DESARROLLO DE SOFTWARE
Simulacro Examen Parcial

NOTA FINAL:

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

Enunciado:

- Desarrollar el backend y frontend de una aplicación utilizando 2 instancias de Visual Studio Code, una para el back y la otra para el front. Se deberá generar una funcionalidad nueva a los proyectos existentes.
- El proyecto del backend está en el siguiente repositorio GIT
  https://github.com/jiglesias2000/dds-backend-2025.git
- El proyecto del frontend está en el siguiente repositorio GIT
  https://github.com/jiglesias2000/dds-fronted-2025.git

Clonar ambos repositorios en la carpeta parcial2 de su estación de trabajo. Y realizar las modificaciones solicitadas en dichos proyectos.

Importante para instalar las librerías dependientes del back y del front: Ejecutar el comando npm install en ambos proyectos (misma ruta donde está el archivo package.json).

Tiempo de resolución: 1h 30 minutos.

Desarrollo del backend: Se deberá agregar una funcionalidad de consulta para un recurso nuevo “Contratos”

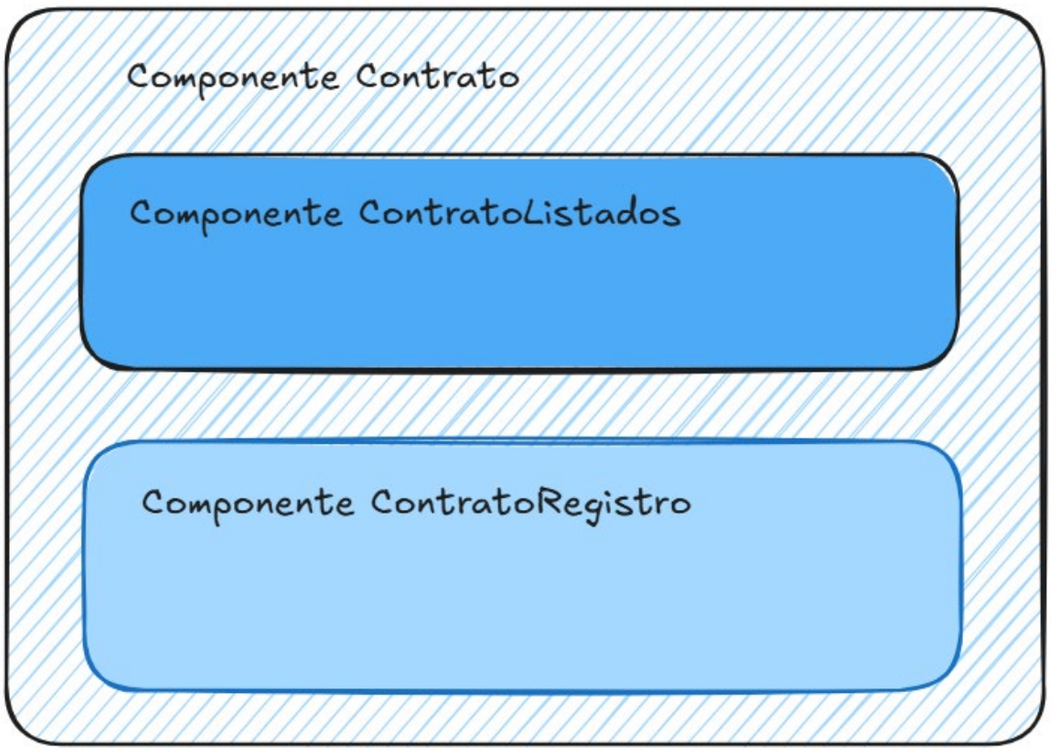
- Desarrollar el código para conectar con la Base de datos, crear la tabla “Contratos” si no existe e insertar los valores asignados para sus registros. En el archivo **datos_contratos.json** (que está en la raíz del proyecto) están los datos necesarios.
- Crear el modelo de sequelize que mapee la tabla "contratos", acorde a la estructura descrita en el punto anterior.
- Programar para una ruta webapi con las siguientes características:
  - Un router para el endpoint contratos que mediante
    - el verbo/método GET reciba un parámetro: NombreContrato y utilizando el ORM del punto anterior, recupere los registros filtrando por el parámetro recibido Deberá devolver todos los registros en cuyo campo NombreContrato contenga el valor del parámetro recibido (sin tener en cuenta mayúsculas/minúsculas), ordenados por NombreContrato. Devolver todos los campos de la tabla.
    - el verbo/método POST reciba un json y utilizando el ORM del punto anterior, inserte en la tabla un registro nuevo.
  - La aplicación deberá registrar como middleware el router con la ruta “/api/contratos”.
- Asegure la funcionalidad probando las siguientes url desde el browser:
  - http://localhost:3000/api/contratos

    Debe devolver todos los registros.
  - http://localhost:3000/api/contratos?NombreContrato=casa

    Debe devolver los registros que contengan “casa” (sin tener en cuenta mayúsculas/minúsculas)

Desarrollo del frontend: se deberá agregar una interface visual de consulta/alta para el recurso “contratos” desarrollado en el backend

- Desarrollar un componente Contrato con el siguiente esquema:



- Agregar el código necesario para poder realizar la consulta de contratos. Se deberá realizar los componentes de react y consumir el backend desarrollado en el punto anterior.
- En el componente Contrato, incluir la interface de búsqueda con el campo a filtrar: “NombreContrato”, un botón “Buscar” y un botón “Agregar” (cada uno con su respectiva funcionalidad)

  Deberá cumplir los siguientes requisitos:

  - Comunicarse vía axios con el endpoint del back.
  - Utilizar para el formulario de búsqueda y de alta, la librería react-hook-form

- En el componente ContratoListado incluir una tabla de html para mostrar los resultados de la búsqueda, incluya en la misma todos los campos. (no se pide paginación) Hacer uso del framework bootstrap y formatear correctamente los campos fecha (dd/mm/aaaa) y moneda (con 2 decimales siempre)

- En el componente ContratoRegistro permitir agregar un registro nuevo, pidiendo todos los campos del recurso contrato, con la interface adecuada a cada tipo de datos y validando que sean todos datos requeridos; y particularmente el campo NombreContrato que no supere los 70 caracteres, ni que sea menor a 5 caracteres.

- 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 (*)	10
Filtrar con parámetro	10
FrontEnd	
Componente Contrato	10
Componente ContratoListado (*)	15
Componente ContratoRegistro (*)	15
Filtrar con parámetro	10
Componente Menu	10
Total	

(*) Estos puntos funcionando 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