

## Parcial práctico Arquitectura de Software(GraphQL)

Grupo 4

Mateo Rivera Monsalve

Kieffer Alexander Páez Rozo

Diego Armando Muñoz Carvajal

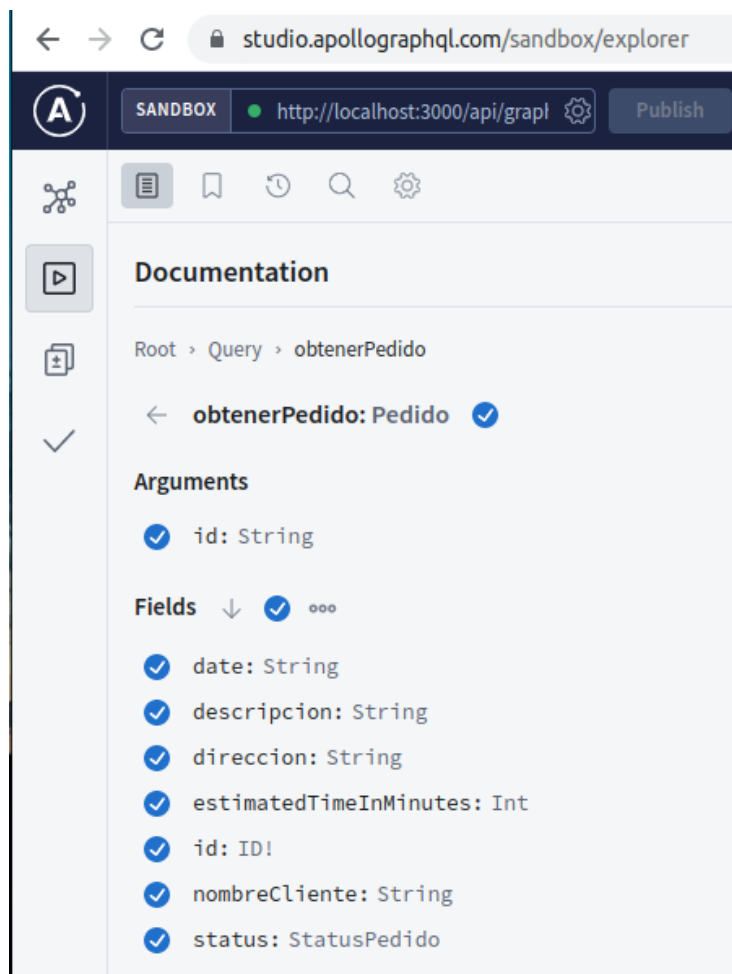
Santiago Bedoya Diaz

Jhonatan Tamayo Morales

### Pantallazos del funcionamiento de la API

#### 1) Consultar un pedido

- Formato del Query con los argumentos de la petición y los campos de la respuesta



\* Cuerpo de la petición

The screenshot shows a REST client interface with two tabs: "ExampleQuery" and "ObtenerPedido". The "ObtenerPedido" tab is active. The interface is divided into two main sections: "Operation" and "Variables".

In the "Operation" section, a GraphQL query is defined:

```
1 query ObtenerPedido($obtenerPedidoId: String) {  
2   obtenerPedido(id: $obtenerPedidoId) {  
3     date  
4     descripcion  
5     direccion  
6     estimatedTimeInMinutes  
7     nombreCliente  
8     status  
9     id  
10  }  
11 }
```

Below the query, there is a "Variables" section with a JSON object:

```
1 {  
2   "obtenerPedidoId": "clabs0qy500032mcig0mdbvq3"  
3 }
```

The interface includes standard REST client controls like "ObtenerPedido" button, copy, and save icons.

\* Respuesta del servidor

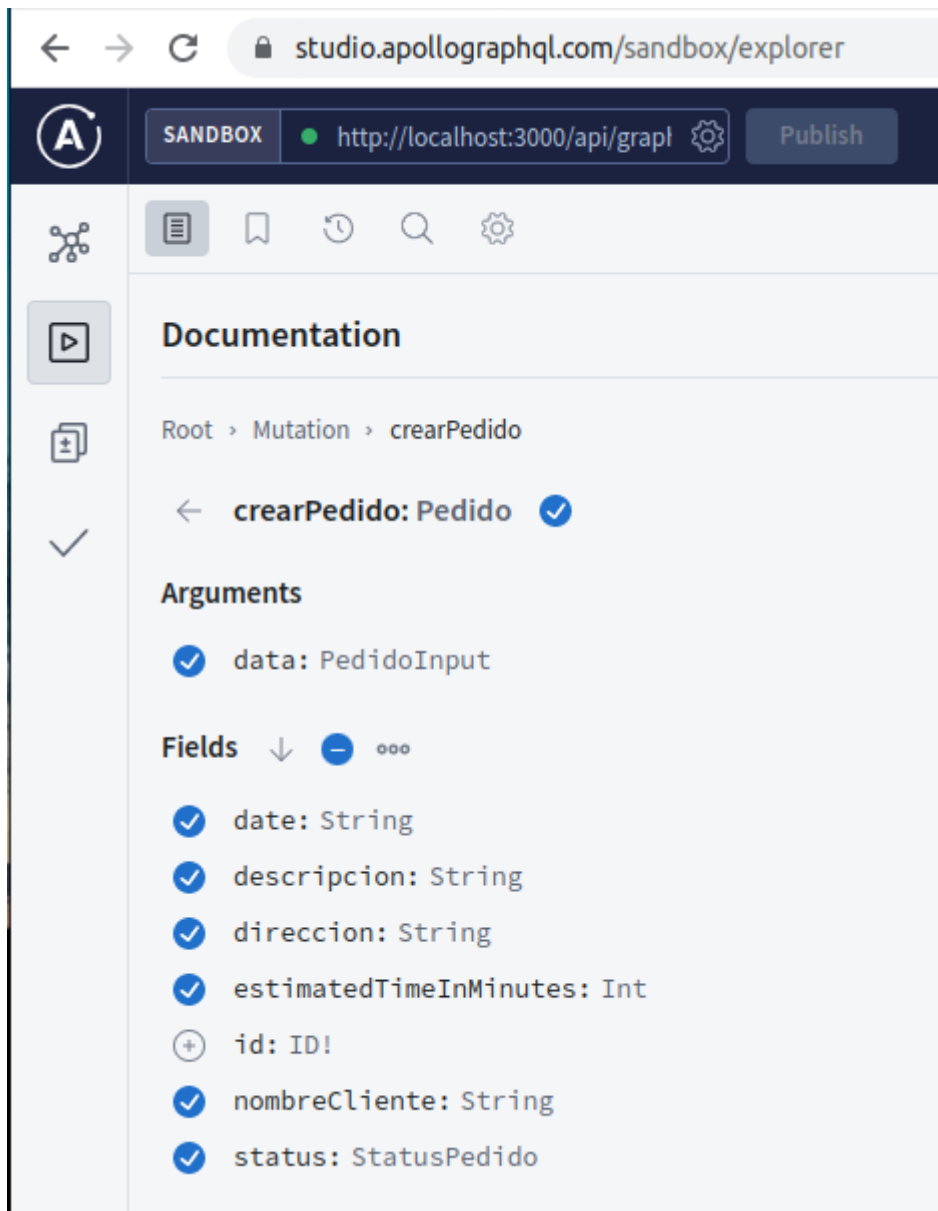
```
Response  ▾  [Icons]  STATUS 200 | 503ms | 210B

{
  "data": {
    "obtenerPedido": {
      "date": "1668127461485",
      "descripcion": "Un hotdog",
      "direccion": "Mi casita",
      "estimatedTimeInMinutes": 45,
      "nombreCliente": "Kieffer",
      "status": "Ingresado",
      "id": "clabs0qy500032mcig0mdbvq3"
    }
  }
}
```

2) Realizar un pedido

- Formato de la petición “Crear pedido” con los argumentos de la petición y los campos de la respuesta.

Notar que el argumento para la creación de un pedido (data) es de tipo “PedidoInput” el cual fue definido en el schema de graphql



## Cuerpo de la petición “Crear Pedido”

ExampleQuery

ObtenerPedido

Mutation

×

+

Operation

↑

▼

📄

▼

▶ Mutation

```
1  mutation Mutation($data: PedidoInput) {  
2    crearPedido(data: $data) {  
3      date  
4      descripcion  
5      direccion  
6      estimatedTimeInMinutes  
7      nombreCliente  
8      status  
9    }  
10 }
```

...

🔍

Variables

Headers

▼

```
1  {  
2    "data": {  
3      "descripcion": "Una gaseosa con hamburguesa doble carne",  
4      "direccion": "Oficina UdeA",  
5      "estimatedTimeInMinutes": 35,  
6      "nombreCliente": "Diego Botia"  
7    }  
8  }
```

JSON

Respuesta de la petición "Crear pedido":

```
Response ▾ [Icons] STATUS 200 | 1.08s | 212B

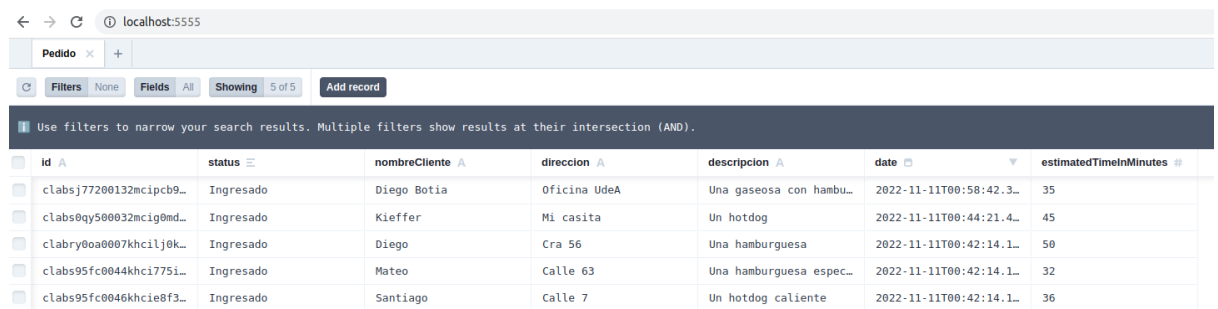
{
  "data": {
    "crearPedido": {
      "date": "1668130076351",
      "descripcion": "Una gaseosa con
hamburguesa doble carne",
      "direccion": "Oficina UdeA",
      "estimatedTimeInMinutes": 35,
      "nombreCliente": "Diego Botia",
      "status": "Ingresado"
    }
  }
}
```

## Schema del data graph

```
.env  types.ts  resolvers.ts  schema.prisma
graphql > server > Pedido > types.ts > ...
1  import { gql } from 'apollo-server-micro';
2
3  const PedidoTypes = gql`
4    enum StatusPedido {
5      Ingresado
6      EnProceso
7      Despachado
8      Entregado
9    }
10   input PedidoInput {
11     nombreCliente: String!
12     direccion: String!
13     estimatedTimeInMinutes: Int!
14     descripcion: String!
15   }
16   type Pedido {
17     id: ID!
18     status: StatusPedido
19     nombreCliente: String
20     direccion: String
21     descripcion: String
22     date: String
23     estimatedTimeInMinutes: Int
24   }
25
26   type Query {
27     obtenerPedido(id: String): Pedido
28   }
29   type Mutation {
30     crearPedido(data: PedidoInput): Pedido
31   }
32 `;
33
34 export { PedidoTypes };
35
```

3) Para la base de datos se utilizó PostgreSQL y el ORM Prisma

**URI de la base de datos:** "postgres://mycreds.db.elephantsql.com:5432/ngimluxm"



id	status	nombreCliente	direccion	descripcion	date	estimatedTimeInMinutes
clabsj77200132mcipcb9...	Ingresado	Diego Botia	Oficina UdeA	Una gaseosa con hambu...	2022-11-11T00:58:42.3...	35
clabs0qy500032mcig0md...	Ingresado	Kieffer	Mi casita	Un hotdog	2022-11-11T00:44:21.4...	45
clabry0oa0007khcilj0k...	Ingresado	Diego	Cra 56	Una hamburguesa	2022-11-11T00:42:14.1...	50
clabs95fc0044khci775i...	Ingresado	Mateo	Calle 63	Una hamburguesa espec...	2022-11-11T00:42:14.1...	32
clabs95fc0046khcie8f3...	Ingresado	Santiago	Calle 7	Un hotdog caliente	2022-11-11T00:42:14.1...	36

Vista de la base de datos, con todos los pedidos que se han creado y los datos de cada uno de estos

## Enlace al github

<https://github.com/MateoRivera/ArqSoftExam>