 10268,
    stdout: null,
    stderr: null
  }
  Installing template dependencies using npm...
  added 17 packages, and changed 1 package in 18s
  269 packages are looking for funding
  run 'npm fund' for details
  Removing template package using npm...
  Contratacion-crud > frontend > src > components > DocenteList.js > DocenteList()
  15:48 CRLF UTF-8 4 spaces
```



```
ContratacionCrudApplication
Project
  ContratacionCrudApplication.java
  Navbar.js
  DocenteForm.js
  App.js
  DocenteList.js
  ...
Terminal
  Local (2) x Local (2) x Local (5) x Local x Local (3) x Local (4) x Local (6) x +
  -----
  removed 1 package, and audited 1340 packages in 9s
  269 packages are looking for funding
  run 'npm fund' for details
  9 vulnerabilities (3 moderate, 6 high)
  To address all issues (including breaking changes), run:
  npm audit fix --force
  Run 'npm audit' for details.
  Success! Created frontend at C:\Users\LEONARDO CASTRO\IdeaProjects\contratacion-crud\frontend
  Inside that directory, you can run several commands:
  npm start
    Starts the development server.
  npm run build
    Bundles the app into static files for production.
  npm test
  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and create into the app directory. If you do this, you can't go back!
  Contratacion-crud > frontend > src > components > DocenteList.js > DocenteList()
  15:48 CRLF UTF-8 4 spaces
```

We suggest that you begin by typing:

```
cd C:\Users\LEONARDO CASTRO\IdeaProjects\contratacion-crud\frontend
npm start
```

Happy hacking!

```
PS C:\Users\LEONARDO CASTRO\IdeaProjects\contratacion-crud\frontend> npm start
```

```
> frontend@0.1.0 start
```

```
> react-scripts start
```

```
(node:6920) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'ares' option.
(Use 'node --trace-deprecation ...' to show where the warning was created)
(node:6920) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'eares' option.
Starting the development server...
```

4. Resultado

En la vista de la aplicación, al abrir la interfaz, se pueden ver dos secciones importantes:

1. La sección "Registrar Docente", donde se llena y envía el formulario.
2. La sección "Lista de Docentes", donde se visualizan en tabla todos los docentes previamente registrados.

Puedes sacar capturas desde el navegador ejecutando la app localmente (<http://localhost:3000>), mostrando:

- La vista del fomulario lleno.
- La tabla con los docentes cargados.

5. Conclusión

En esta primera fase se implementó de forma exitosa el registro de docentes en la base de datos y su posterior visualización. Todo está conectado entre backend y frontend, y los datos quedan almacenados en PostgreSQL.

Este módulo servirá como base para seguir ampliando el sistema, incorporando funcionalidades como login, edición y eliminación de registros en fases posteriores.

Enlace de las diapositivas:

[https://www.canva.com/design/DAGv0JYlhXM/BoZetNWgqi709aK3Hyihwg/edit?utm_content=DAGv0JYlhXM&utm_campaign=designshare&utm_medium=link2&utm_source=sharebut
ton](https://www.canva.com/design/DAGv0JYlhXM/BoZetNWgqi709aK3Hyihwg/edit?utm_content=DAGv0JYlhXM&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)