**Sprint 5: GraphQL, NodeJs y ReactJs y aplicación de la interfaz gráfica con React Js**

Uberney Ramírez Restrepo

Jaime Andrés Sánchez Rodríguez

Diana Villamil Collazos

Ministerio de las TIC

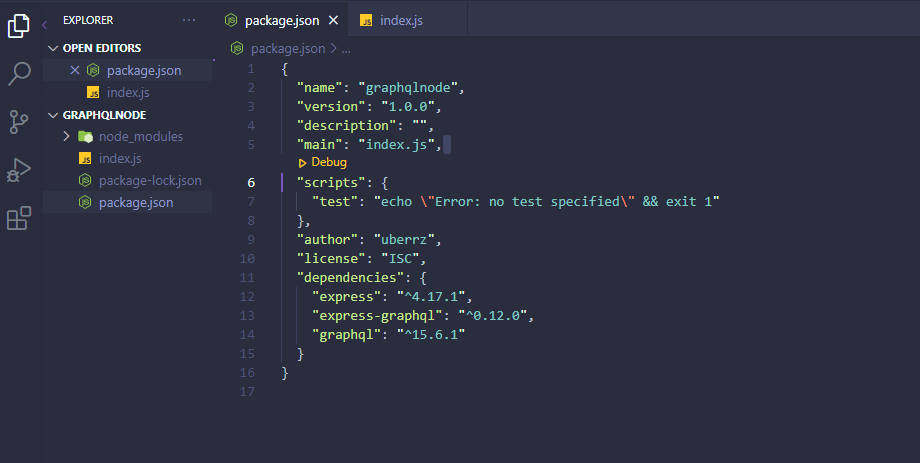
Universidad de Antioquia

Ciclo 4A Programación Web Grupo 2

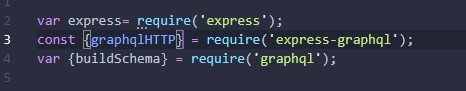
17 de octubre de 2021

Para esta última entrega se presenta la evidencia de los diferentes requisitos solicitados, entre los cuales se destacan; la creación, actualización y consultas en bases de datos (mongoDb) mediante el uso de Schemas, mostrar la creación y posterior utilización del servidor NodeJs con la ayuda del framework Express para poder acceder y desarrollar las diferentes consultas a través de los Schemas que posteriormente permiten interactuar con la base de datos.

En esta primera parte se crea el archivo index.js en donde se organizan las especificaciones para trabajar con graphQl.



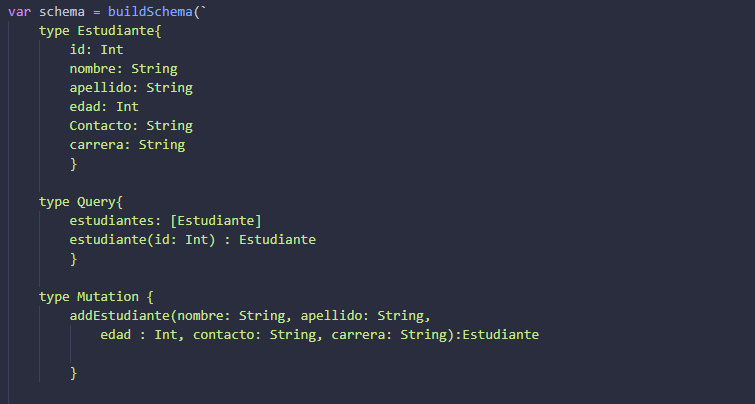
De la misma manera se procede a la creación de los Schemas.



En la siguiente imagen se muestra la construcción del Schema, en donde se definen los paramatros que solicitará cada documento de la colección de estudintes y el tipo de variable al que corresponde, en este caso la mayorí corresponde a tipo String, a excepcion de la edad que es de tipo int.

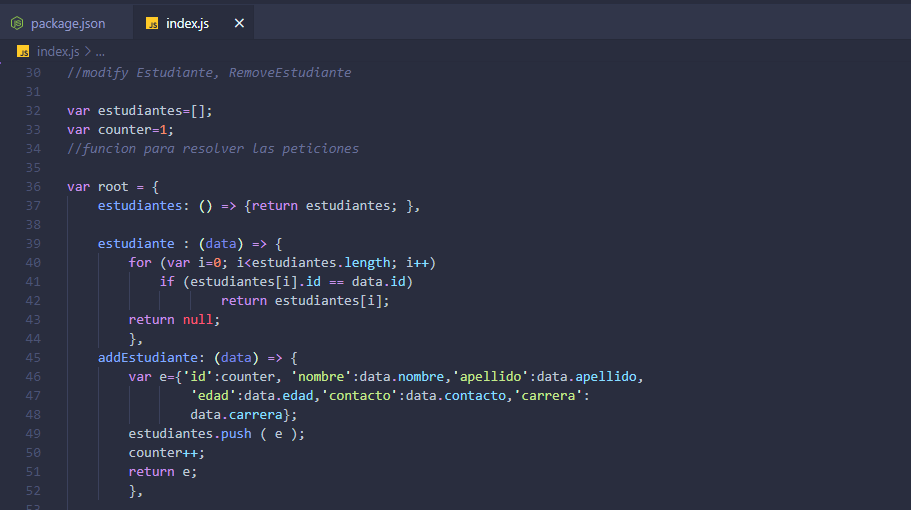
Por otro parte se muestran los querys de consulta, el primero de ellos, “estudiantes” será el que mostrará un arrray de TODOS los documentos que pertenezcan a dicha colección, y por otro lado el array “estudiante(id:int)” mediante el cual se plantea hacer la consulta especifica de un estudiante mediante le uso de su respectivo id.

Finalemente, en esta imagen se encuentra al final de ella la mutación que nos permitirá adicionar datos a la colección de estudiantes.

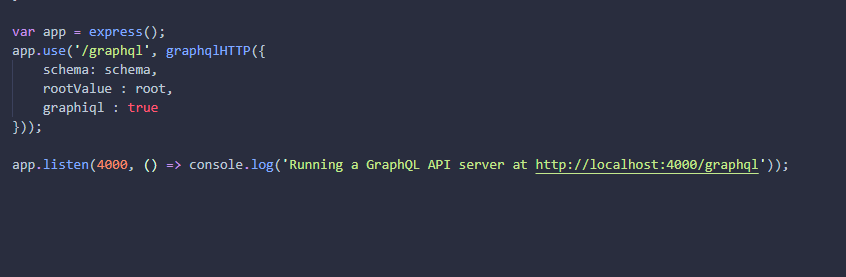


En la siguiente imagen se muestra de manera mas detallada el como se van a adicionar los estudiantes a la colección. Primero que todo se creó la variable de estudiantes que en principio va a estar vacia y posteriormenete se le van agregando sus documentos, ademas se creó la variable counter mediante la cual se irá creando de manera automatica el id de cada estudiante y mediante el cual se podrá hacer una consulta mas adelante.

Cerca del final del codigo se observa un estudiantes.push(e); mediante el cual se hace la respectiva adición a la colección y procede posteriormente a aumentar el counter y dejarlo listo para el proximo estudiante a agregar.

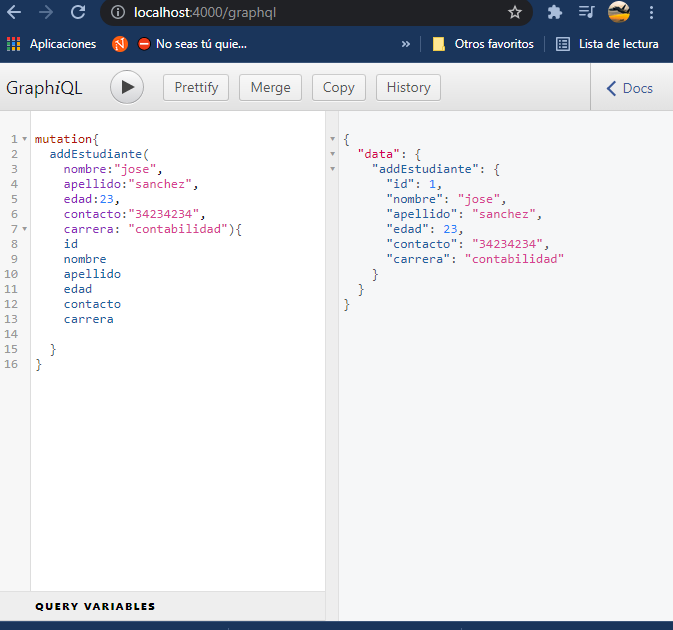


En la siguiente parte se observa el como se utiliza el framework Express para poder interacutar con la base de datos, ademas se ve como hacemos un “llamado” para el localhost:4000 en nuestro caso.

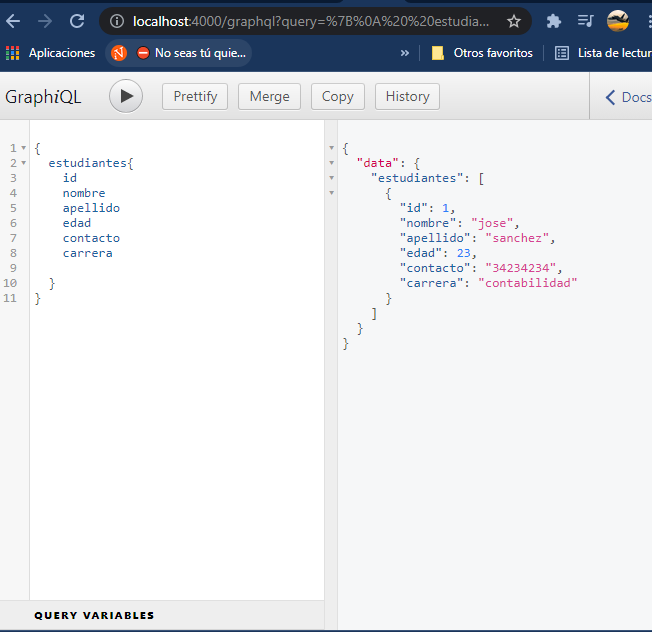


A continuación se corre en localhost ubicado en el puerto 4000 y se procede a agregar a un estudiante.

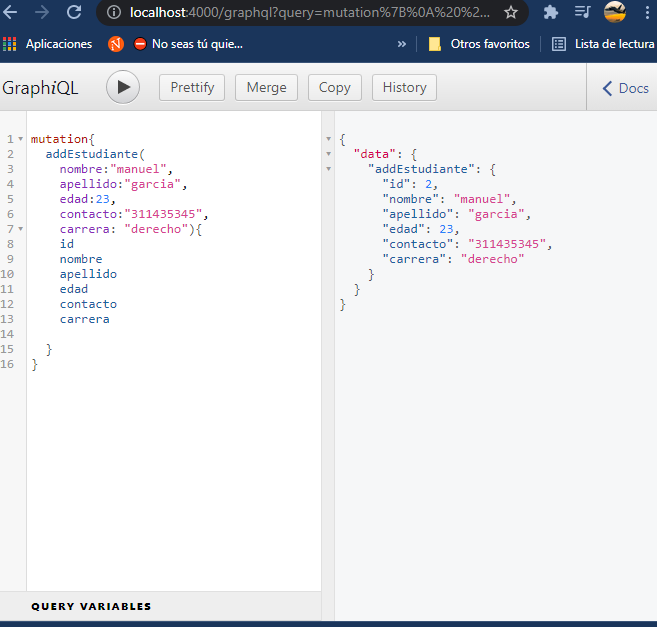
En la parte izquierda de la pantalla se observa cómo se crea el estudiante mediante una mutación, allí se registran los diferentes datos a agregar, en la parte derecha se muestra en formato json el dato como ha sido agregado a la colección.



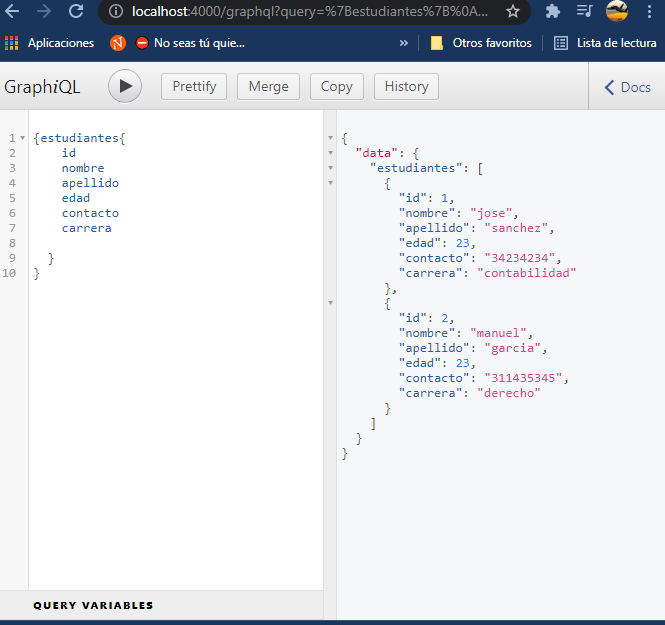
En la imagen que se muestra a continuación se desarrolla la consulta del listado de estudiantes, como hasta este momento solo había un estudiante en la colección, tan solo se muestra este registro, sin embargo al ir adicionando más estudiantes, se mostraran todos ellos en la parte derecha de la pantalla y usando el mismo comando que aparece en la parte izquierda, tal como se mostrará mas adelante.



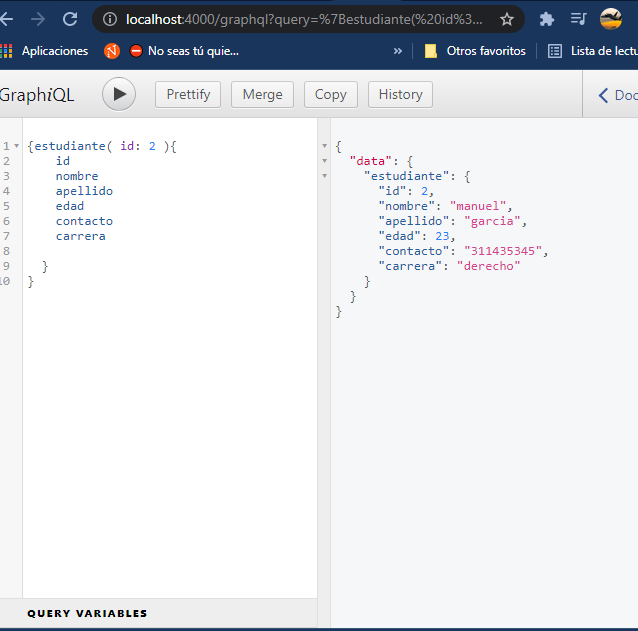
En la siguiente imagen se muestra como se agrega otro estudiante a la colección de estudiantes y como de manera automatizada el id va cambiando de manera incremental.



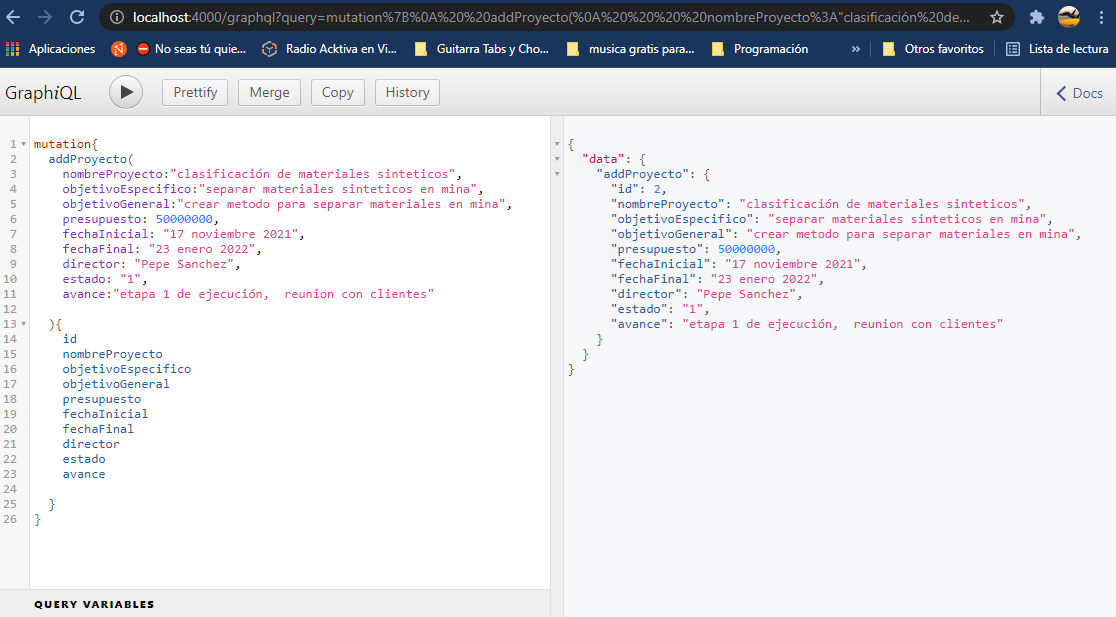
De esta manera y como se había previsto de manera anticipada, utilizando el mismo query de consulta que está a la izquierda, se muestran los dos estudiantes que actualmente harían parte de la colección de nombre estudiantes.



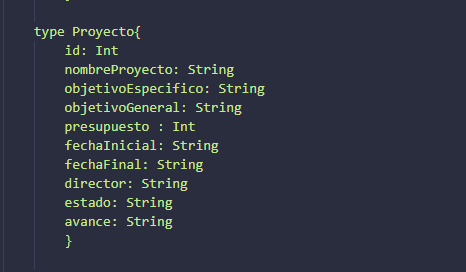
Ahora que ya existen varios documentos (estudiantes) dentro de nuestra colección, se hace necesario el desarrollar una consulta especifica de solo alguno de dichos documentos, para ello se desarrolla el query de la parte izquierda en la que se consulta un estudiante por id( para este caso el estudiante con id=2), y a la derecha se muestra la información solicitada para este estudiante.



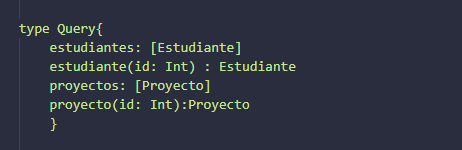
De la misma manera como crea un documento para la colección de estudiantes, se muestra a continuación como se adicionan documentos a la colección de proyectos.



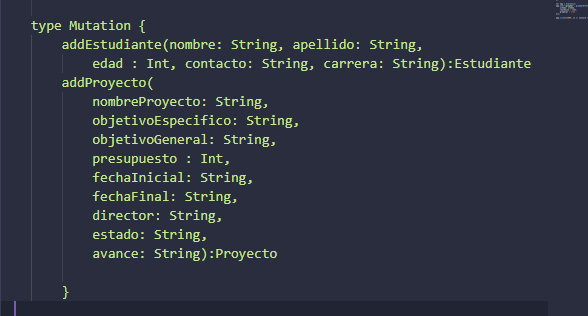
Para ello se requirió previamente de crear el type Proyecto con sus respectivos parámetros como se muestra a continuación:



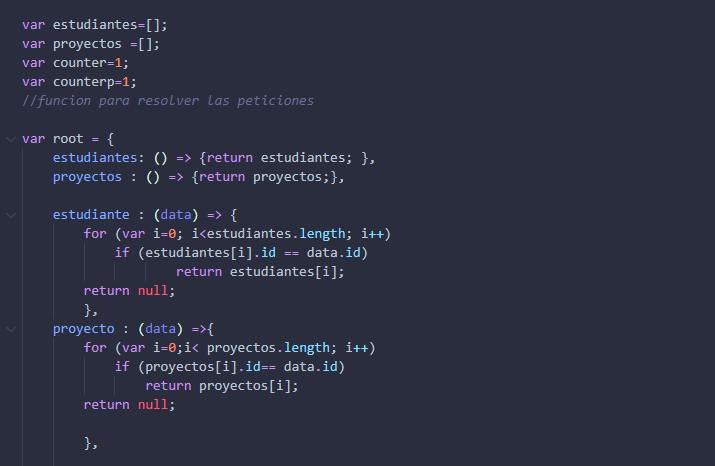
De igual manera para los proyectos se debe desarrollar su respectivo query de consulta ya sea para todos los proyectos o tan solo para alguno de ellos con su respectivo id como se muestra a continuación:



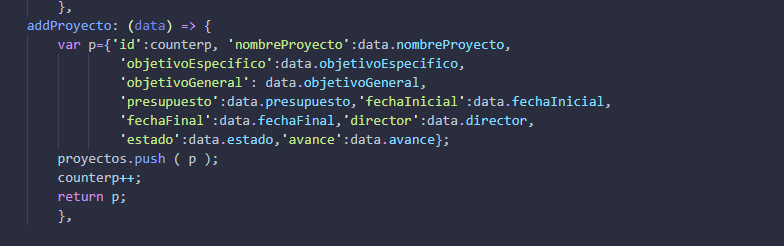
Para poder agregar nuevos proyectos a nuestra colección se crea la respectiva mutación para poder interactuar con el graphQl, tal como se ve en la siguiente imagen:



Se adiciona la forma de crear la variable proyectos:

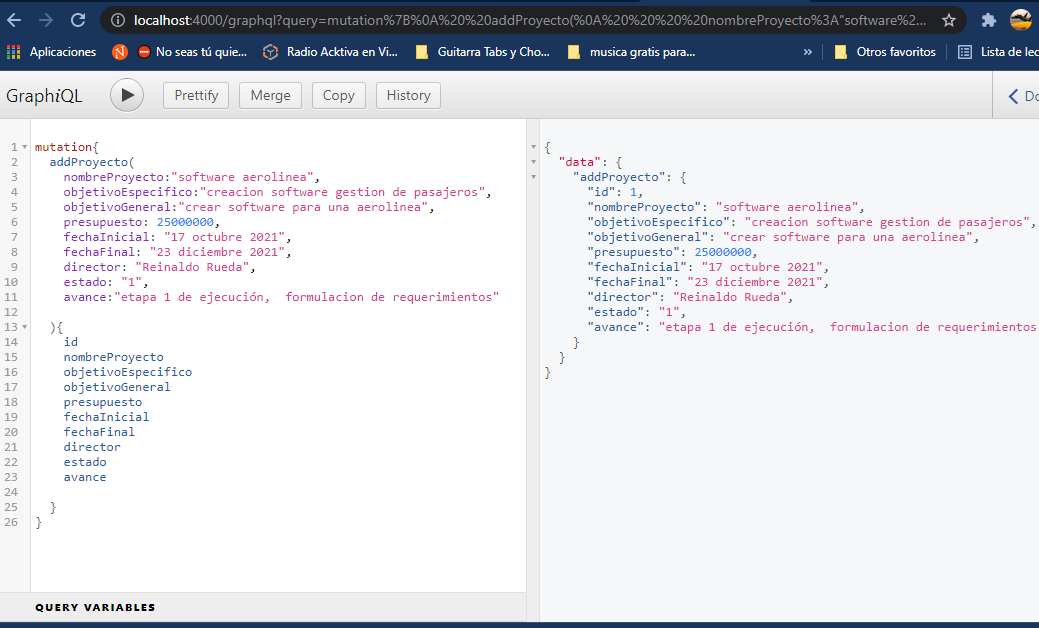


Y de esta manera se muestra como se creó la función para crear los proyectos:

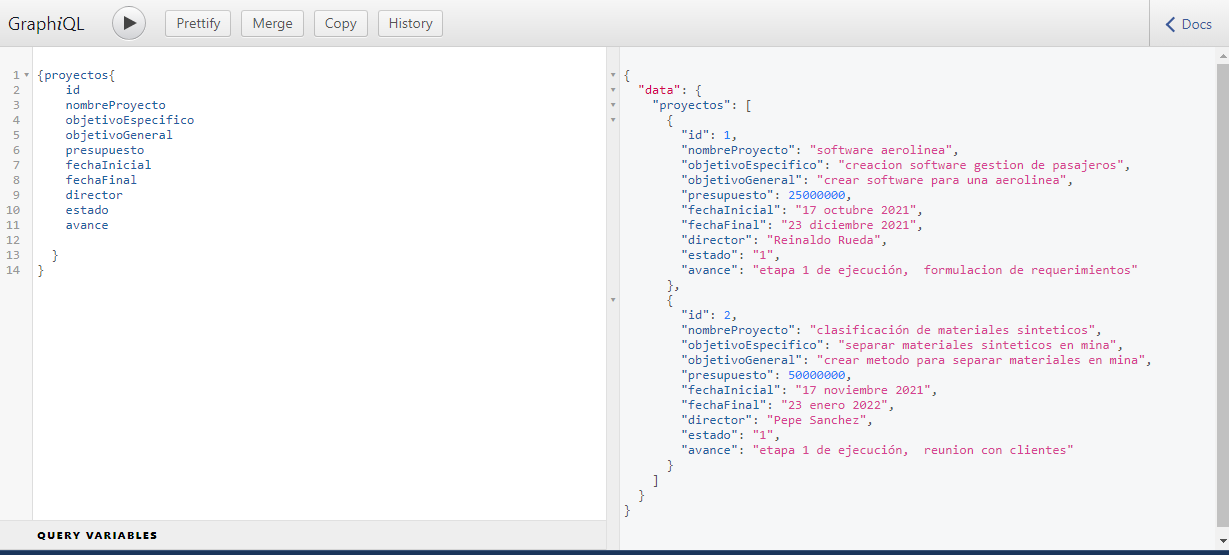


Posterior a ello se corre el mismo localhost para poder interactuar.

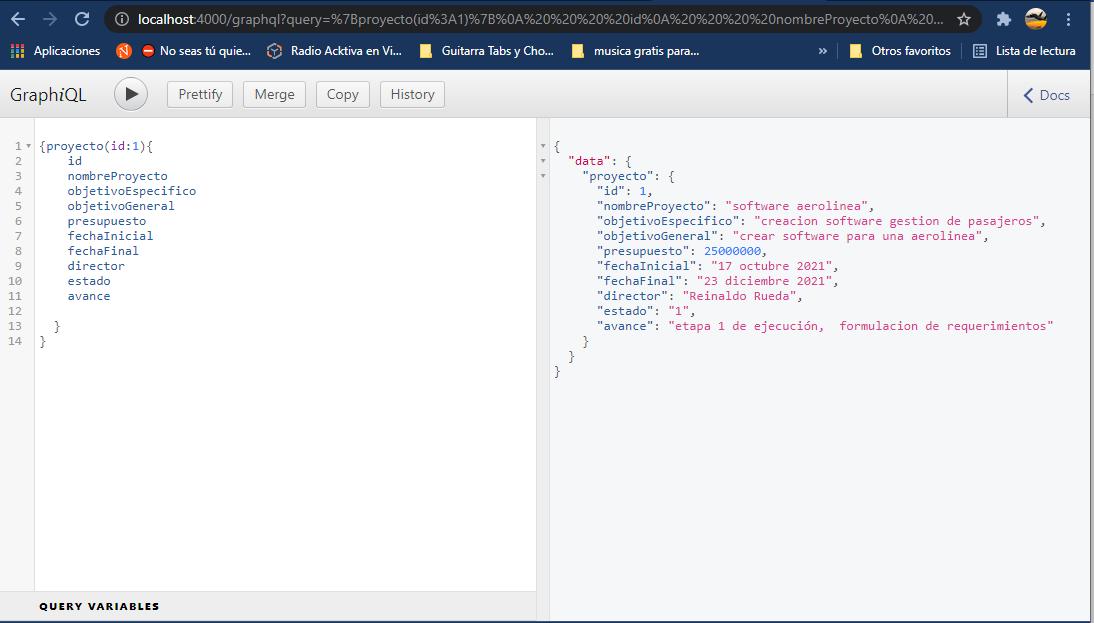
Luego de ejecutar y agregar un proyecto y se consulta a la vez el proyecto creado.



Así como se pueden usar el query para determinar todos los estudiantes, se desarrolla ahora para encontrar todos los proyectos disponibles, así;



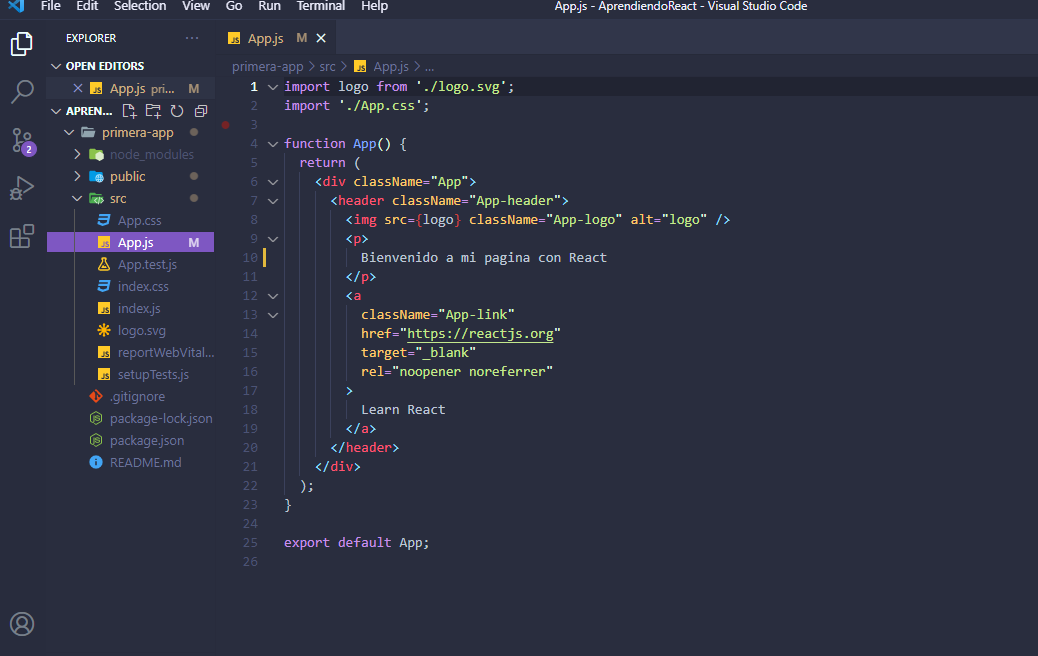
Ahora según las especificaciones también se solicitaba el hacer la consulta de algún proyecto dependiendo de un id determinado.



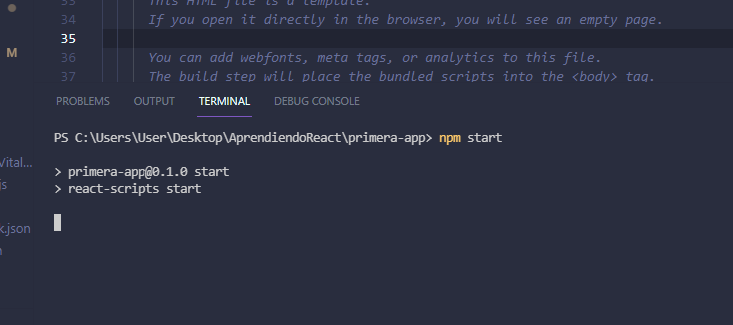
PARTE REACT:

A partir de esta parte se presenta la forma como se crea el proyecto en React.

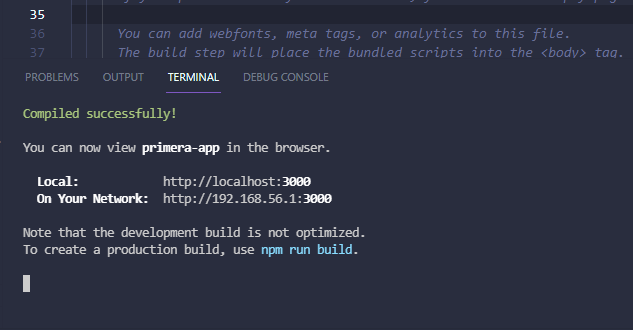
Primero, se inicia el proyecto mediante el gestor de paquetes npm, para ello se inicia con el comando= npx create-react-app nombre\_aplicacion



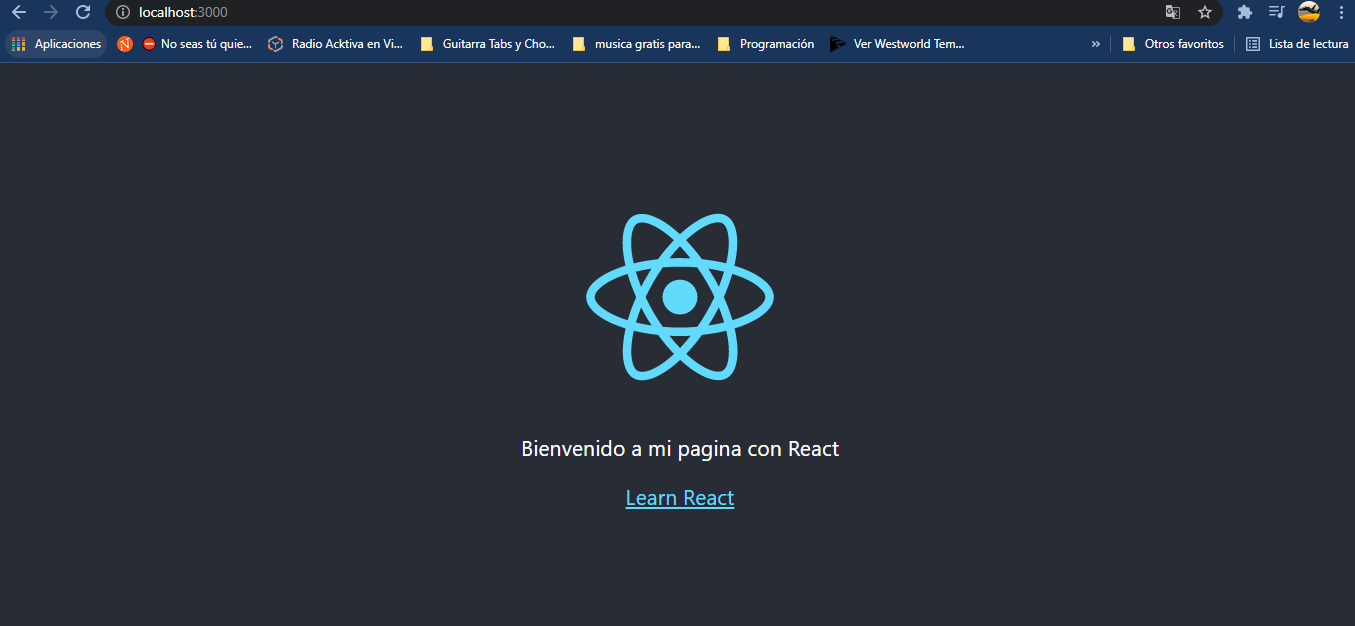
Y luego la ponemos a correr en el localhost:



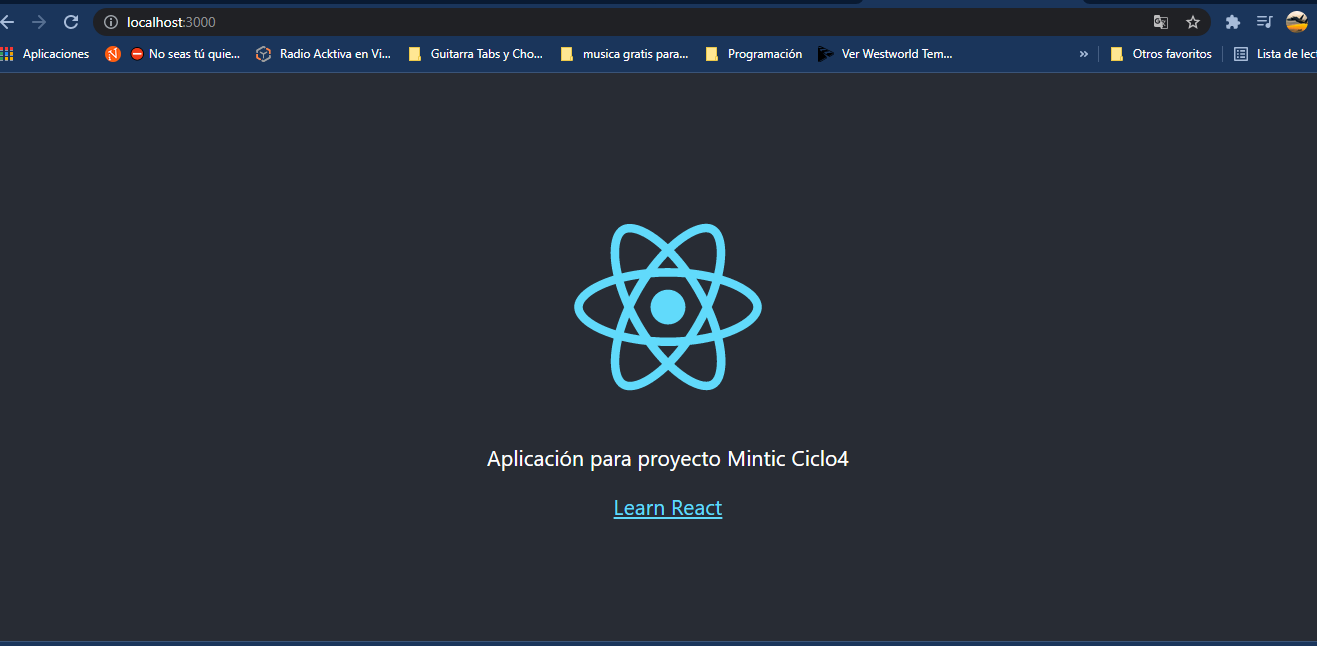
Posterior a ello aparece un mensaje de conexión exitoso.



De esta manera se inicia la interacción con el frontenend de nuestra aplicación con React en el respectivo localhost, en este caso el 3000.



Ya para finalizar si se puede proceder a hacer las modificaciones que sean necesarias, para este ejemplo cambia el mensaje que aparecia y aparece el mensaje “Aplicación para proyecto Mintic ciclo4”.



De la misma manera en dicha parte se agrega la parte del frontened adecuada a cada proyecto que se esté desarrollando.

A continuación se presenta el link de github donde se encuentrar los documentos que se han venido desarrollando a lo largo del curso.

Repositorio (URL-gitHub)= <https://github.com/jaimeand/RepoProyectoWebG6.git>