

Apellido y Nombres:

Legajo:

Enunciado:

Se solicita desarrollar una aplicación frontend utilizando React.js que interactúe con una API REST ya desarrollada. La API REST expone dos URL para el recurso **deudor**:

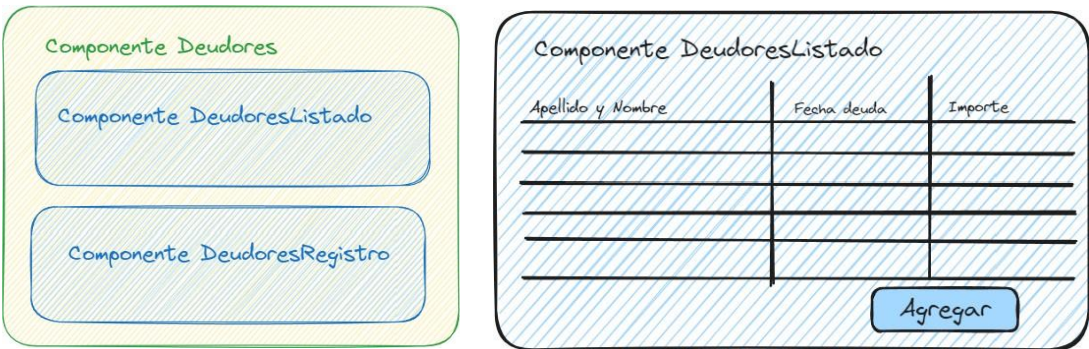
- 1. **POST: /deudores**: Esta ruta se utiliza para registrar los datos de un deudor con los siguientes campos: Apellido y nombre del deudor, importe, fecha de deuda y fecha límite de pago. Debes enviar una solicitud POST a esta URL con los datos del deudor desde un formulario de carga.
- 2. **GET: /deudores**: Esta ruta se utiliza para recuperar todos los datos de los deudores registrados utilizando Sequelize. Al realizar una solicitud GET a esta URL, devuelve todos los datos de los deudores registrados previamente.

Requerimientos:

- 1. Utiliza React.js para desarrollar la aplicación frontend.
- 2. Modifique el componente llamado **Menu.jsx**, que actúa como el menú principal de la aplicación y gestione las rutas utilizando React Router de forma tal permita acceder a la opción del menú **Deudores**. Este componente debe definir una ruta:
 - **/deudores**: que mostrará el componente **Deudores.jsx**.
- 3. Diseña el componente llamado **Deudores.jsx** que realice la gestión entre los componentes **DeudoresListado** y **DeudoresRegistro**. De manera inicial, debe visualizarse únicamente el componente **DeudoresListado** y se deben visualizar los datos de todos los deudores (provenientes del backend) .



- 4. El componente **DeudoresListado.jsx** deberá visualizar todos los datos de los deudores de la siguiente manera:



Si se presiona el botón agregar, se deberá visualizar únicamente el componente **RegistroDeudores** para permitir la carga del mismo.

- 5. Diseña el componente llamado **RegistroDeudores.jsx** que permita ingresar los datos de un deudor y validarlos mediante un formulario HTML e invoque a la api REST para guardarlo en la base de datos. Este componente debe incluir:
 - Un formulario con campos para ingresar los datos del deudor (Apellido y nombre del deudor, fecha de deuda e importe,).
 - Lógica de validación para asegurar que los datos ingresados sean correctos antes de enviarlos a la API.



Al registrar exitosamente los datos de un deudor, se deberá renderizar el componente **DeudoresListado** que muestre los datos de todos los deudores registrados hasta el momento.

El componente Registro debe tener un botón "Volver" que permita regresar al componente de menú visualizado inicialmente.

6. Utiliza solicitudes HTTP para interactuar con la API REST proporcionada mediante fetch o mediante la librería Axios vista en clases. Asegúrate de manejar adecuadamente las respuestas de la API, incluyendo casos de éxito y errores.

Entrega

Al finalizar el parcial realizar lo siguiente para poder evaluar:

- 1) Abrir un terminal y ubicarse en la carpeta /tmp (“cd /tmp”).
- 2) Ejecutar `git clone https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales2024/parcial2/3kC/XXXXXX` (reemplazar XXXXX por el número de su legajo y C por el número de división)
- 3) Moverse a la carpeta “cd XXXXX” (reemplazar XXXXX por el número de su legajo).
- 4) Crear un archivo .zip para el frontend y copiarlo en la carpeta del punto anterior (No olvidar eliminar la carpeta /Node_modules)
- 5) Subir los archivos zip al repositorio.
`git add .`
`git commit -m “resolución parcial”`
`git push`
- 6) Subir a Moodle los archivos comprimidos.
- 7) Subir a Moodle el enlace al repositorio del pto 2.
- 8) Subir los cambios al repositorio:
`https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales2024/parcial2/3kC/XXXXXX`

Evaluación

Consigna	Puntaje
Componente Menú	10
Componente Registro	40
Componente Consulta	20
Opción Volver	5
Opción Agregar	5
Diseño y comunicación entre componentes	10
Integración correcta con backend y Bootstrap	10

Tiempo de resolución: **1h 30 minutos.**

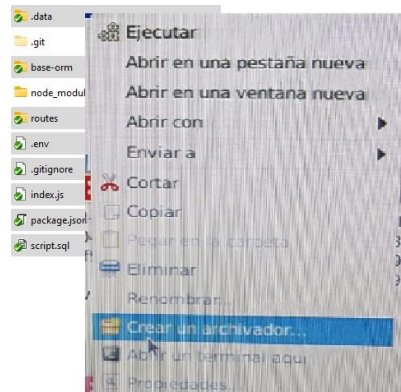
Escala de notas

NOTA	PORCENTAJE	CALIFICACIÓN
1		No Aprobado
2		No Aprobado
3		No Aprobado
4	55% a 57%	Aprobado
5	58% a 59%	Aprobado
6	60% a 68%	Aprobado
7	69% a 77%	Aprobado
8	78% a 86%	Aprobado
9	87% a 95%	Aprobado
10	96% a 100%	Aprobado

Instructivo para realizar los archivos ZIP

Realizar dos archivos .ZIP, uno para cada proyecto, sin incluir la carpeta **node_modules**: front_legajo.zip y back_legajo.zip (reemplazar legajo por su número de legajo) y el día del parcial indicará donde subir o enviar dichos archivos.

Los archivos ZIP se deberán realizar **SIN** la carpeta **node_modules**. Para ello dentro de la carpeta donde se encuentra el proyecto, Seleccionar todos los archivos y carpetas (menos node_modules) y del menú flotante selecciona **Crear un Archivador**



Indicar el formato de compresión (margen inferior izquierdo), seleccionando **ZIP**, e indicar el nombre y presionar el botón **CREAR**

