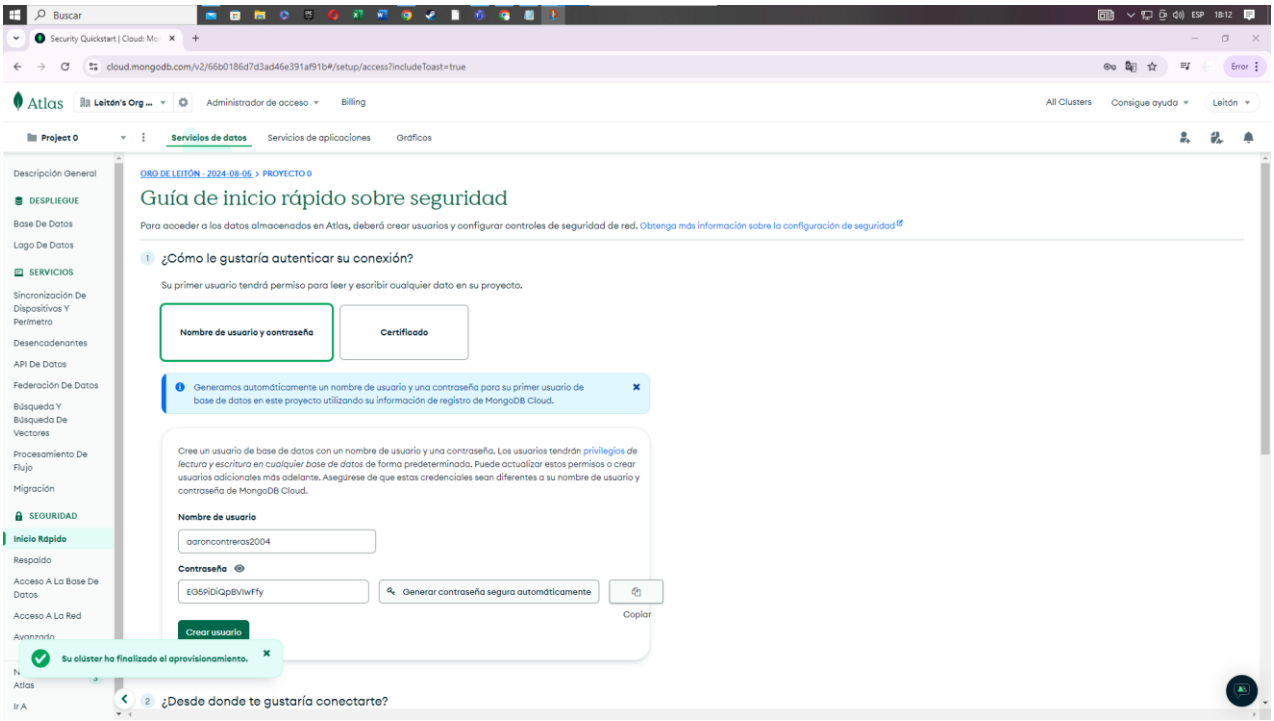
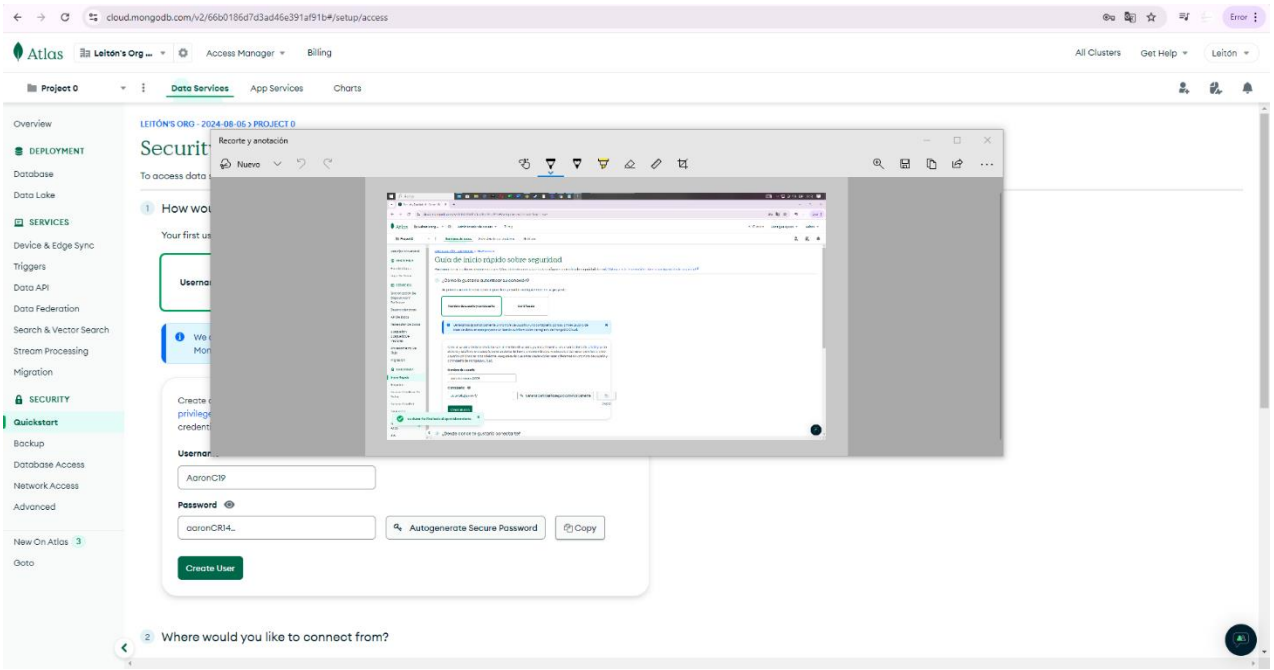
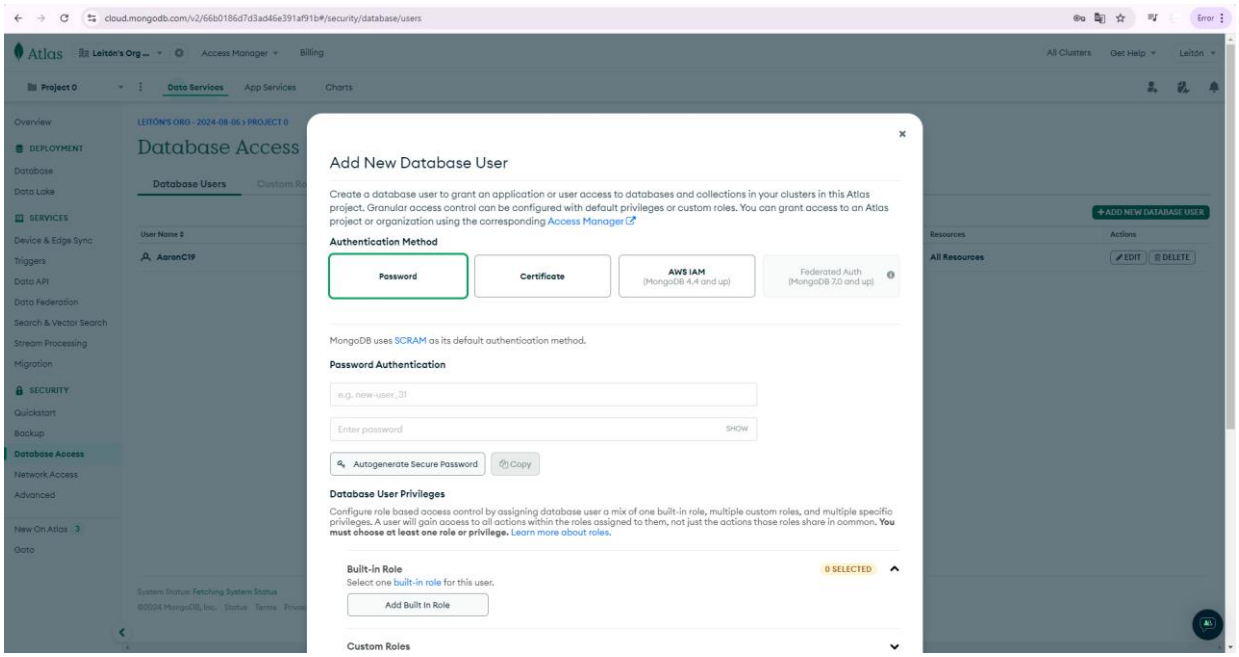


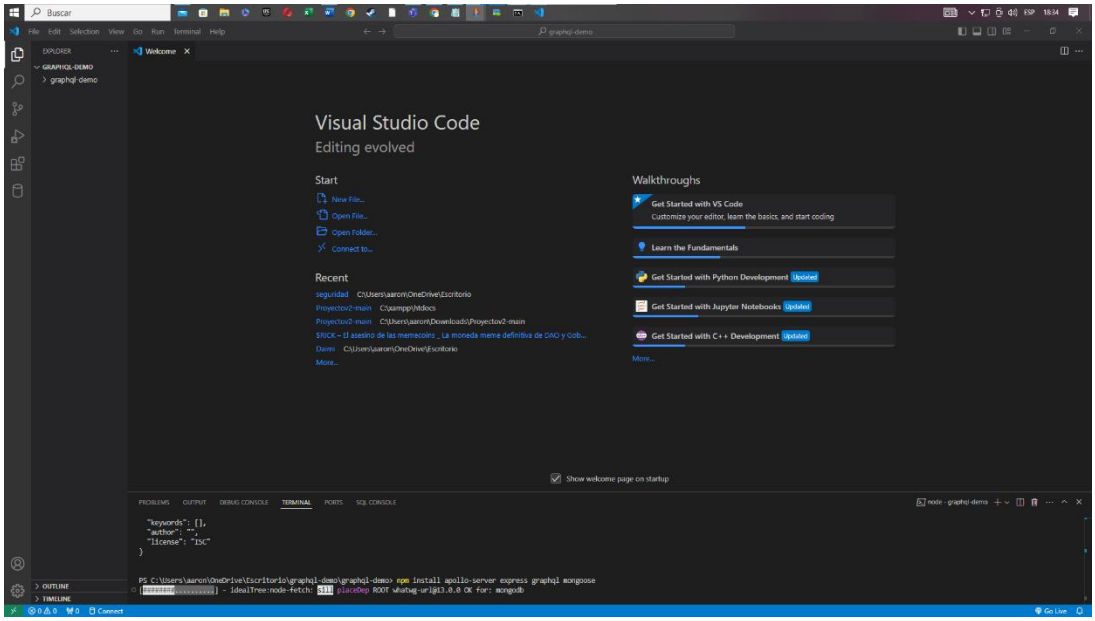
Una vez creada la cuenta de mongoDB Atlas, procedemos a configurar el inicio rápido



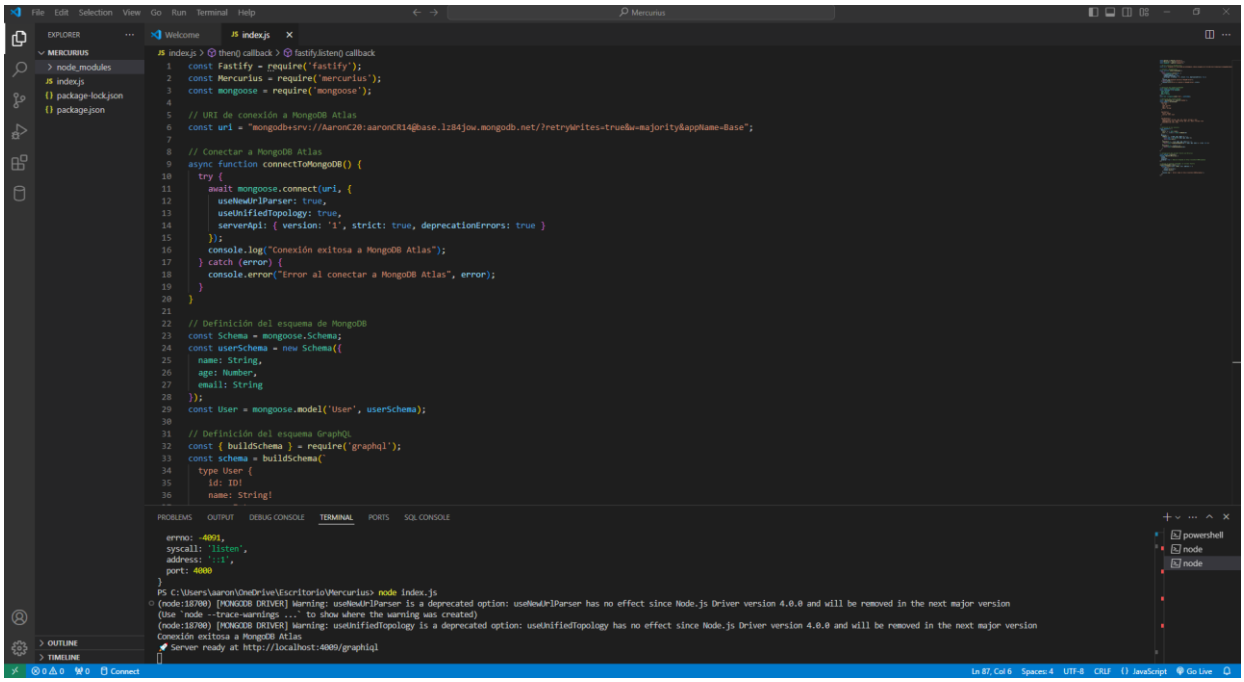
Creamos un usuario con permiso de lectura y escritura en la base de datos. Le asignamos el usuario y la contraseña la cual tendremos que ingresar en el index.js para poder establecer la conexión también.



Terminamos de configurarlo y le damos a crear nuevo usuario

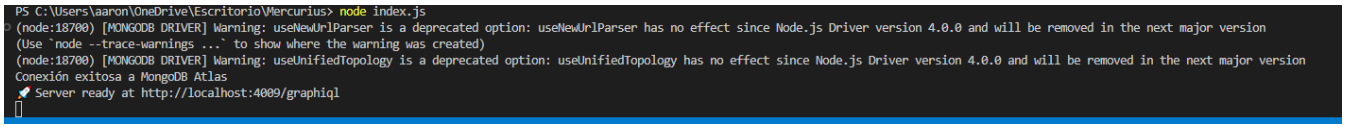


Una vez creado el usuario y base de datos Abrimos Visual Studio Code y seleccionamos la carpeta en la cual vamos a realizarlo

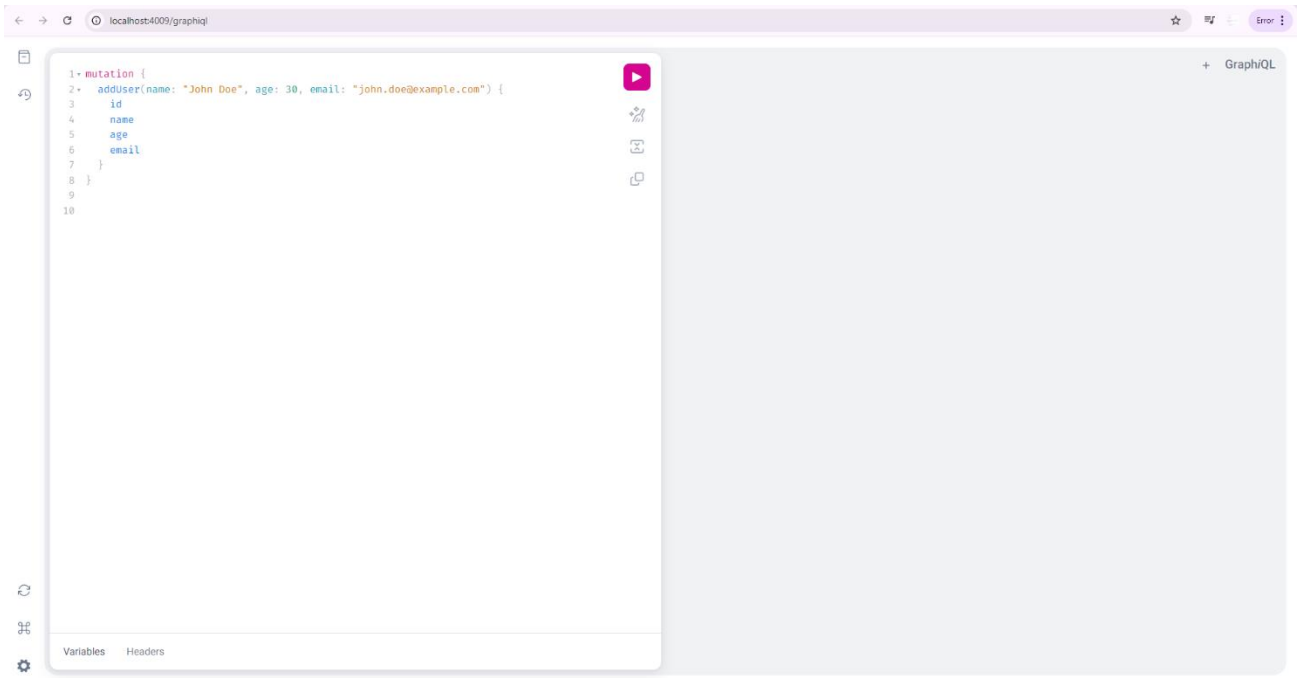


En mi caso utilicé Mercurius, entonces procedí a instalarlo en la terminal con le siguiente comando: npm install fastify mercurius mongoose

Creamos la carpeta index.js la cual lleva el código en el cual se realiza el proceso necesario.

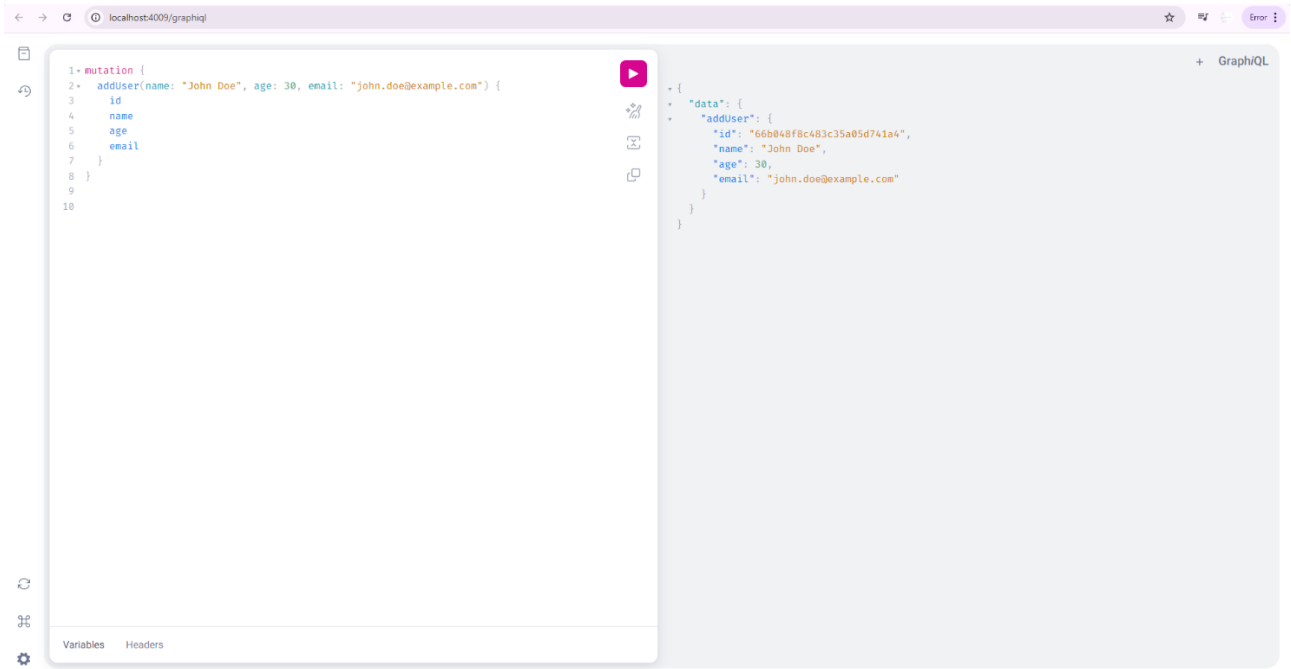


Una vez ejecutado el comando node index.js el cual ejecuta el index.js que creamos anteriormente. Nos brindará la url en la cual procederemos a realizar lo siguiente

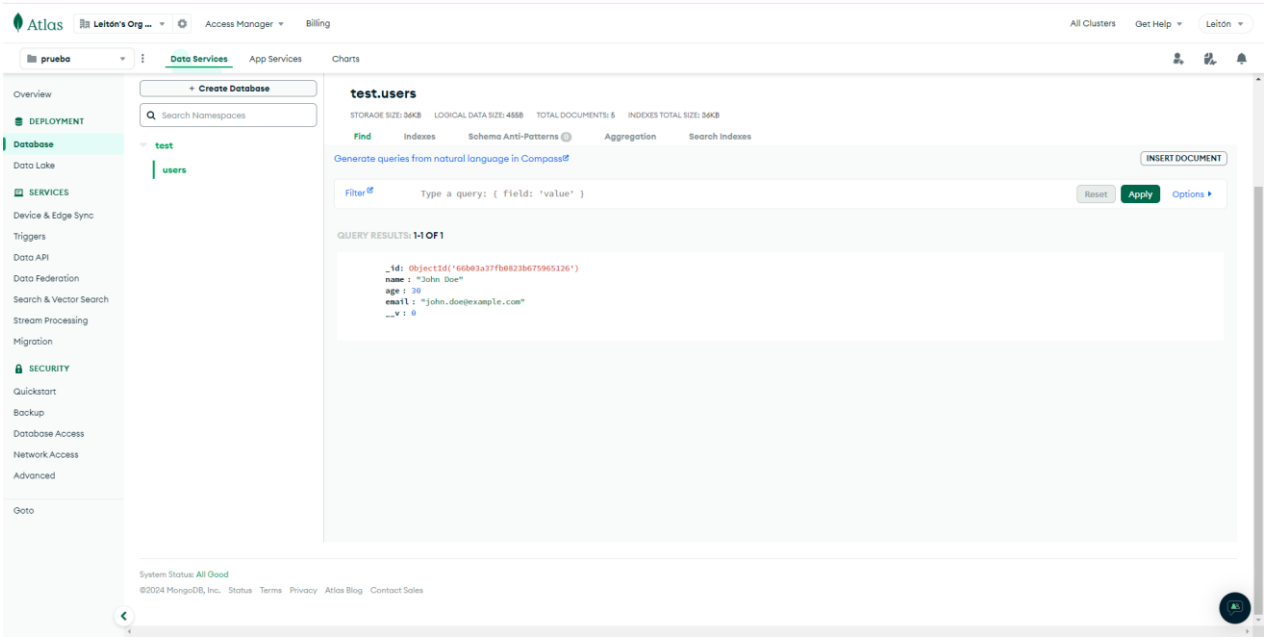


Se utiliza Mercurius como el servidor GraphQL y GraphiQL como la interfaz gráfica de usuario para interactuar con el servidor GraphQL

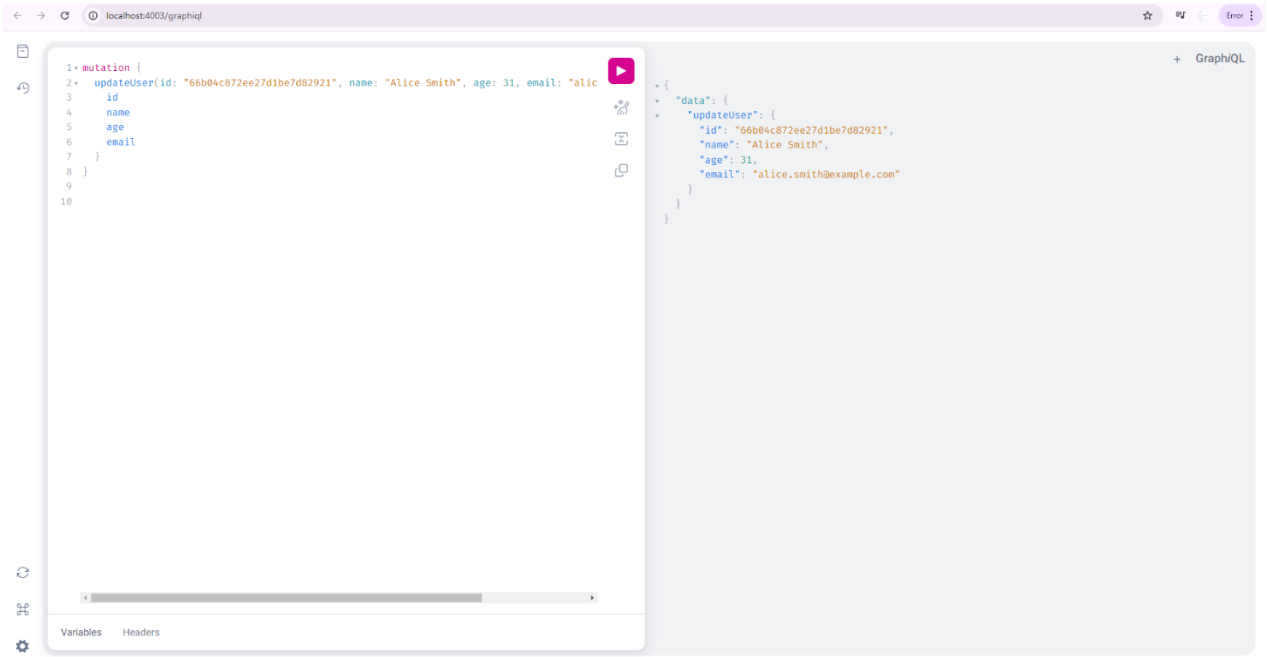
Mercurius es un plugin para Fastify que permite crear un servidor GraphQL de manera eficiente



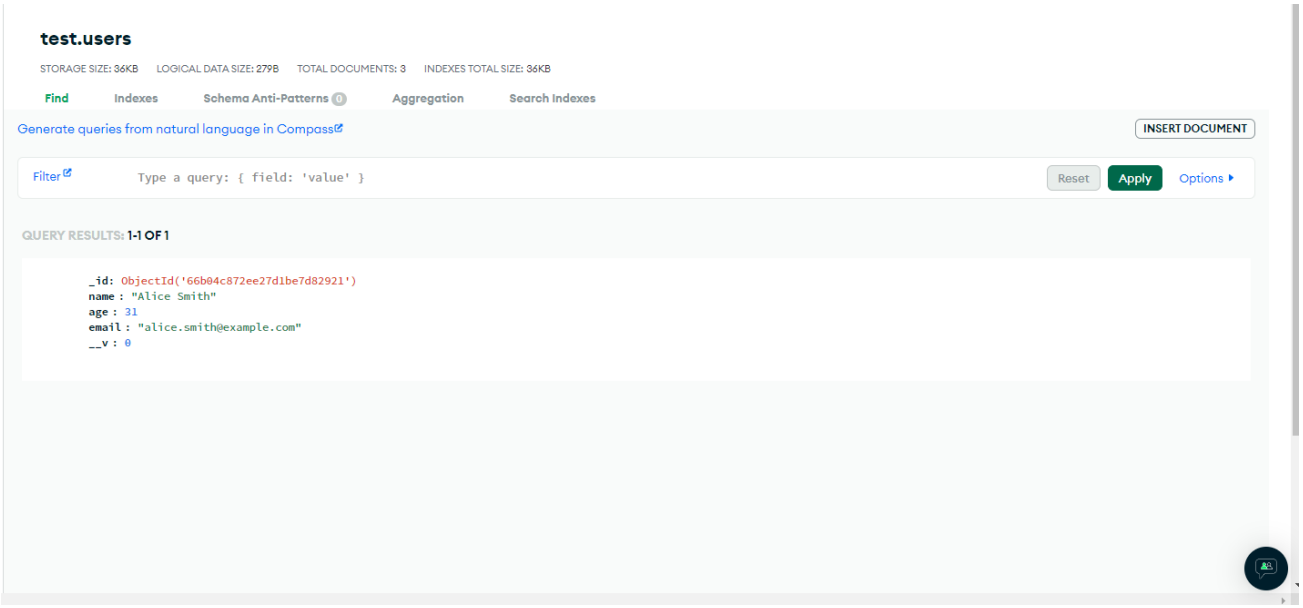
Una vez ejecutada la mutación, iremos a verificar en nuestra base de datos de mongoDB Atlas que se haya hecho la inserción correctamente.



Podremos observar que los datos que ingresamos en la interfaz fueron añadidos correctamente en la base de datos.



En este caso realicé un update



El cual fue aplicado también en la base de datos.

Estas serían las consultas que podríamos realizar:

(Consultar Todos los Usuarios)

```
query {
  users {
    id
    name
    age
    email
  }
}
```

(Consultar un Usuario Específico o campos espécificos)

```
query {
  user(id: "ID_DEL_USUARIO") {
    id
    name
    age
    email
  }
}
```

(Consultar el Total de Usuarios)

```
query {
  totalUsers
}
```

(Eliminar un usuario)

```
mutation {
  deleteUser(id: "ID_DEL_USUARIO") {
    id
    name
    age
    email
  }
}
```

(contar el número de usuarios por edad)

```
query {
  usersByAge {
    age
    count
  }
}
```

(Agregar un nuevo usuario)

```
mutation {
  addUser(name: "Alice", age: 30, email: "alice@gmail.com") {
    id
    name
    age
    email
  }
}
```

(Modificar un usuario)

```
mutation {
  updateUser(id: "ID_DEL_USUARIO", name: "Bob Marley", age: 26, email:
"bob.marley@gmail.com") {
    id
    name
    age
    email
  }
}
```