

---

**Primer Parcial de Desarrollo de Software 3k6**

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

**Enunciado e instrucciones:**

1. Abrir un terminal y ubicarse en la carpeta /tmp ("cd /tmp").
2. Ejecutar en la terminal: **git clone** [https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales-2023/3k6/\[legajo\]](https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales-2023/3k6/[legajo]) (reemplazar **[legajo]** por el número de su legajo)
3. Moverse a la carpeta "cd **[legajo]**" (reemplazar **[legajo]** por el número de su legajo).
4. Descargar el archivo parcial\_dds\_3k6\_a.zip (desde la UV pestaña Evaluaciones>>Parcial) en la carpeta /tmp/**[legajo]**
5. Descomprimir el archivo con el siguiente comando: **unzip** parcial\_dds\_3k6\_a.zip o "tar xzvf parcial\_dds\_3k6\_a.zip".
6. Programar la funcionalidad descrita bajo el título **Consigna**.
7. Comprimir la carpeta parcial\_dds\_3k6\_a usando el comando **zip -r** resolucion\_dds\_3k6\_a.zip \* o el comando "tar -cvzf resolucion\_dds\_3k6\_a.tar.gz"
8. Subir a Moodle el comprimido generado en el *punto 8*.
9. Subir a Moodle el enlace al repositorio.
10. Borrar el zip generado.
11. Subir los cambios al repositorio: [https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales-2023/3k6/\[Legajo\]](https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales-2023/3k6/[Legajo]) (reemplazar **[legajo]** 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**).

**I- Requerimientos del back:**

Se deberá agregar una funcionalidad de consulta para el recurso: "Pedidos".

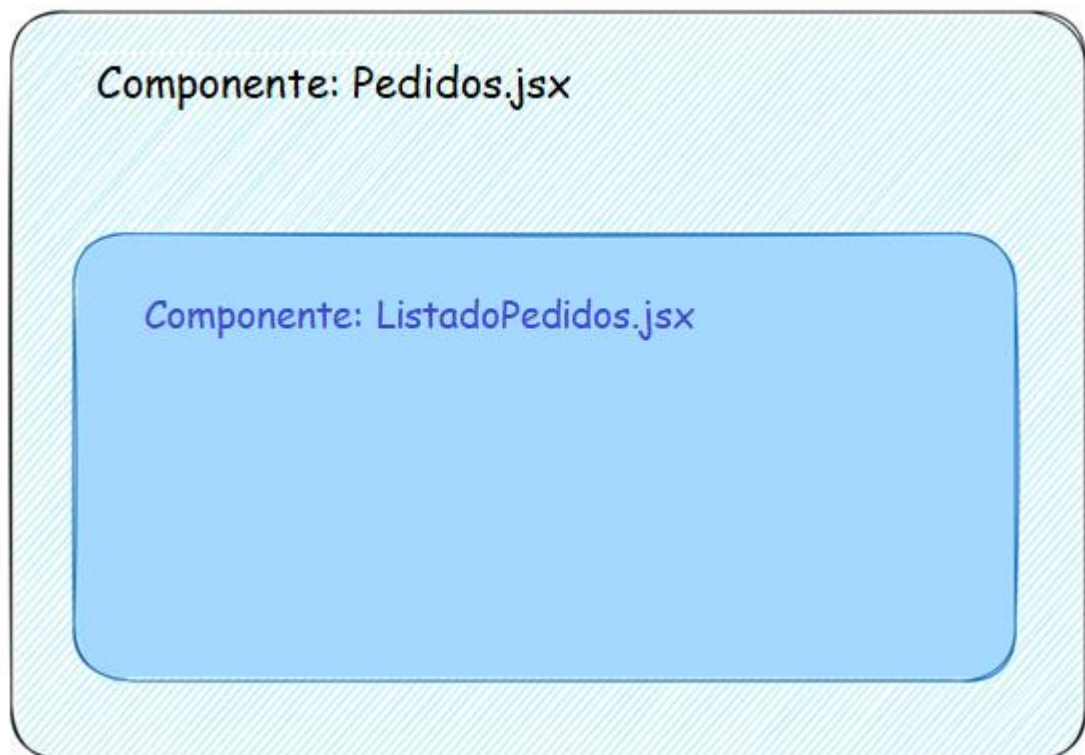
- En el archivo *script\_pedidos.js* (que está en la raíz del proyecto) se encuentra el código necesario para conectarse con la base de datos y generar el modelo correspondiente. Ud. deberá utilizar conveniente este script para que cuando arranque el proyecto back se cree e inicialice el origen de datos del proyecto.
- Programar una ruta con las siguientes características:

- Un router para el **endpoint de pedidos** que mediante el verbo/método **GET** reciba como parámetros: DireccionEntrega y Entregado, y utilizando el ORM del punto anterior, recupere los registros filtrando por el/los parámetros recibidos. En caso de no recibir ningún parámetro deberá recuperar todos los registros 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/pedidos>  
Debe devolver todos los registros.
  - <http://localhost:4000/api/pedidos?DireccionEntrega='san'&Entregado='S'>  
Debe devolver los registros que contengan **"san"** en el campo DireccionEntrega y hayan sido entregados (campo Entregado = 'S')

## **II- Requerimientos del frontend:**

Deberá refactorizar una interface de consulta para el recurso "pedidos" desarrollado en el backend

- Desarrollar un componente **Pedidos** con el siguiente esquema:



- Agregar el código necesario para poder realizar la consulta de pedidos. Se deberá completar los componentes de react proporcionados y consumir el backend desarrollado en el punto anterior.

⇒ En el componente Pedidos, incluir la interface de búsqueda con el campo a filtrar (“DireccionEntrega”, “Entregado”) y el botón Buscar.

Deberá cumplir los siguientes requisitos:

- Consumir vía **axios** el endpoint del back.
- Podrá utilizar para el formulario de búsqueda la librería **react-hook-form** o un formulario **HTML 5** con las variables de estado que considere necesarias.

⇒ En el componente **ListadoPedidos** completar el código HTML para mostrar los resultados de la búsqueda con todos los campos del modelo (sin paginación).

### **III- Condiciones de entrega:**

Durante el desarrollo del parcial, deberá trabajar con su repositorio local. Al finalizar el desarrollo y luego de asegurarse que el código está commiteado localmente, deberá subir el código (git push) a un repositorio remoto que se ha generado en gitlab para cada alumno, identificado con su legajo: [https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales-2023/3k6/\[legajo\]](https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales-2023/3k6/[legajo]) (reemplazar **[legajo]** por el número de su legajo)

Seguir los siguientes pasos:

1. Asegurarse que los cambios están comiteados al repositorio local.  
**git status**
2. Cambiar el repositorio remoto al identificado con su legajo  
**git remote set-url origin** [https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales-2023/3k6/\[legajo\]](https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/parciales-2023/3k6/[legajo])
3. Subir el código al repositorio del punto anterior.  
**git push**

### **Tabla de valoración de los ítems evaluados**

Consigna	Puntaje
<b>BackEnd</b>	
Crear Api-Rest	20
Router GET con parámetros(*)	30
<b>FrontEnd</b>	
Componente Pedidos (*)	30
Componente ListadoPedidos	20
<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