

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

### Instrucciones:

- 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-2023/3k5/XXXXX>  
(reemplazar XXXXX por el numero de su legajo)
- 3) Moverse a la carpeta "cd XXXXX" (reemplazar XXXXX por el número de su legajo).
- 4) Descargar el archivo parcial\_dds\_3k5\_20230621.zip en la carpeta /tmp/XXXXX
- 5) Descomprimir el archivo con el siguiente comando:

```
unzip parcial_dds_3k5_20230621.zip
```

- 6) Tanto en la carpeta backend como en la carpeta frontend borrar la carpeta node\_modules (si existe).
- 7) Programar la funcionalidad descrita bajo el título **Consigna**.
- 8) Subir los cambios al repositorio:  
<https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales-2023/3k5/XXXXX>  
(reemplazar XXXXX por el número de su legajo).

Seguir los siguientes pasos:

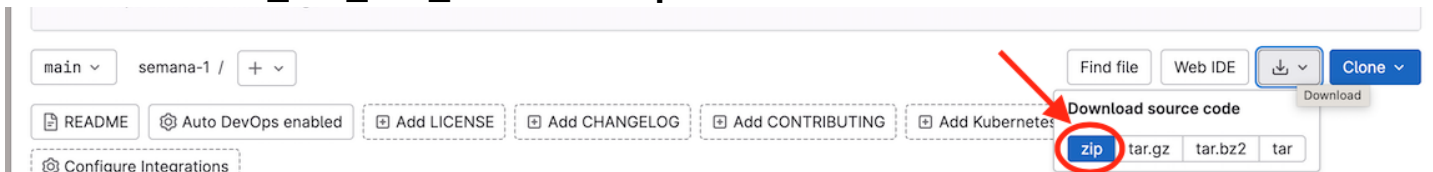
- a. Asegurarse que los cambios están comiteados al repositorio local.

**git status**

- b. Subir el código al repositorio del punto anterior.

**git push**

- 9) Subir a Moodle el enlace al repositorio.
- 10) Descargar repositorio de GitLab en un archivo zip con nombre **resolucion\_dds\_3k5\_20230621.zip**:



- 11) Subir a Moodle el comprimido **resolucion\_dds\_3k5\_20230621.zip** generado en el punto 10.

### Importante!

- Para instalar las librerías dependientes del back y del front:
  1. Eliminar (**si existe**) el archivo package-lock.json en ambos proyectos.
  2. Ejecutar el comando **npm install** en ambos proyectos (misma ruta donde está el archivo **package.json**).
- Desarrollar el **backend** en el port 4000 y para el **frontend** en el port 3000.

### Consigna:

**Desarrollo del back:** Se deberá agregar una funcionalidad de consulta para un recurso nuevo **Albumes** que contiene información sobre álbumes musicales:

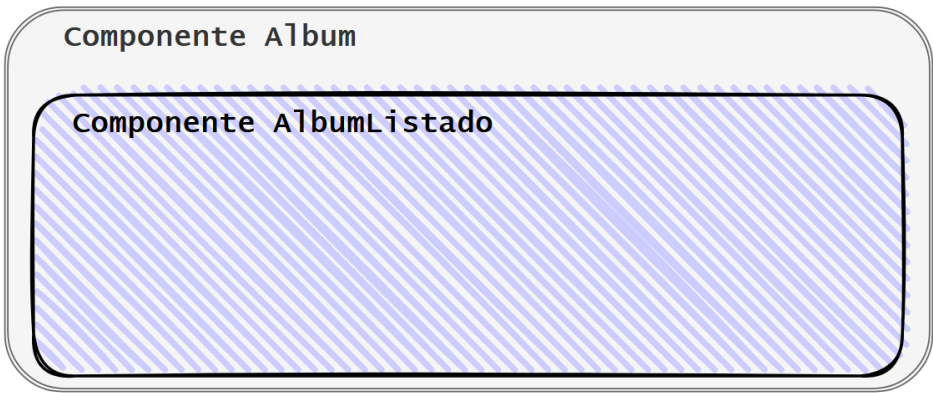
- Crear el modelo de sequelize que mapee la tabla **albumes**, acorde a la estructura. La tabla ya se encuentra creada y con datos en la base de datos ./data/pymes.sqlite:

álbumes	
Id	int
Artista	text
Album	text
Genero	text
Soporte	text
Precio	integer

- Crear una ruta webapi con las siguientes características:
  - Un router para el endpoint Álbumes que mediante el verbo/método **GET**
    - Reciba un parámetros: artista.
    - Utilizando el modelo álbumes de sequelize recupere los registros filtrando por el parámetro recibido (Deberá devolver todos los registros que contengan el valor del parámetro artista)
    - En caso de que el parámetro venga en blanco (cadena vacía) debe devolver todos los registros.
    - Retornar y mostrar solo los campos: artista, álbum, género y precio.
  - La aplicación deberá registrar como middleware el **router** con la ruta `"/api/álbumes"`.
- Asegure la funcionalidad probando las siguientes url desde el browser:
  - `http://localhost:4000/api/álbumes`  
Debe devolver todos los registros.
  - `http://localhost:4000/api/álbumes?artista=monk`  
Debe devolver los registros filtrados.

**Desarrollo del frontend:** Se deberá agregar una interface de consulta para el recurso "álbumes" desarrollado en el backend

- Desarrollar un componente Album con el siguiente esquema



- Agregar el código necesario para poder realizar la consulta de álbumes. Se deberá realizar los componentes de react y consumir el backend desarrollado en el punto anterior.
- En el componente **Album**, incluir la interfaz de búsqueda con el campo a filtrar ("artista") y el botón buscar.
  - Deberá cumplir los siguientes requisitos:
    - Consumir vía axios el endpoint del back.
    - Utilizar para el formulario de búsqueda la librería react-hook-form

- En el componente **AlbumListado** incluir una tabla de html para mostrar los resultados de la búsqueda, incluya en la misma los campos mencionados anteriormente.
- Agregar al componente menú de la aplicación, el acceso al nuevo componente.

Evaluación

Consigna	Puntaje
<b>BackEnd</b>	
Modelo Sequelize	10
EndPoint GET (*)	20
Filtrar con parámetros	20
<b>FrontEnd</b>	
Componente Album	10
Componente AlbumListado (*)	20
Filtrar con parámetro	10
Componente Menu	10
<b>Total</b>	

(\*) 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

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