

Instrucciones:

Importante: no hagas desde el punto 6 al 10 pero podés ir viendo cómo vas a tener que entregar el parcial

- 1) Abrir un terminal y ubicarse en la carpeta /tmp ("cd /tmp").
- 2) Ejecutar git clone Moverse a la carpeta "cd XXXXX" (reemplazar XXXXX por el número de su legajo).
- 3) Descomprimir el archivo con el siguiente comando
unzip simulacro_3k5_1.zip
o
tar xzvf simulacro_3k5_1.zip
- 4) Tanto en la carpeta backend como en la carpeta frontend borrar la carpeta node_modules.
- 5) Programar la funcionalidad descrita bajo el título **Consigna**.
- 6) Comprimir la carpeta simulacro_3k5_1 usando el comando zip -r
resolucion_simulacro_3k5_1.zip *
o el comando
tar -cvzf resolucion_simulacro_3k5_1.tar.gz
- 7) Subir a Moodle el comprimido generado en el punto 8.
- 8) Subir a Moodle el enlace al repositorio.
- 9) Borrar el zip generado.
- 10) Subir los cambios al repositorio: <https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/simulacros-2023/3k5/XXXXX> (reemplazar XXXXX por el número de su legajo).

Seguir los siguientes pasos:

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

git status

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

git push

- Desarrollar el **backend** y **frontend** de una aplicación para el back el port 4000 y para el front el port 3000.
- **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**).

Consigna

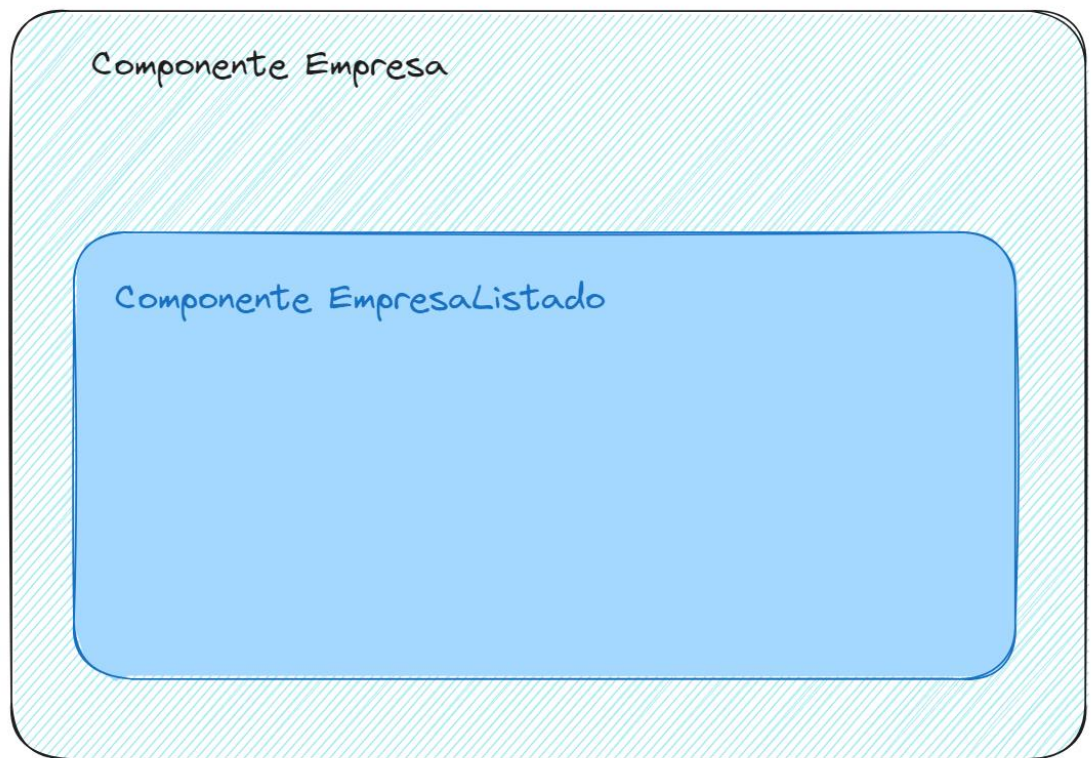
Desarrollo del back: Se deberá agregar una funcionalidad de consulta para un recurso nuevo "Empresas"

- Desarrollar el código para conectar con la BD, crear la tabla si no existe e insertar los valores asignados. En el archivo script_empresas.sql (que está en la raíz del proyecto) están los script sql necesarios.
- Crear el modelo de sequelize que mapee la tabla Empresas, acorde a la estructura descrita en el punto anterior.
- Programar una ruta webapi con las siguientes características:
 - Un router para el endpoint Empresas que mediante el verbo/método **GET** reciba un parámetro: RazonSocial y utilizando el ORM del punto anterior, recupere los registros filtrando por el parámetro recibido. Deberá devolver todos los registros en cuyo campo RazonSocial contenga el valor del parámetro recibido. Devolver todos los campos de la tabla.

- La aplicación deberá registrar como middleware el **router** con la ruta “/api/empresas”.
- Asegure la funcionalidad probando las siguientes url desde el browser:
 - http://localhost:4000/api/empresas
Debe devolver todos los registros.
 - http://localhost:4000/api/empresas?RazonSocial=MICRO
Debe devolver los registros que contengan “MICRO”

Desarrollo del frontend: se deberá agregar una interface de consulta para el recurso “empresas” desarrollado en el backend

- Desarrollar un componente Empresa con el siguiente esquema:



- Agregar el código necesario para poder realizar la consulta de empresas. Se deberá realizar los componentes de react y consumir el backend desarrollado en el punto anterior.
- En el componente Empresa, incluir la interface de búsqueda con el campo a filtrar (“RazonSocial”) 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 EmpresaListado incluir una tabla de html para mostrar los resultados de la búsqueda, incluya en la misma todos los campos. (no se pide paginación)
- Agregar al componente menú de la aplicación, el acceso al nuevo componente.

Evaluación

Consigna	Puntaje
BackEnd	
Conectar Db, crear tabla ...	10
Modelo Sequelize	10
EndPoint GET (*)	20
Filtrar con parámetro	10
FrontEnd	
Componente Empresa	10
Componente EmpresaListado (*)	20
Filtrar con parámetro	10
Componente Menu	10
Total	

(*) 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.