

Apellido y Nombres:Legajo:

Consigna

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 *reparacion*:

- 1. POST: /reparaciones: Esta ruta se utiliza para registrar los datos de una orden de servicio de reparación de electrodomésticos. Los datos a registrar son: fecha ingreso, tipo de electrodoméstico, Dni, solicitante, diagnostico, estado (reparada Si/No). Debes enviar una solicitud POST a esta URL con los datos del ingreso desde un formulario de carga.
- 2. GET: /reparaciones: Esta ruta se utiliza para recuperar todas las órdenes de reparación registradas hasta el momento utilizando Sequelize.

Requerimientos

- 1. Utiliza React.js para desarrollar la aplicación frontend.
- 2. Diseña un componente llamado Registro.jsx que permita registrar los datos de una orden de reparación mediante un formulario HTML. Este componente debe incluir:
 - Un formulario con campos para ingresar los datos de la orden.
 - Lógica de validación para asegurar que los datos ingresados sean correctos antes de enviarlos a la API.
 - Un componente hijo: Consulta.jsx

Reparaciones. Orden de servicio

DNI solicitante:

Fecha Ingreso

dd-mm-aaaa

Tipo electrodoméstico:

Diagnóstico

☐ Reparado

Registrar

Limpiar

- 3. Al registrar exitosamente los datos de una orden se deberá renderizar el componente hijo Consulta.jsx que muestre todas las órdenes de reparación registradas hasta el momento.
- 4. Implementa un componente llamado Consulta.jsx que recupere todos los ingresos mediante la API y los muestre en una tabla HTML.

Listado de órdenes de reparación

| Dni | Fecha ingreso | Tipo electrodoméstico | Diagnostico | Reparado |
|---------|---------------|-----------------------|-------------|----------|
| 98745 | 2024-06-02 | Plancha | sdfsaf | Si |
| 1234567 | 2024-06-17 | Plancha | No caliente | No |
| 1234567 | 2024-06-17 | Plancha | No caliente | No |
| 1234567 | 2024-06-17 | Plancha | No caliente | No |
| 7686587 | 2024-06-11 | Plancha | NADA | Si |

Volver

5. La **Consulta** 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/parciales-2024/parcial2/3k6/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 “resolucion 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/parciales-2024/parcial2/3k6/XXXXXX>

Evaluación

| Consigna | Puntaje |
|---|---------|
| Componente Registro (*) | 40 |
| Componente Consulta | 25 |
| Opción Volver | 10 |
| Diseño y comunicación entre componentes (*) | 15 |
| Integración correcta con backend(*) | 10 |

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

(*) Estos puntos son requisitos mínimos para aprobar.

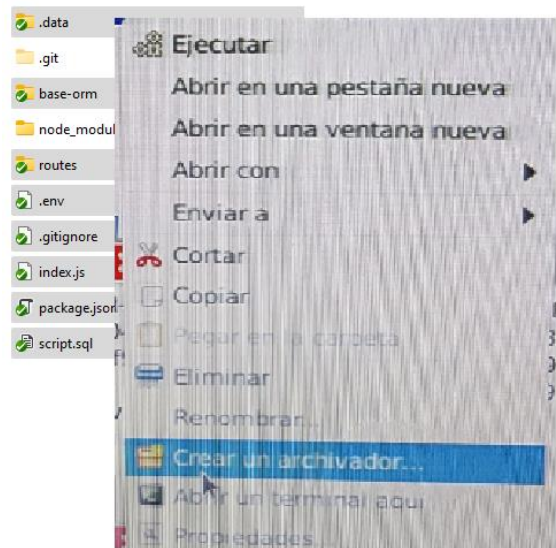
Escala de notas

| 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

Para la entrega del ejercicio, 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 numero 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**

