SPRINT 5. GraphQL y NodeJs y aplicación de interfaz gráfica con ReactJs

Desarrollo de aplicaciones web

Marcela Peñaranda Ortega – C.C: 1091670131

Brandom Mosquera Cadena– C.C: 1013674995

Sergio González Betin -C.C: 92529506

Miguel González Rodríguez– C.C: 1010005478

Alejandra Rosero Maya – C.C: 1233194367

MISION TIC 2022 – DESARROLLO WEB

UNIVERSIDAD DE ANTIOQUIA

20 DE OCTUBRE DE 2021

**Objetivo Grupal**

* Crear las diferentes vistas implementando el uso de Express Js, GraphQL y React Js para brindar un adecuado funcionamiento de la página web hacia el usuario por medio de interfaces gráficas.

**Proceso para alcanzar los objetivos y requerimientos del sprint.**

Para poder llevar a cabo la ejecución del Sprint 5 se realizaron diferentes reuniones con el fin de definir qué herramienta para desarrollar el frontend iba a hacer utilizada y cuáles componentes desarrollaría cada miembro.

Con el propósito de garantizar la trazabilidad del proyecto se implementó el uso de GitHub y Trello, pues en Github se puede registrar los cambios del proyecto (como agregación de las diferentes vistas junto con su funcionamiento), mientras que en Trello se especifican las tareas de cada miembro dentro de este sprint:

1. Dentro de Trello, se gestionaron las tareas del sprint 5 para cada miembro, pues se especificaron los procesos para la parte construcción de la parte visual de la app usando tecnologías como GraphQL, Express JS y React Js
2. En cuanto a la actualización del proyecto, cada miembro se encargó de subir su desarrollo al repositorio de github para finalmente hacer un pull request.
3. Para crear las vistas se hizo seguimiento de las diferentes clases impartidas por el docente y la tutora
4. Cada integrante del equipo realizó la instalación de Node Js, necesario para usar Express Js.

De nuevo, para actualizar el repositorio de Git se hizo uso de las ramas iniciales y se modificó el proyecto para indicar el backend y el correspondiente frontend, donde se indica las vistas junto con sus métodos.

**Rutas**

**GitHub:**   <https://github.com/AlejandraRo/ciclo4-proyectofinal>

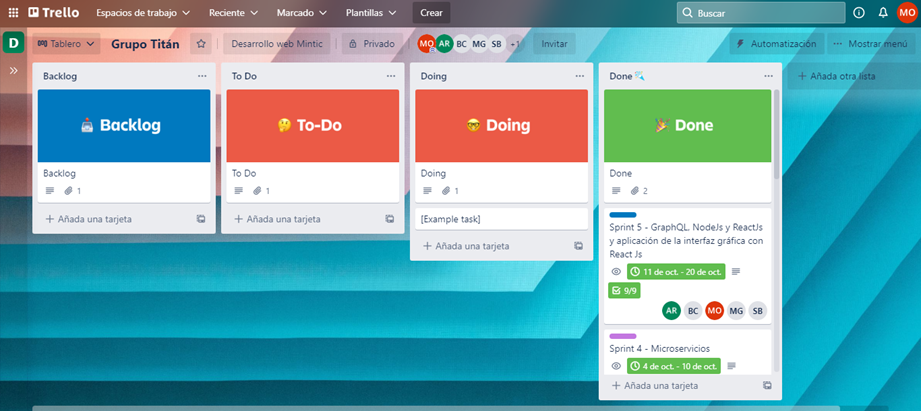
(Dentro de la rama Development. Frontend: carpeta frontend, Backend: Carpeta javascript)

**Trello:** <https://trello.com/b/21Ga3KWA/grupo-tit%C3%A1n>

**MongoDB:** <https://cloud.mongodb.com/v2/613e16830a992558f04f17cc#access>

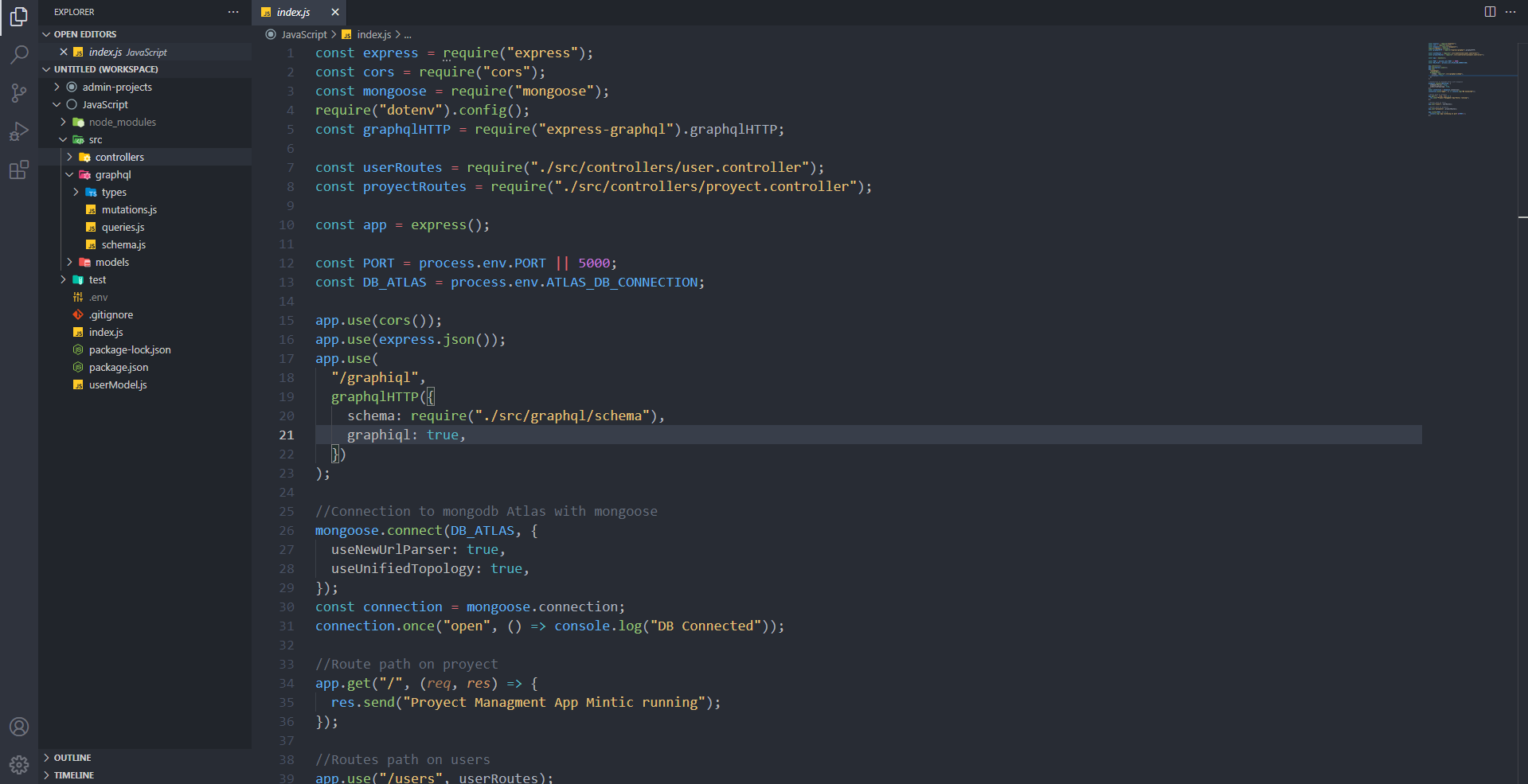
**Azure:** <https://dev.azure.com/ssgonzalezb/manage_proyects/_build>

**EVIDENCIAS**

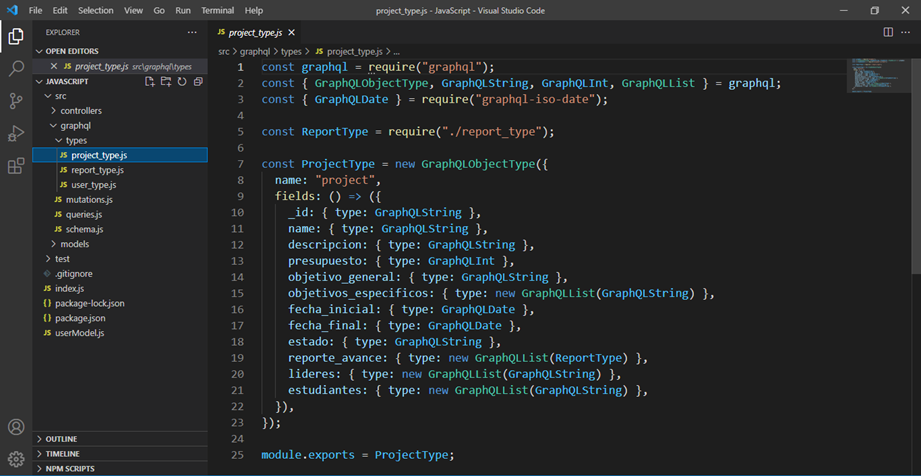


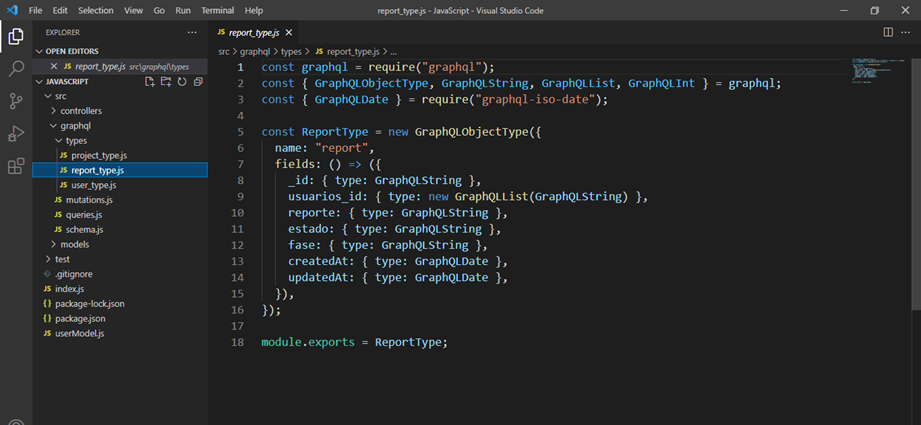
**BACKEND**

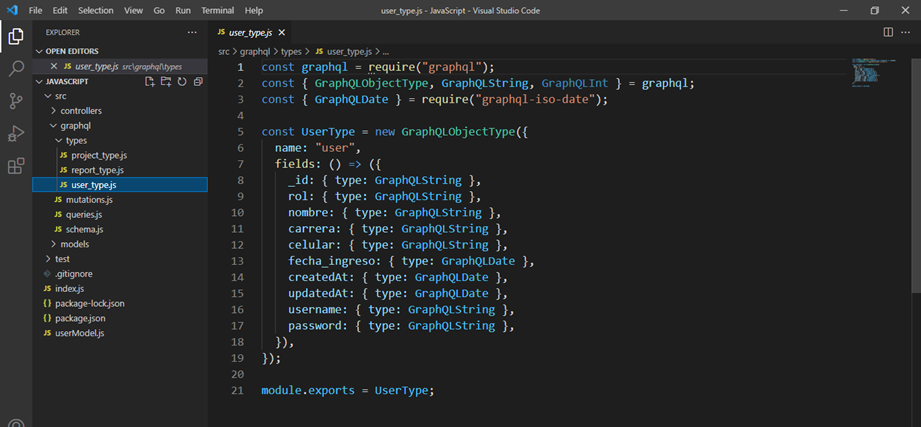
**PROYECTO DE EXPRESS ACTUALIZADO CON LAS DEPENDENCIAS DE GRAPHQL**

****

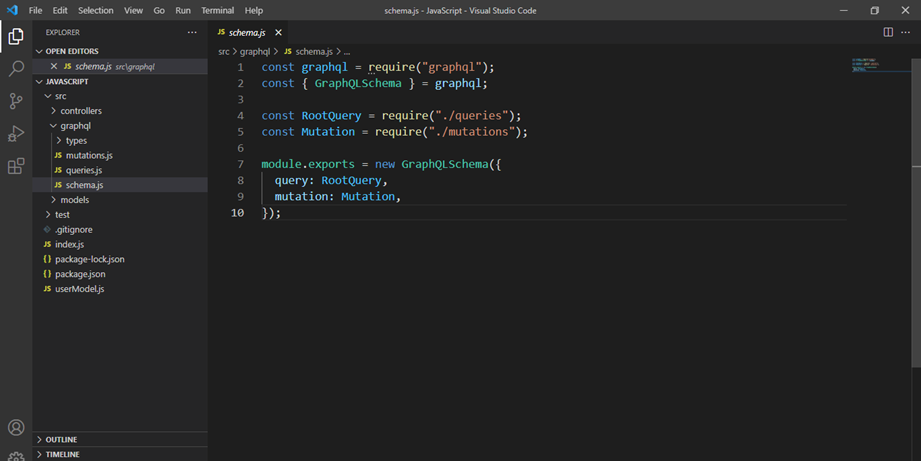
**DEFINIENDO TIPOS DE PROYECTOS Y USUARIOS EN GRAPGQL**

****

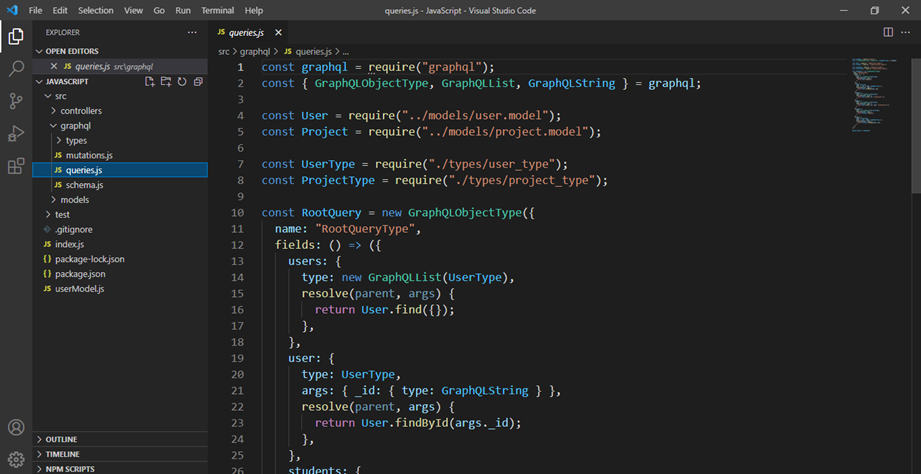
****

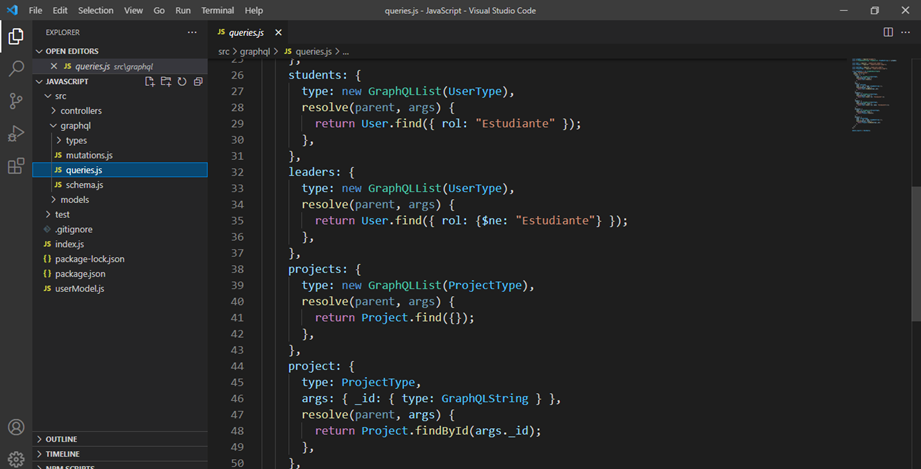
****

**DEFINIENDO SCHEMA GRAPHQL**

****

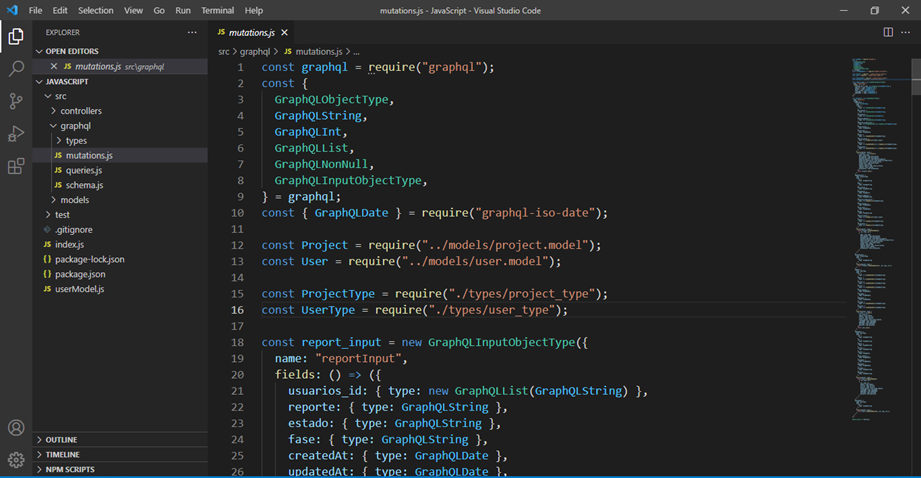
**DEFINIENDO QUERIES PARA PROYECTOS Y USUARIOS**

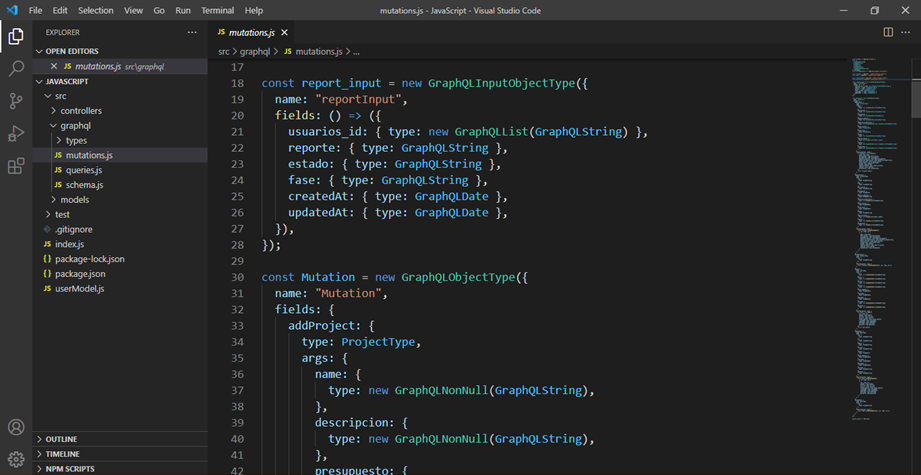
****

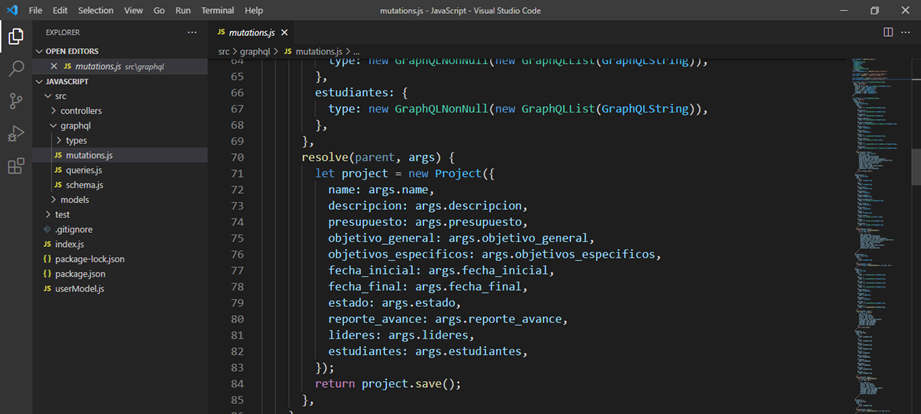
****

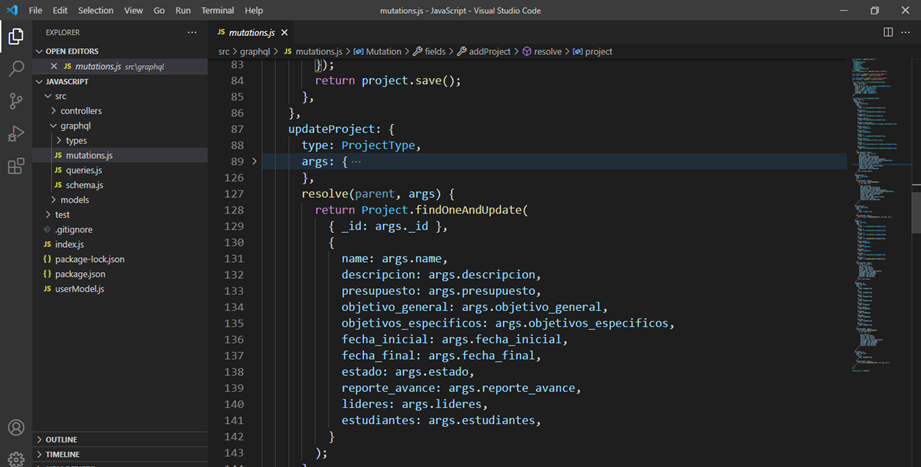
**DEFINIENDO MUTATIONS PARA PROYECTOS Y USUARIOS**

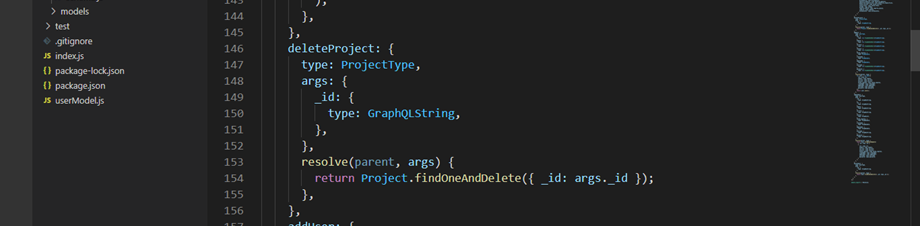
Mutaciones de proyectos: Crear, actualizar y eliminar.

****

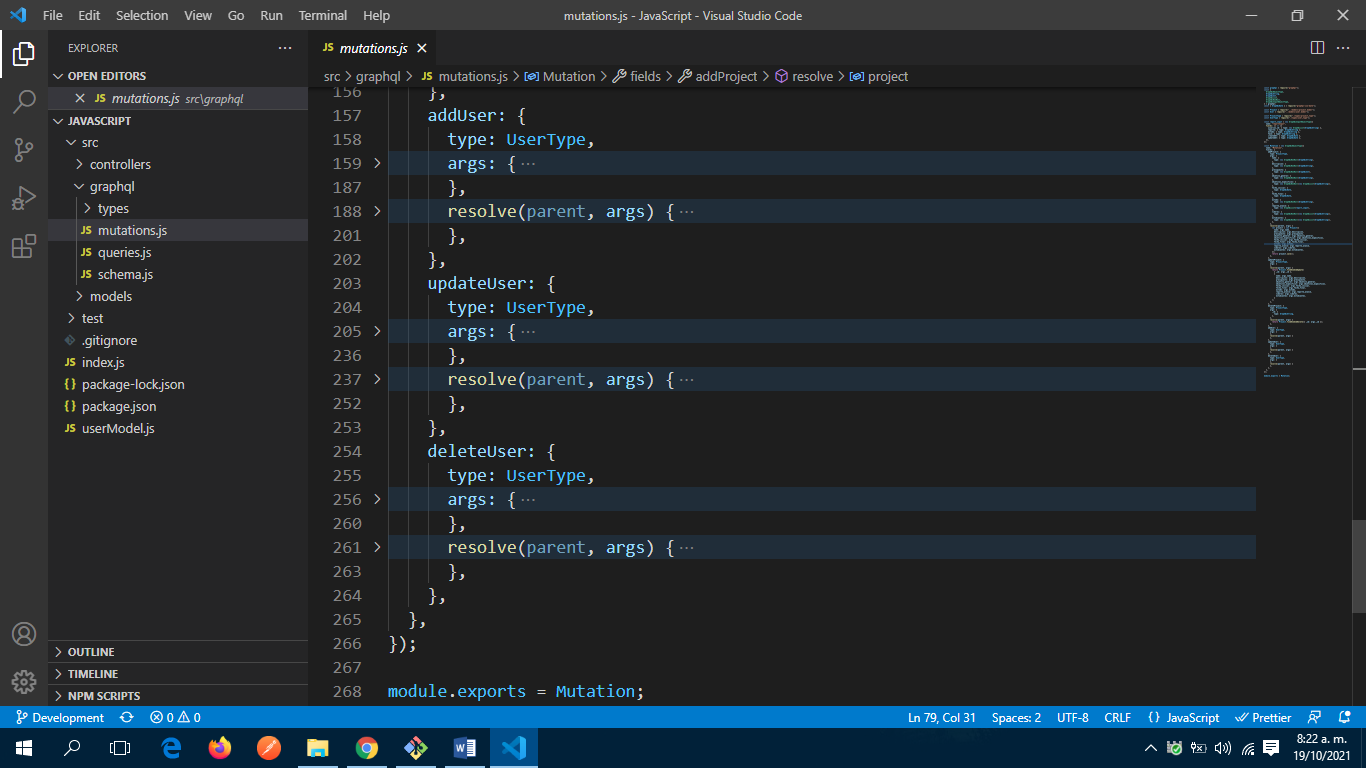
****

****

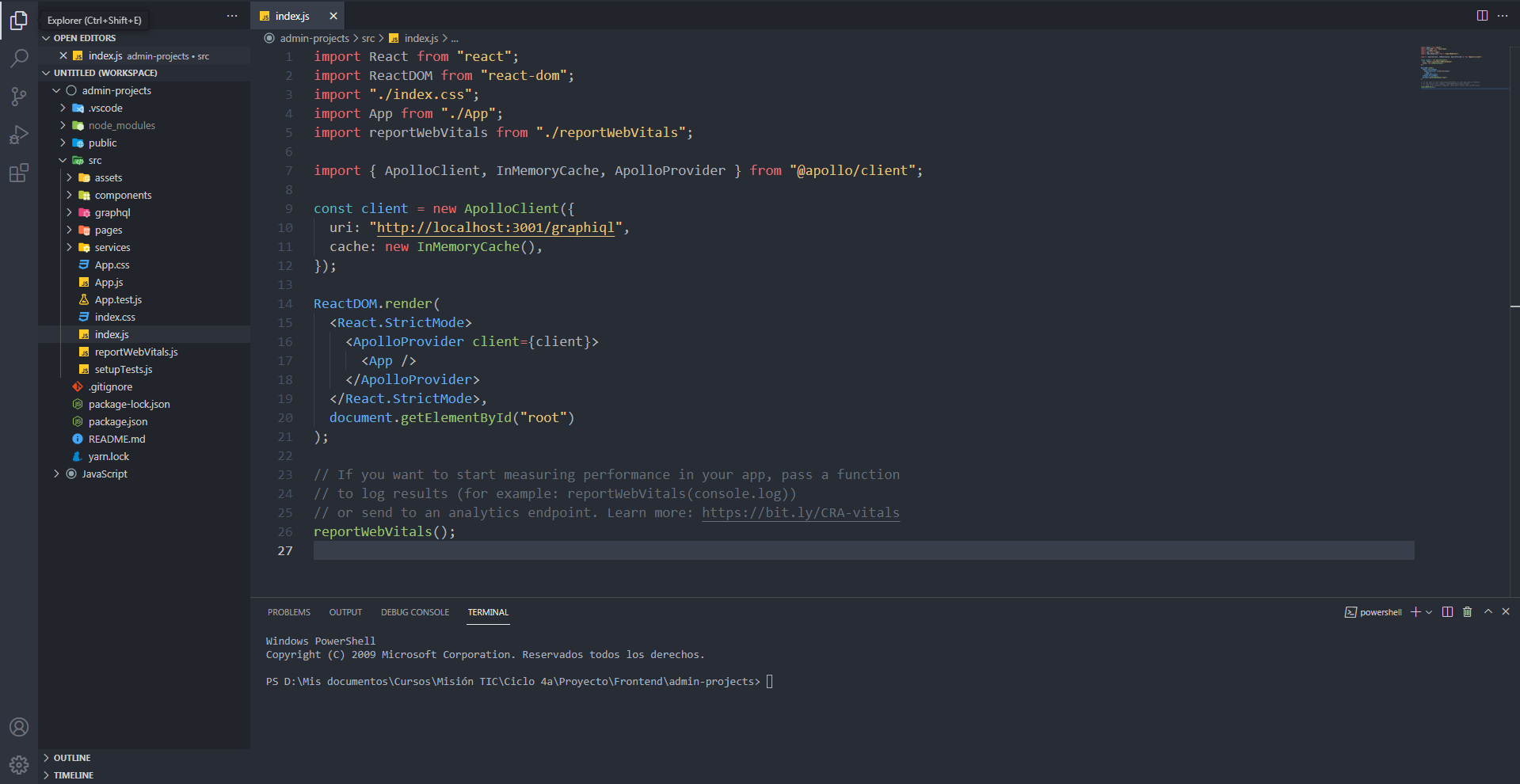
****

****

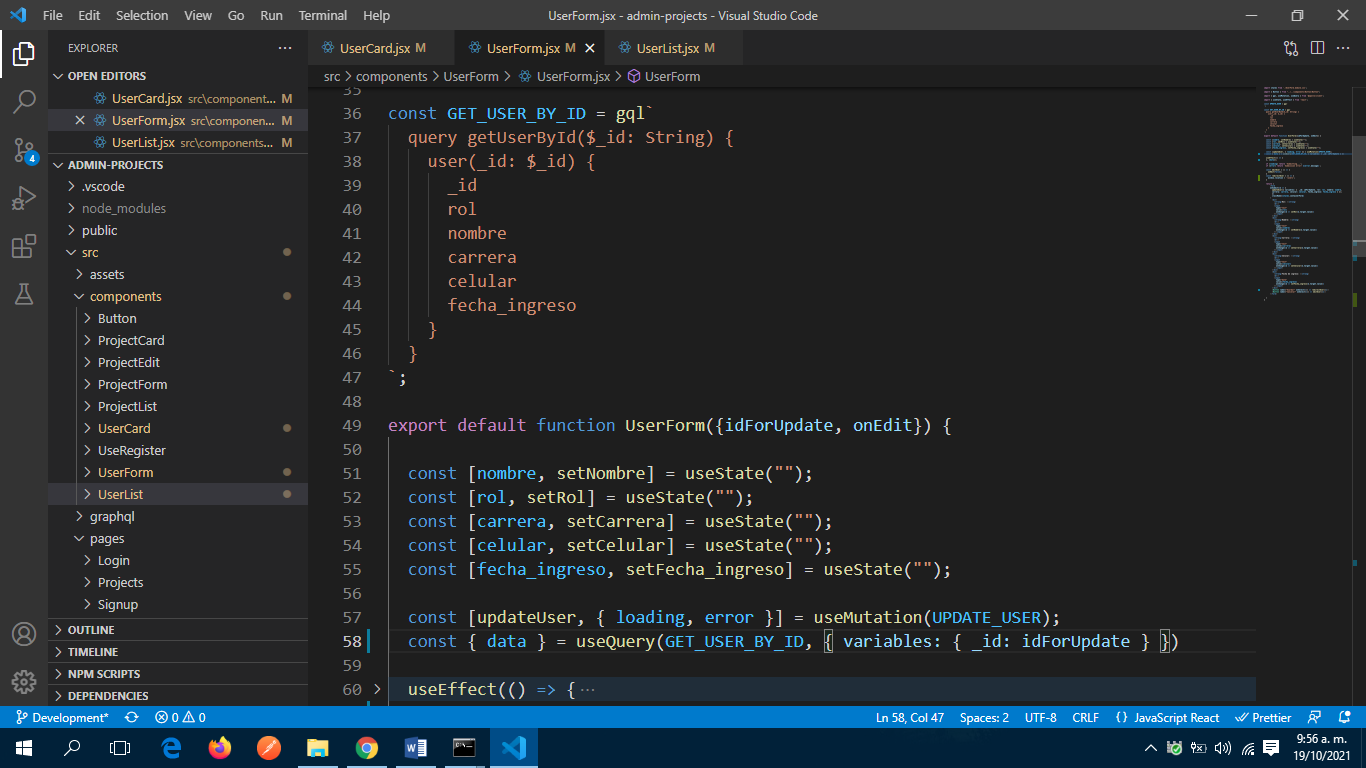
Mutaciones de usuarios: Crear, actualizar y eliminar.



**PROYECTO INICIALIZADO USANDO CREATE-REACT-APP**

****

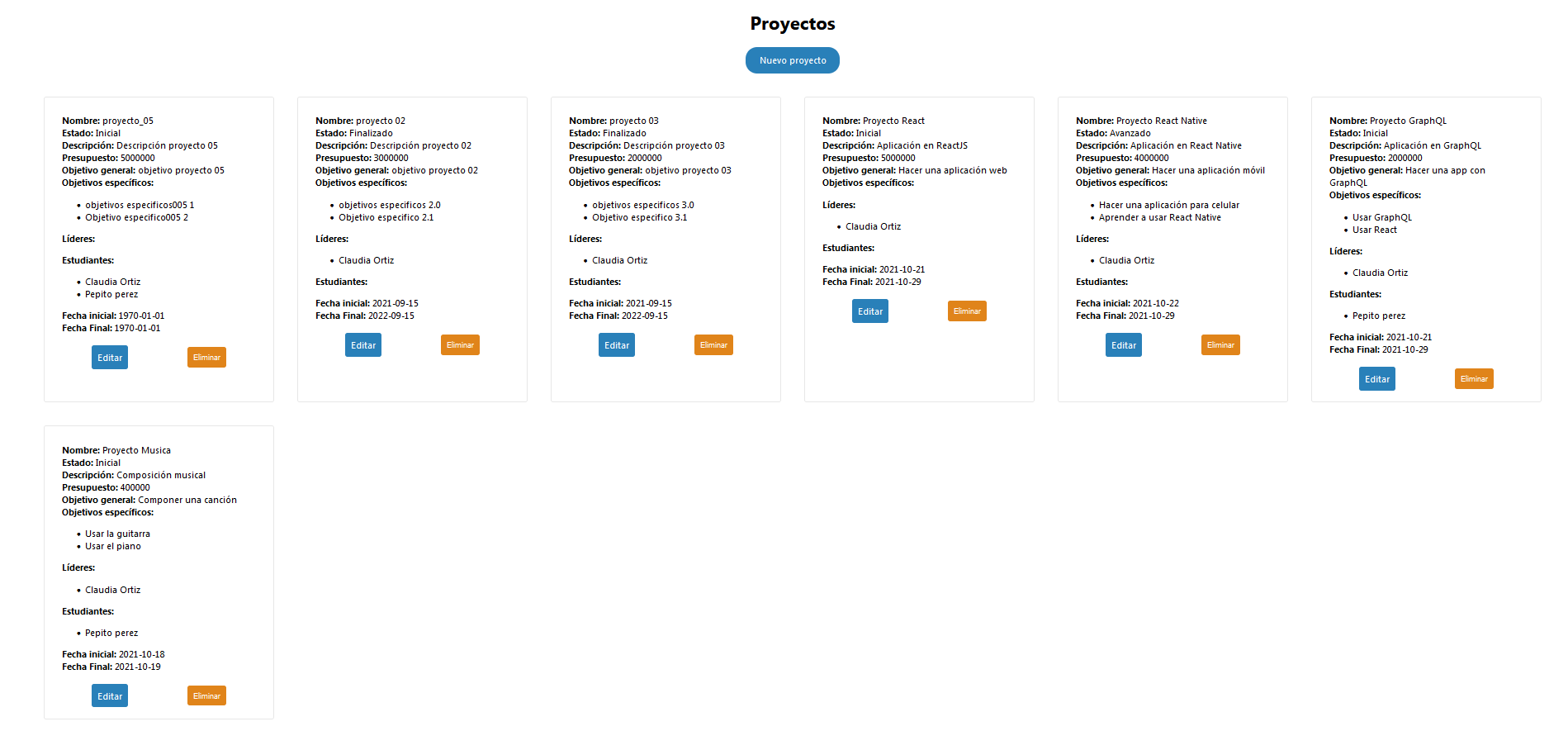
Tanto para proyectos como usuarios definimos los query y mutations que queremos ejecutar envolviéndolas en el template literal ‘gql’, y usamos useQuery y useMutation que permiten compartir datos de GraphQL con la interfaz.



**DESARROLLO DE LAS HISTORIAS DE USUARIO**

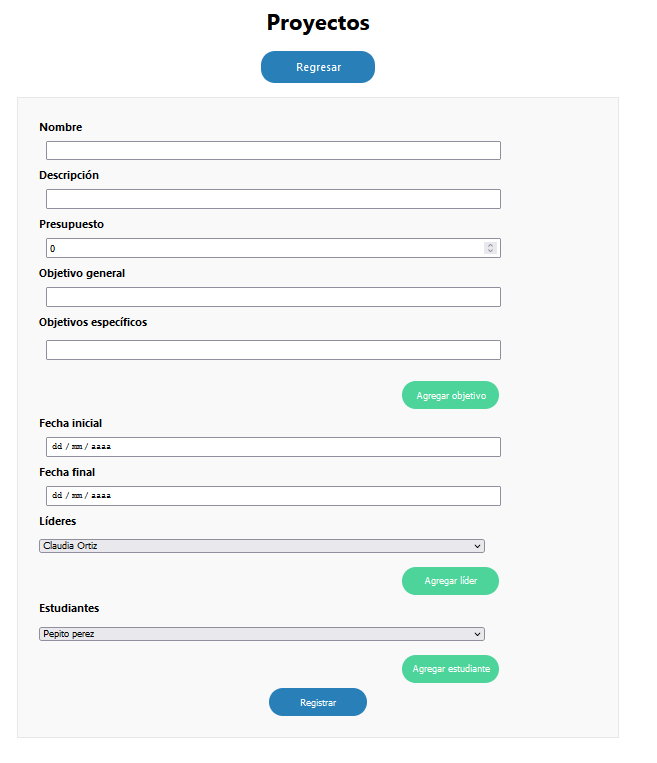
**MÓDULO DE PROYECTOS**

-Visualización de la información de los proyectos almacenados en la base de datos

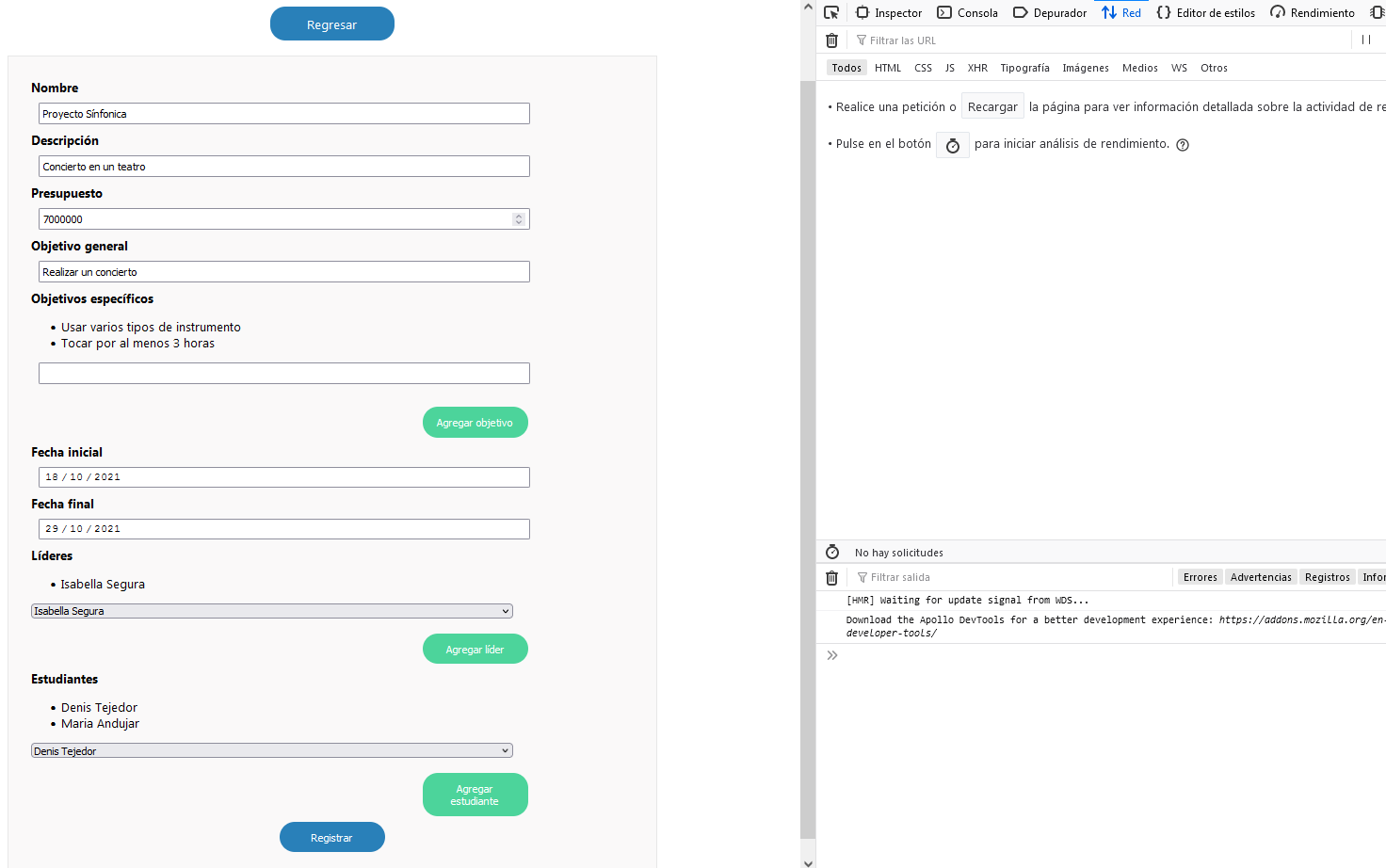


-Almacenamiento la información del proyecto en base de datos

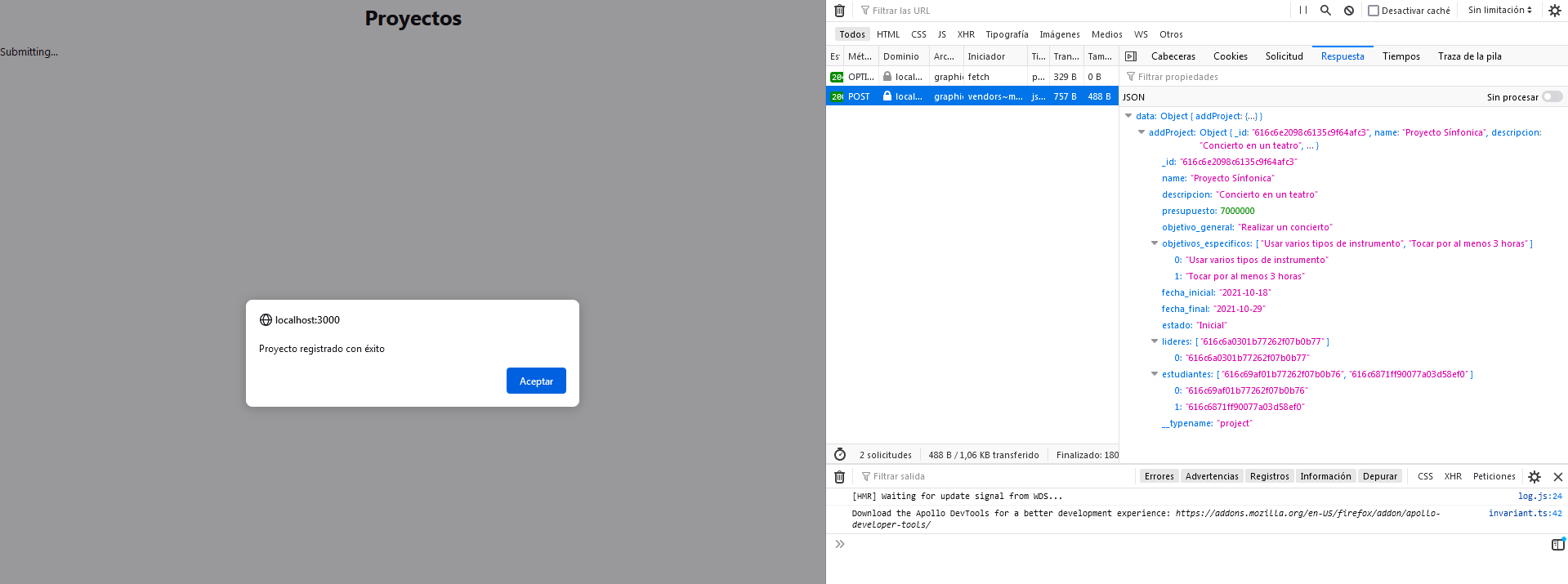
Formulario de registro de proyectos



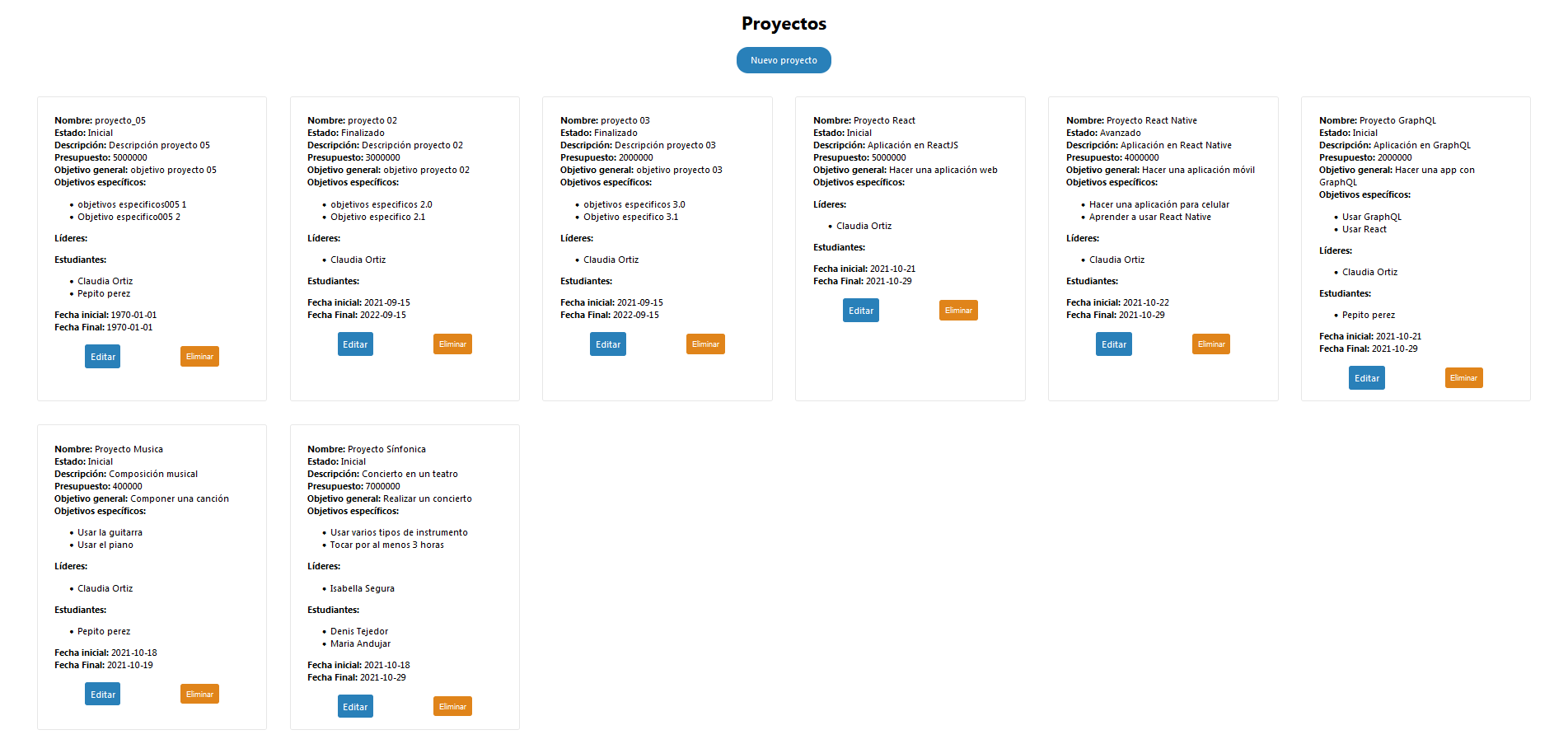
Campos llenados con la información del nuevo proyecto, momento previo al registro no se ve tráfico en la pestaña de red del navegador



Una vez se le da a la opción de registrar aparece una alerta de registro exitoso y en red se puede ver la solicitud hecha través de GraphQL

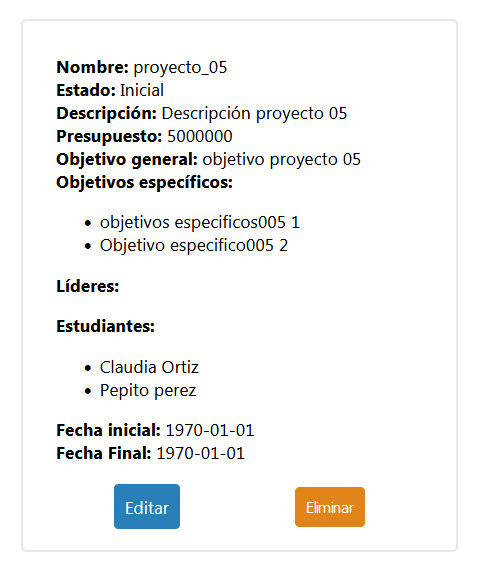


Al revisar nuevamente todos los proyectos se puede apreciar el nuevo proyecto agregado

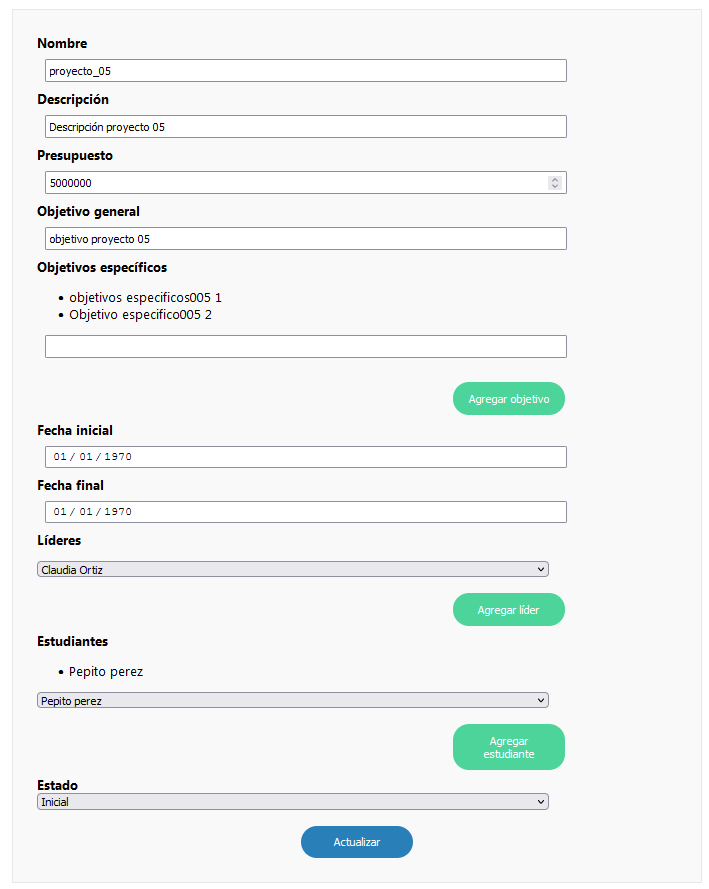


-Actualizar la información de un proyecto

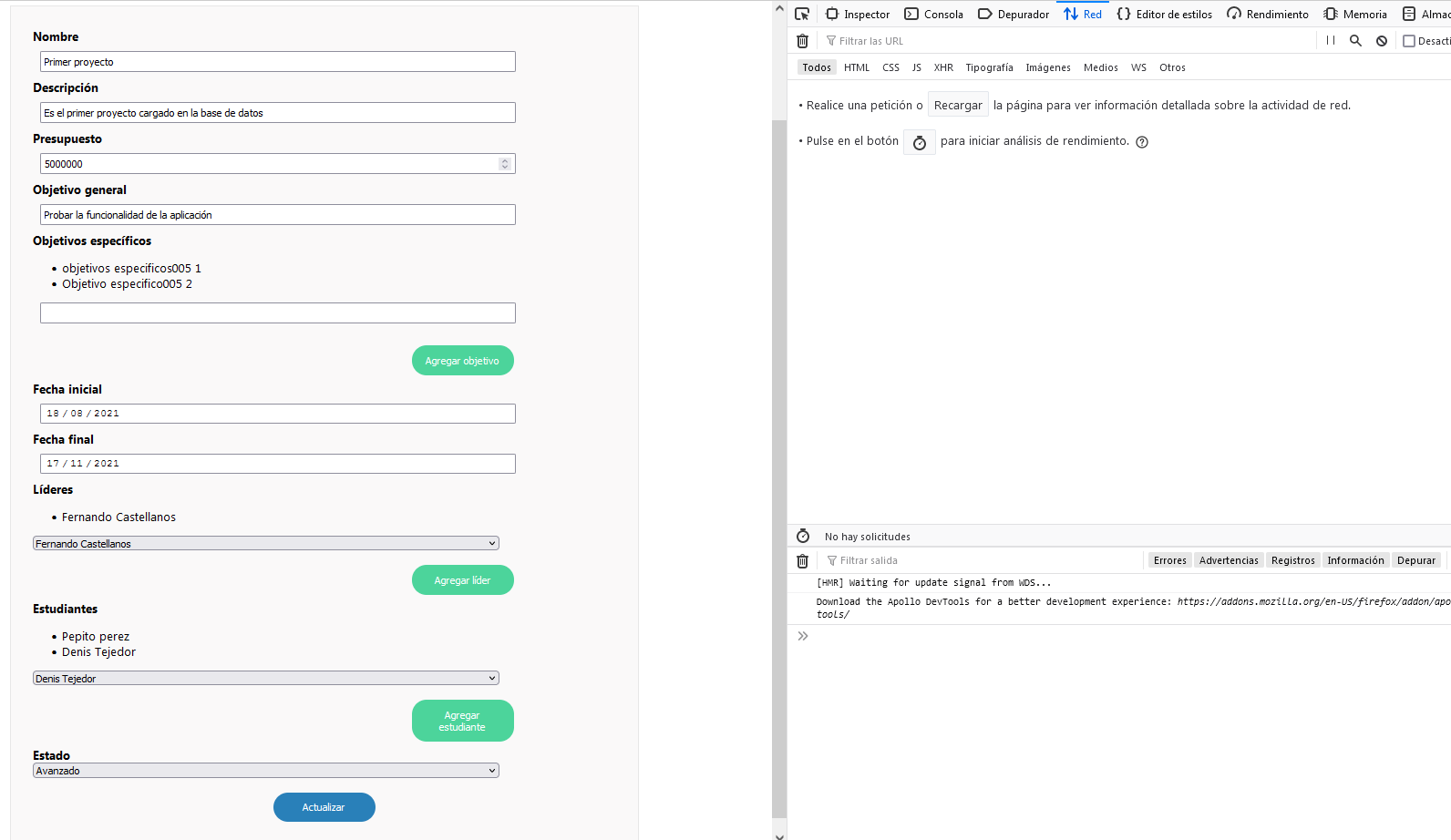
Proyecto que será actualizado:



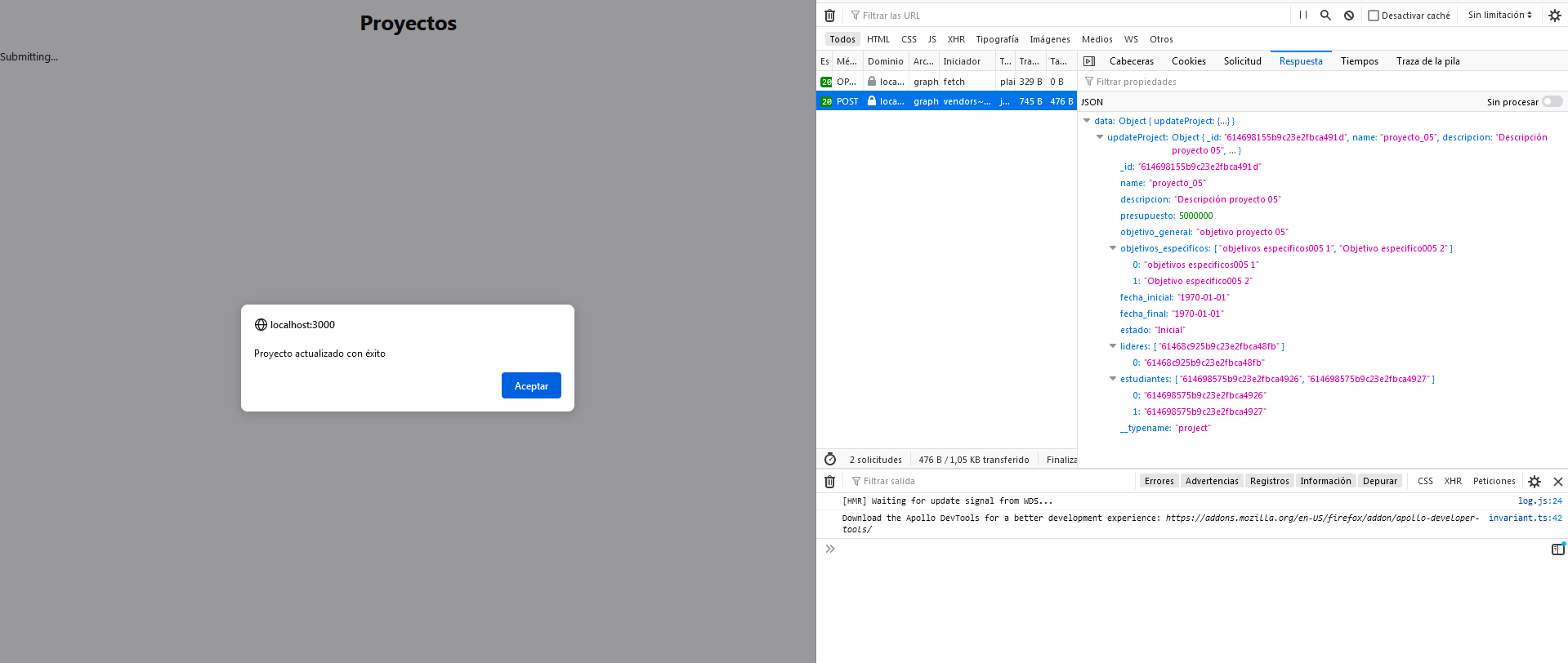
Formulario de edición:



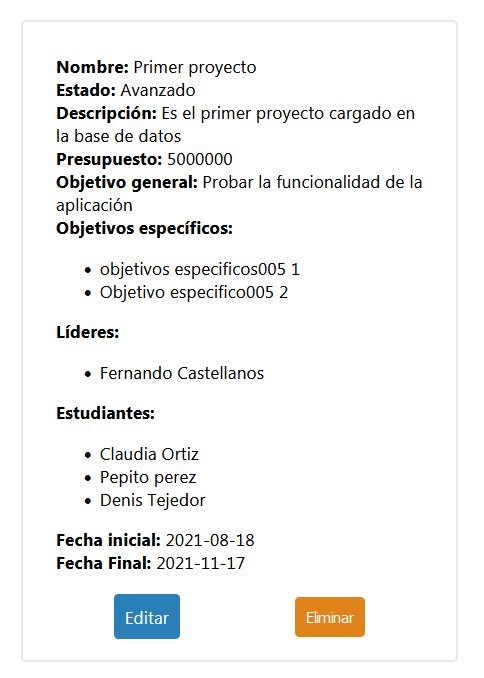
Momento previo a la actualización. Los campos están modificados y no se ve tráfico en la pestaña de red del navegador



Al seleccionar la opción de actualizar aparece una alerta y se ve la solicitud hecha con GraphQL



Nuevo estado del proyecto:

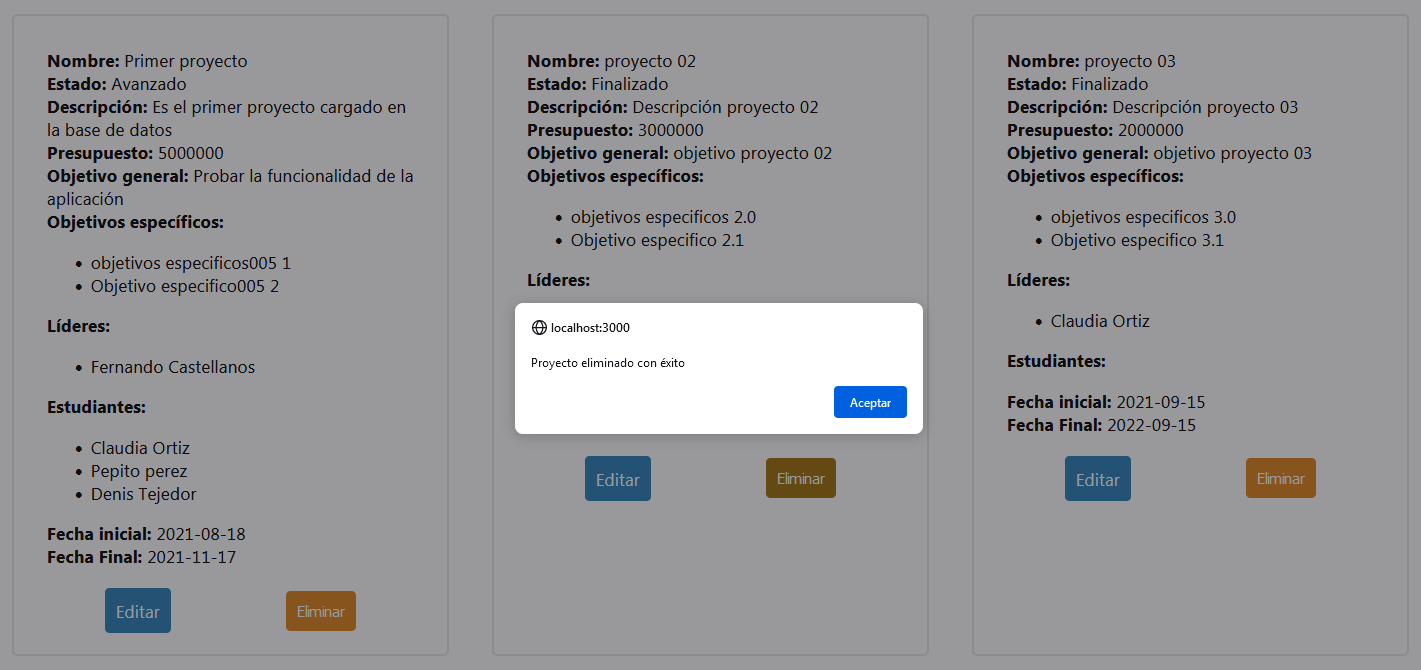


-Borrar proyectos

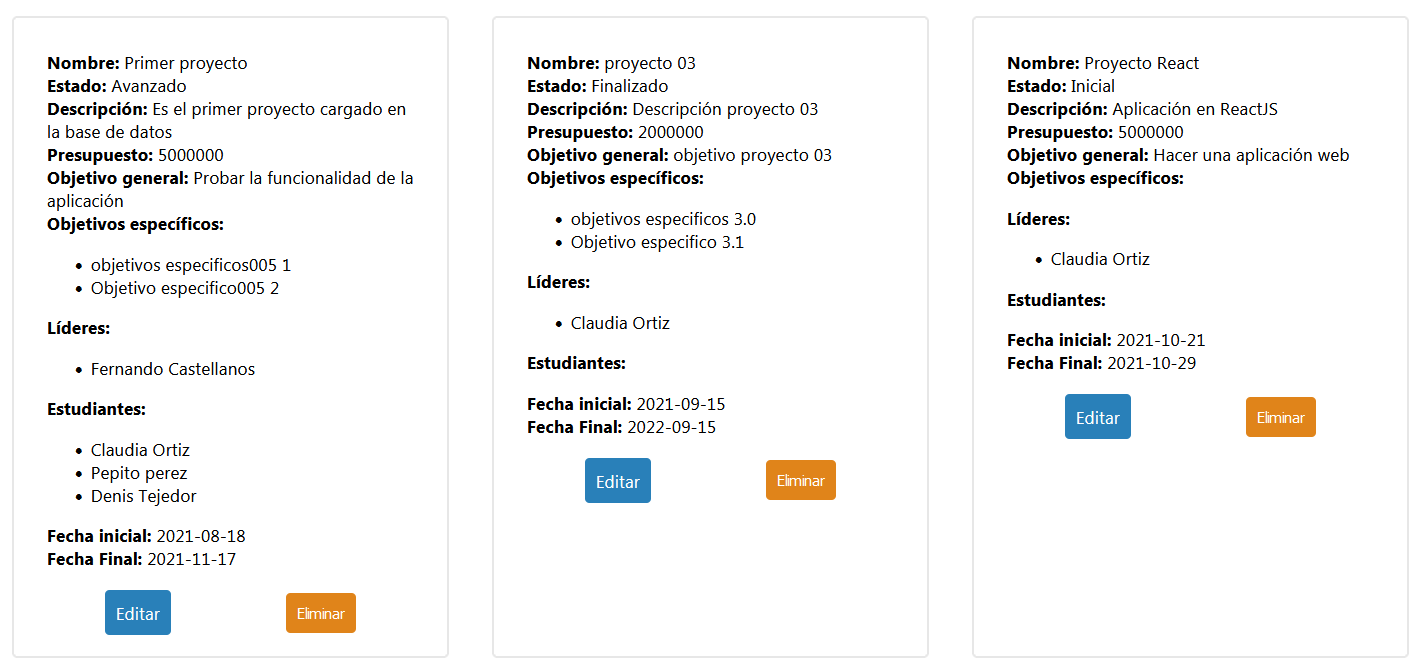
Vista de proyectos antes de la eliminación de uno de ellos:



Proceso de eliminación:



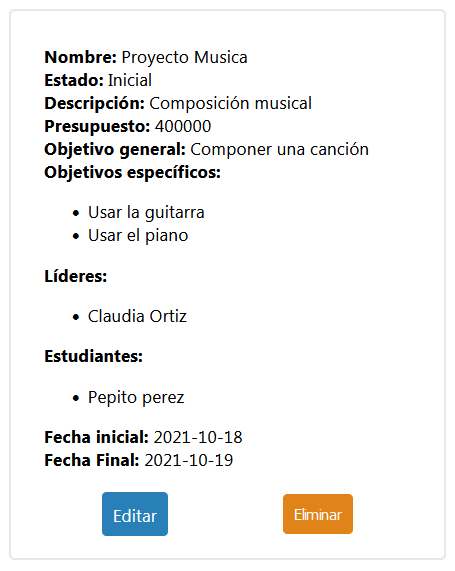
Vista de proyectos después de la eliminación. Ya no está el proyecto 02:



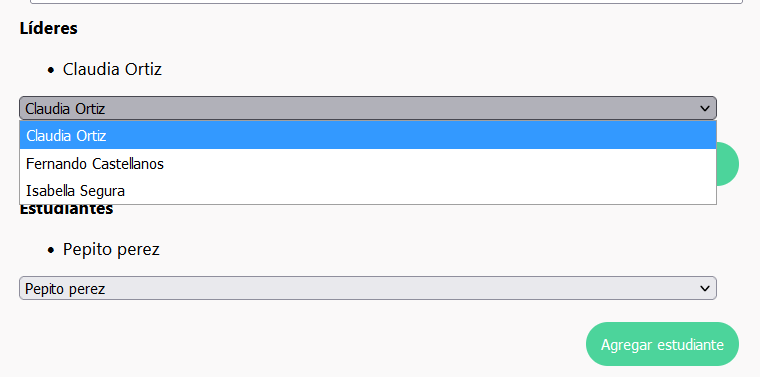
-Adicionar integrantes a los proyectos de investigación.

La adición se puede hacer desde el formulario de nuevo proyecto o desde el de edición de proyecto.

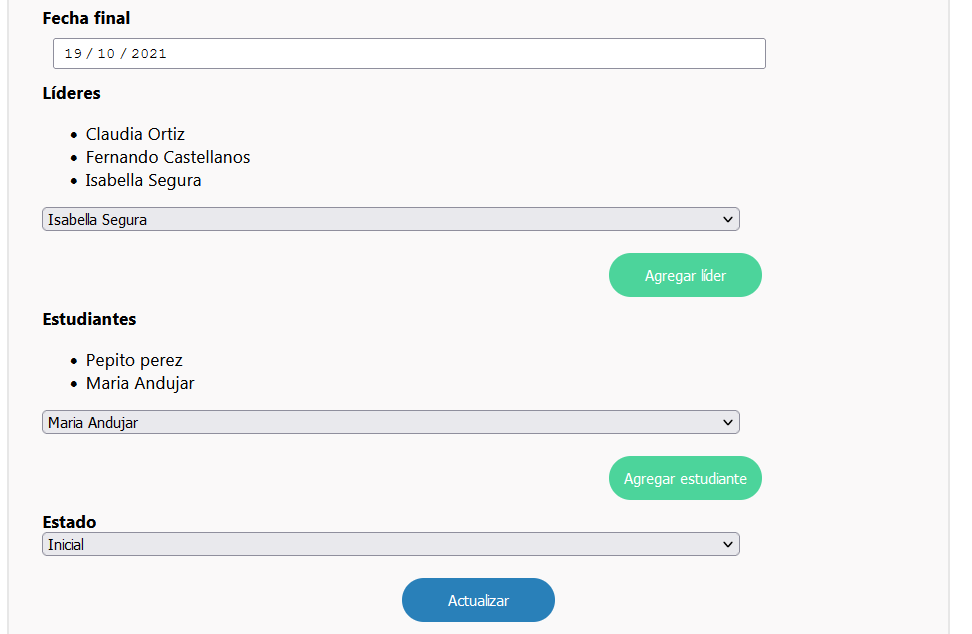
Proyecto al que se le adicionarán integrantes:



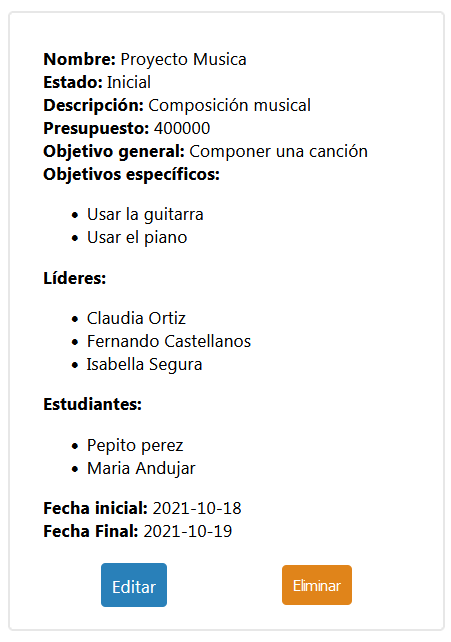
Proceso de adición de integrantes:



Integrantes agregados desde el formulario:



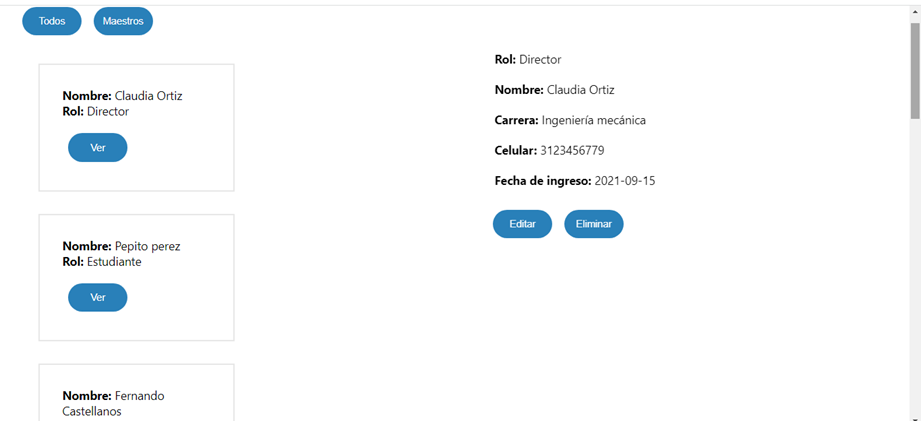
Nuevo estado del proyecto después de guardar los cambios:

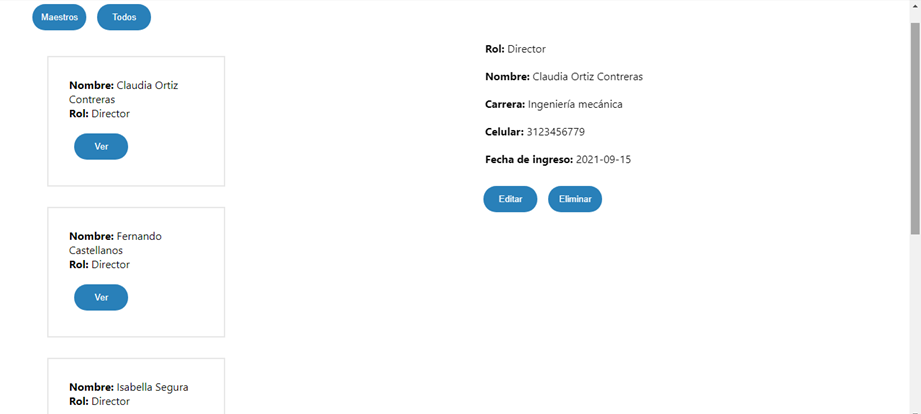


**MÓDULO DE USUARIOS**

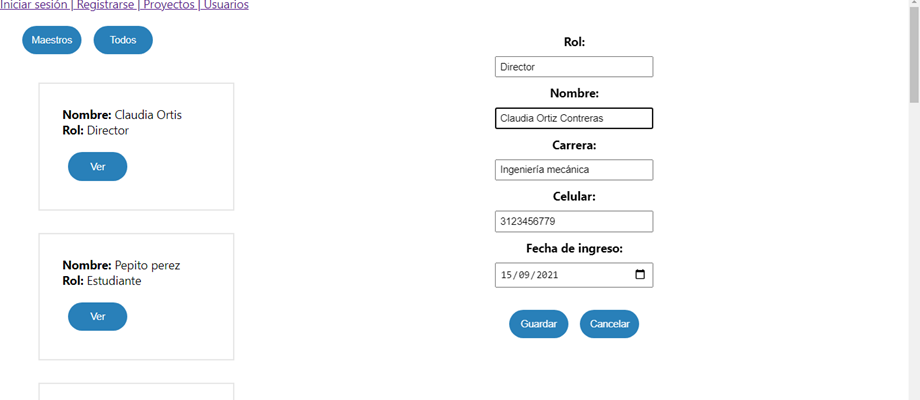
-Visualización de la información de los usuarios almacenados en la base de datos.

El botón todos obtiene todos los usuarios almacenados en la base de datos y el botón maestros, obtiene los usuarios con rol “Director”. El botón ver muestra la información específica de cada usuario.

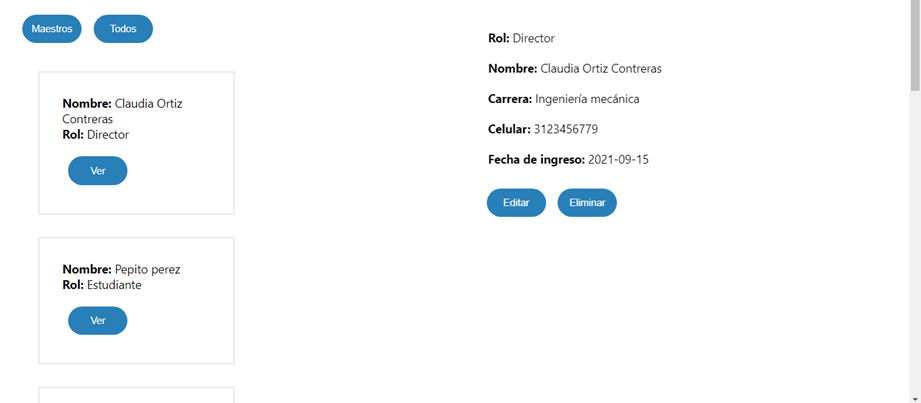




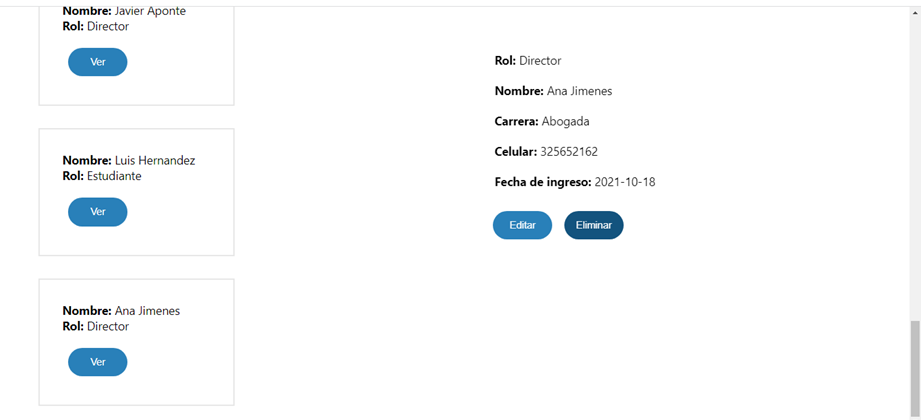
El botón editar activa el formulario con los datos por defecto de usuario que se quiere editar



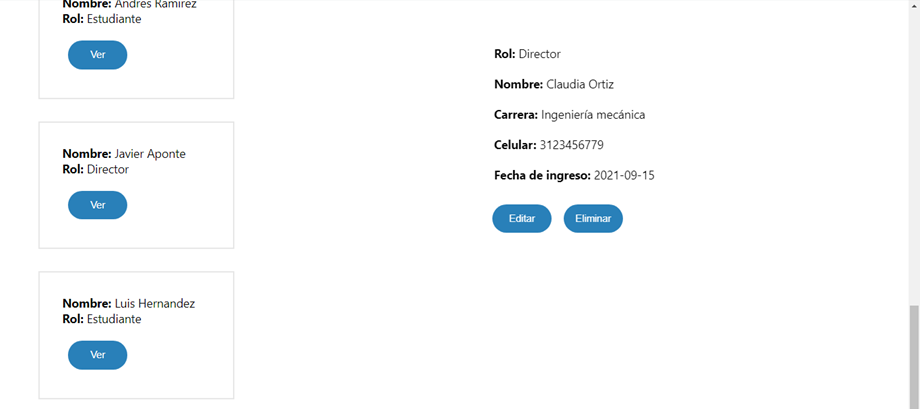
Evidencia de la edición:



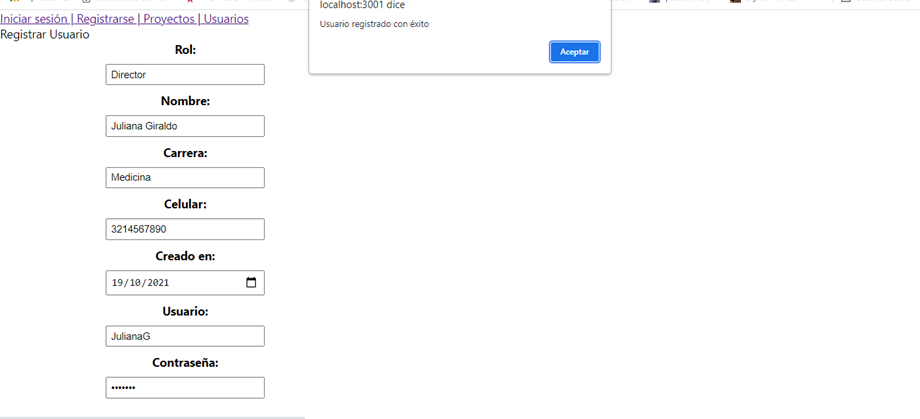
El botón eliminar, elimina el usuario por sus id



Evidencia usuario eliminado:



Para crear un nuevo usuario dentro del link de Registrarse aparece el formulario con los datos solicitados para su creación:



Evidencia del registro en usuarios:

