**Documentación**

**1. Documentación del Servicio GraphQL**

**1.1. Uso de GraphQL Playground**

**GraphQL Playground** es una herramienta integrada que permite interactuar con tu API GraphQL, facilitando la documentación y prueba de los endpoints.

**1.1.1. Acceso a GraphQL Playground**

Cuando ejecutas tu aplicación NestJS localmente, GraphQL Playground está disponible en:

http://localhost:3000/graphql

**1.1.2. Exploración del Esquema**

* Al abrir GraphQL Playground, verás una interfaz dividida en tres secciones: **Editor**, **Resultados** y **Documentación**.
* En la sección de **Documentación** (barra lateral derecha), puedes explorar el esquema de tu API, incluyendo los **Tipos**, **Consultas**, **Mutaciones** y **Suscripciones** disponibles.

**1.1.3. Ejemplos de Operaciones**

**Consultar Todos los Items**

**Consulta:**

graphql

Copiar código

query GetItems {

items {

id

nombre

descripcion

fechaCreacion

}

}

**Consultar un Item por ID**

**Consulta:**

graphql

Copiar código

query GetItem {

item(id: 1) {

id

nombre

descripcion

fechaCreacion

}

}

**Crear un Nuevo Item**

**Mutación:**

graphql

Copiar código

mutation CreateItem {

createItem(createItemInput: { nombre: "Nuevo Item", descripcion: "Descripción del nuevo item" }) {

id

nombre

descripcion

fechaCreacion

}

}

**Actualizar un Item Existente**

**Mutación:**

graphql

Copiar código

mutation UpdateItem {

updateItem(updateItemInput: { id: 1, nombre: "Item Actualizado", descripcion: "Descripción actualizada" }) {

id

nombre

descripcion

fechaCreacion

}

}

**Eliminar un Item**

**Mutación:**

graphql

Copiar código

mutation RemoveItem {

removeItem(id: 1) {

id

nombre

descripcion

fechaCreacion

}

}

**1.1.4. Documentación de Esquemas y Tipos**

Puedes ver la definición de cada tipo y sus campos en la sección de **Documentación**. Por ejemplo, el tipo Item tiene los siguientes campos:

* **id**: Int
* **nombre**: String
* **descripcion**: String
* **fechaCreacion**: DateTime

**2. Instrucciones para Ejecutar y Probar la Aplicación en el Entorno Local**

A continuación, se detallan los pasos para ejecutar y probar la aplicación en tu entorno local.

**2.1. Requisitos Previos**

* **Node.js** (versión 16 o superior)
* **npm** (viene con Node.js)
* **Docker** (opcional, pero recomendado para ejecutar RabbitMQ y PostgreSQL)
* **Git**

**2.2. Clonar el Repositorio**

Clona el repositorio de tu proyecto:

git clone https://github.com/tu\_usuario/nestjs-graphql-api.git

cd nestjs-graphql-api

**2.3. Configurar Variables de Entorno**

Crea un archivo .env en la raíz del proyecto con el siguiente contenido:

env

# .env

DB\_HOST=localhost

DB\_PORT=5432

DB\_USERNAME=postgres

DB\_PASSWORD=postgres

DB\_NAME=nestjs\_graphql

RABBITMQ\_URL=amqp://guest:guest@localhost:5672

**2.4. Iniciar PostgreSQL**

**Opción 1: Usar Docker**

Ejecuta el siguiente comando para iniciar PostgreSQL en un contenedor Docker:

docker run --name postgres -e POSTGRES\_USER=postgres -e POSTGRES\_PASSWORD=postgres -e POSTGRES\_DB=nestjs\_graphql -p 5432:5432 -d postgres:13

**Opción 2: Instalar PostgreSQL Localmente**

Si prefieres instalar PostgreSQL en tu máquina, sigue las instrucciones en <https://www.postgresql.org/download/>.

**2.5. Iniciar RabbitMQ**

Ejecuta el siguiente comando para iniciar RabbitMQ en un contenedor Docker:

docker run -d --hostname my-rabbit --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3-management

* La interfaz de administración de RabbitMQ estará disponible en http://localhost:15672.
* Credenciales por defecto:
  + **Usuario**: guest
  + **Contraseña**: guest

**2.6. Instalar Dependencias**

En la raíz del proyecto, ejecuta:

npm install

**2.7. Ejecutar la Aplicación**

Para iniciar la aplicación en modo de desarrollo:

npm run start:dev

La aplicación estará disponible en http://localhost:3000.

**2.8. Ejecutar el Microservicio Consumidor**

El microservicio consumidor está en un proyecto separado llamado item-consumer.

**2.8.1. Clonar el Repositorio del Consumidor**

Clona el repositorio:

git clone https://github.com/tu\_usuario/item-consumer.git

cd item-consumer

**2.8.2. Instalar Dependencias**

npm install

**2.8.3. Configurar Variables de Entorno**

Crea un archivo .env en la raíz del proyecto item-consumer:

env

# .env

RABBITMQ\_URL=amqp://guest:guest@localhost:5672

**2.8.4. Ejecutar el Consumidor**

npm run start

**2.9. Probar la Aplicación**

Utiliza GraphQL Playground en http://localhost:3000/graphql para realizar operaciones y verificar que el consumidor está recibiendo los mensajes.

* **Crear un Item:** Ejecuta la mutación para crear un nuevo item y observa los logs en la consola del consumidor.
* **Actualizar un Item:** Ejecuta la mutación para actualizar un item existente y verifica los logs.