

Apellido y Nombres:Legajo:

Enunciado:

Desarrollar el 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 al front existente y consumir la api ya disponible en el back

- 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>

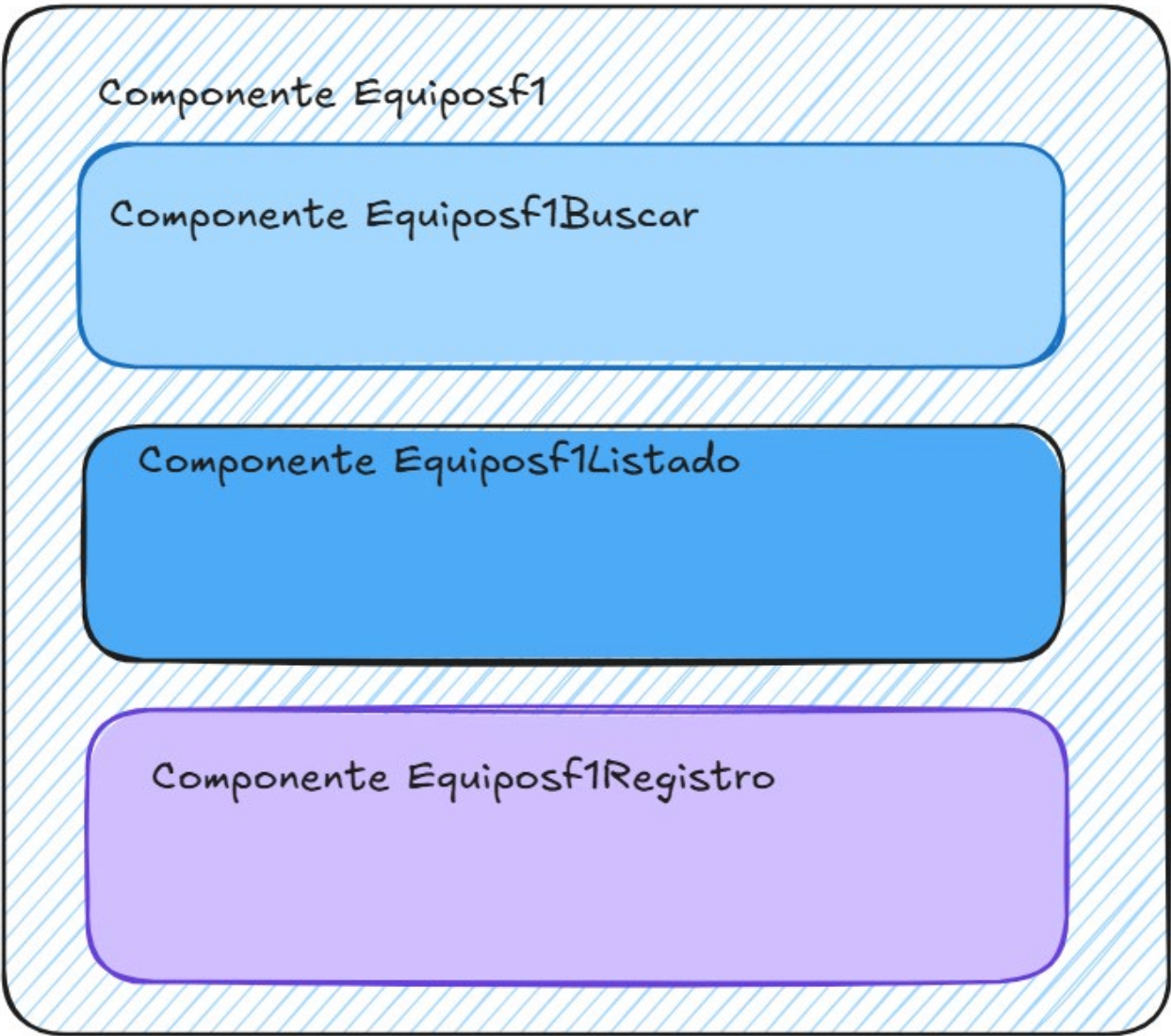
Clonar los repositorios en la carpeta parcial2 de su estación de trabajo. Y realizar las modificaciones solicitadas en el proyecto frontend (en el backend no hay que hacer cambios).

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

Tiempo de resolución: 1h 30 minutos.

Desarrollo del frontend: se deberá desarrollar una interface visual para buscar con filtro, ver los resultados de la búsqueda y dar de alta al recurso “equiposf1”, disponible en la url del backend: <http://localhost:3000/api/equiposf1>

- Desarrollar un componente react Equipof1 con el siguiente esquema:



- Agregar el código necesario para poder realizar la comunicación entre los componentes del esquema propuesto. Se deberá consumir el backend ya funcional mediante **axios**.
- En el componente EquipoflBuscar, incluir la interface de búsqueda con el campo a filtrar: “**NombreCorredor**”, un botón “**Buscar**” y un botón “**Agregar**” (cada uno con su respectiva funcionalidad)
- En el componente EquipoflListado incluir una **tabla de html** para mostrar los resultados de la búsqueda, incluya en la misma todos los campos. (no implementar paginación) Hacer uso del framework **bootstrap** y formatear correctamente los campos fecha (dd/mm/aaaa) y moneda (con 2 decimales siempre).
- En el componente EquipoflRegistro permitir agregar un registro, pidiendo todos los campos del recurso equipofl (menos IdEquipofl), con la interface adecuada a cada tipo de dato y validando que sean **todos datos requeridos**; y particularmente el campo **NombreCorredor** que no supere los 50 caracteres, ni que sea menor a 7 caracteres. Específicamente para el campo **NombreCorredor** si no se cumplen las validaciones mostrar un mensaje para informar la situación. Utilizar en este componente la librería **react-hook-form**.
- Agregar al componente **menú** de la aplicación, el acceso al nuevo componente.

Evaluación:

Consigna	Puntaje
FrontEnd	
Componente Equipof1	10
Componente Equipof1Buscar (sin filtro)	20
Componente Equipof1Buscar Filtro buscar	10
Componente Equipof1Listado	20
Componente Equipof1Registro	30
Componente Menu	10
Total	

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