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 *reserva*:

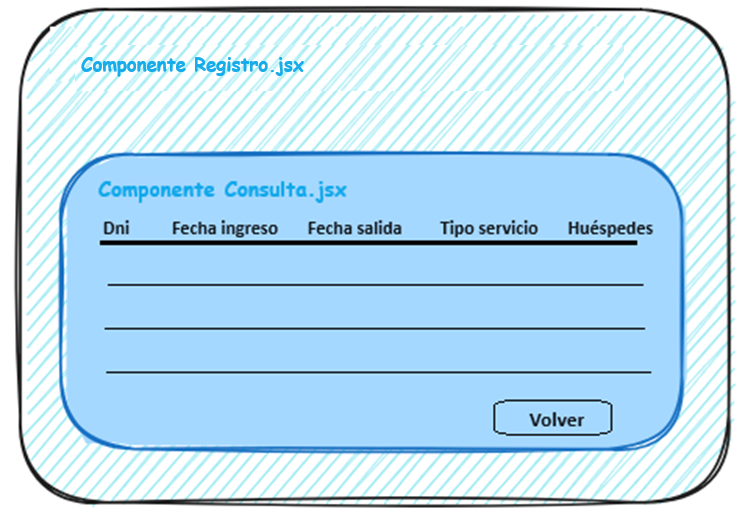
1. POST: /reservas: Esta ruta se utiliza para registrar los datos de una reserva de hotel con los datos fecha ingreso, fecha de salida, cantidad de huéspedes, tipo de estadía, DNI responsable de reserva. Debes enviar una solicitud POST a esta URL con los datos de la reserva desde un formulario de carga.
2. GET: /reservas: Esta ruta se utiliza para recuperar todos los datos de la entidad utilizando Sequelize. Al realizar una solicitud GET a esta URL, se deben devolver todos los datos de las reservas registradas previamente.

Requerimientos

1. Utiliza React.js para desarrollar la aplicación frontend.
2. Implementa un componente principal llamado Menu.jsx que actúe como el menú principal de la aplicación y gestione las rutas utilizando React Router. Este componente debe definir una ruta:
   * /registro: que mostrará el componente Registro.jsx.
3. Diseña un componente llamado Registro.jsx que permita ingresar los datos de una reserva y validarlos mediante un formulario HTML. Este componente debe incluir:
   * Un formulario con campos para ingresar los datos de la entidad.
   * Lógica de validación para asegurar que los datos ingresados sean correctos antes de enviarlos a la API.
   * Un componente hijo: Consulta.jsx



1. Al registrar exitosamente los datos de una reserva se deberá renderizar el componente hijo Consulta.jsx que muestre los datos de todas las reservas registradas hasta el momento.
2. Implementa un componente llamado Consulta.jsx que recupere todas las reservas mediante la API y los muestre en una tabla HTML.



1. Tanto el componente **Registro** como el de **Consulta** deben tener un botón **Volver** que permita regresar al componente de menú visualizado inicialmente.
2. 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-2024/3k**C**/XXXXX](https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales-2024/3kC/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

1. Subir a Moodle los archivos comprimidos.
2. Subir a Moodle el enlace al repositorio del pto 2.
3. Subir los cambios al repositorio: <https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales-2024/3kC/XXXXX>

Evaluación

|  |  |
| --- | --- |
| **Consigna** | **Puntaje** |
| Componente Menú | 10 |
| Componente Registro | 40 |
| Componente Consulta | 20 |
| Opción **Volver** | 10 |
| Diseño y comunicación entre componentes | 10 |
| Integración correcta con backend | 10 |

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

(\*) Estos puntos son requisitos mínimos para aprobar.

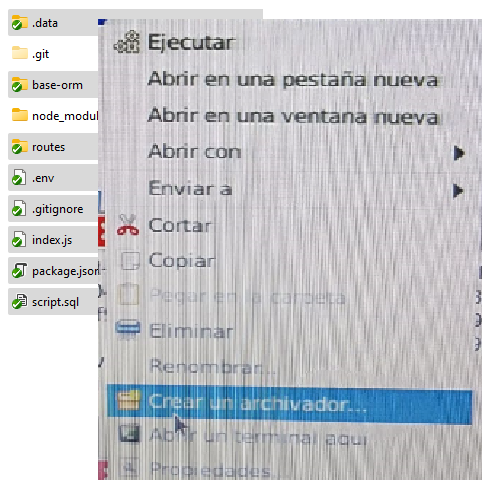
Escala de notas

|  |  |  |
| --- | --- | --- |
| **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  
