DESARROLLO DE SOFTWARE
Evaluación Sumativa

NOTA FINAL:

Fecha: 02/07/2024

Apellido y Nombres: .....Legajo: .....

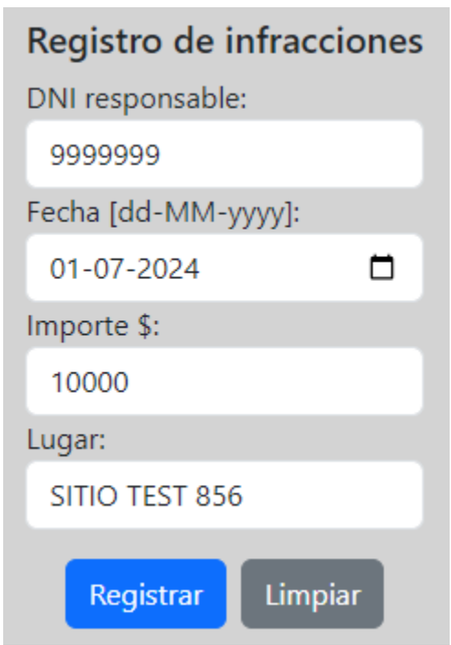
Consigna

Se solicita desarrollar una aplicación frontend utilizando React.js que interactúe con una API REST que deberá completar y hacer funcionar correctamente. La API REST expone dos URL para el recurso *infracciones*:

1. POST: /infracciones: Esta ruta se utiliza para registrar los datos de una infracción con los siguientes campos: DNI conductor, Importe, Fecha, lugar. Permite enviar una solicitud POST a esta URL con los datos de la infracción desde un formulario de carga (**ya desarrollada**).
2. GET: /infracciones/:id: Esta ruta se utiliza para recuperar todos los datos de una infracción identificada por Id (**a desarrollar**).

Requerimientos

1. Desarrollar la ruta /infracciones/:id descrita anteriormente.
2. Utiliza React.js para completar una aplicación frontend provista junto con el enunciado.
3. Diseña un componente llamado **`VerRegistro.jsx`**, hijo del componente Registro, que permita mostrar los datos completos (incluido el *Id*) de una infracción mediante un div HTML. Este componente debe incluir:
    - Una vista que muestre todos los datos de una infracción registrada mediante el componente **Registro.jsx** (**ya desarrollado**)
    - Podrá definir los estilos que cree más convenientes como por ejemplo una tarjeta (card de bootstrap)
    - Un botón **Registrar** que permita mostrar el formulario de carga nuevamente



4. Al registrar exitosamente los datos de una infracción se deberá renderizar el componente hijo VerRegistro.jsx que muestre los datos completos de la infracción.
5. Implementa un componente llamado **`VerRegistro.jsx`** que recupere todos los datos de la infracción registrada.

6. El componente **VerRegistro** debe tener un botón **Registrar** que permita regresar al componente visualizado inicialmente.
7. 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 recuperatorio 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/parcial2/3k**7**/XXXXX (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/parcial2/3k7/XXXXX

Evaluación

Consigna	**Puntaje**
Front: Componente VerRegistro (*)	30
Back: endpoint GET: /infracciones/:id (*)	30
Opción **Registrar** (en componente VerRegistro)	10
Diseño y comunicación entre componentes	20
Integración correcta entre front y back	10

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

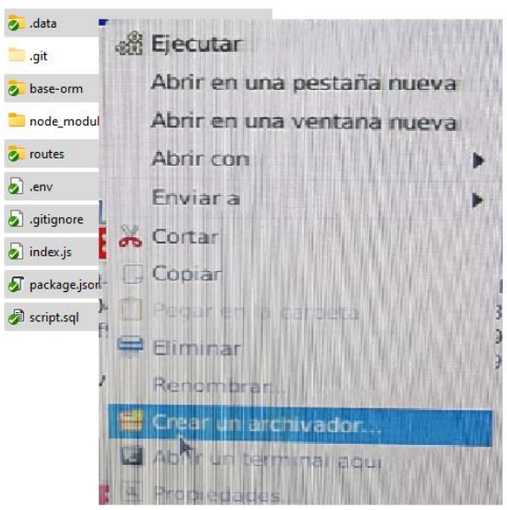
(*) Estos puntos son requisitos mínimos para aprobar.

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