

**Introducción a los Sistemas Inteligentes  
2024-2**

# Solución Práctica 3

**Estudiante:**

**Sebastián Escandón Flórez**

**Profesor:**

**Fabio Augusto González Osorio**

## Desarrollo de la práctica

### Enunciado: “Usted tiene 2 jarras, una con capacidad de 3 litros y otra con capacidad de 5 litros. Además, se cuenta con un grifo. Las jarras se pueden llenar totalmente, vaciar en el piso y vaciar parcialmente el contenido de una jarra en la otra. Las jarras no tienen ninguna clase de marca. El objetivo es medir exactamente un litro de agua”

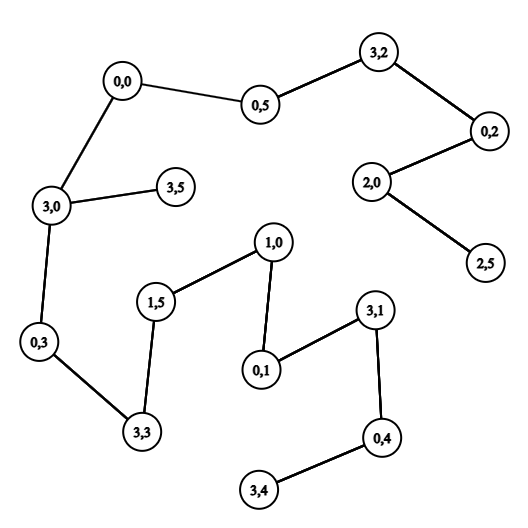
### i. ¿Cuáles son los estados del problema?

Los estados del problema están definido por una tupla (x,y), en donde:

* x equivale al volumen actual de la jarra 1 (En este caso la de 3 litros)
* y equivale al volumen actual de la jarra 2 (En este caso la de 5 litros)

### ii. ¿Cuál es el tamaño del espacio de estados?

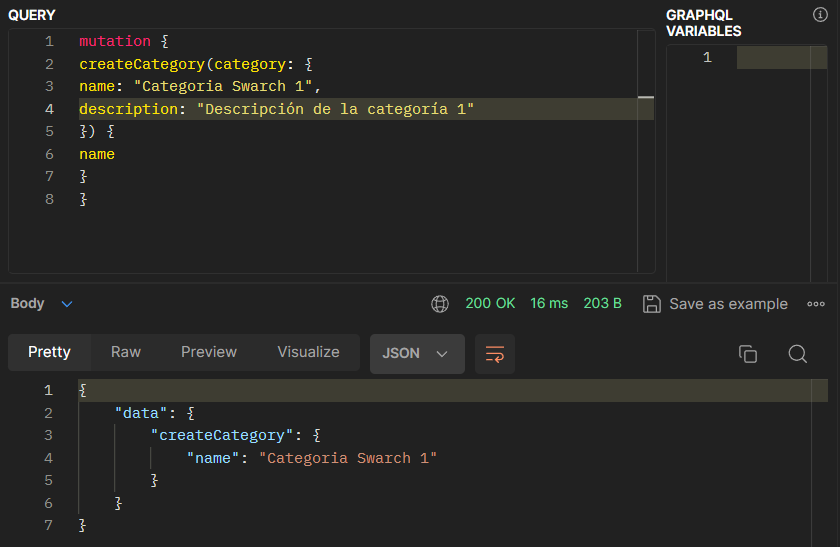
El tamaño del espacio de estados es de 16, correspondiente al siguiente grafo:

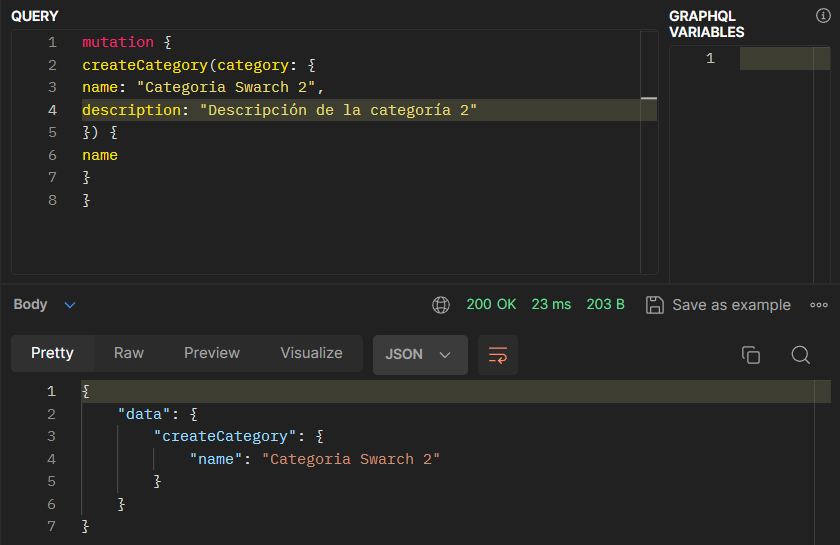
  
Generado en: [csacademy graph editor](https://csacademy.com/app/graph_editor/)

### iii. Soporte visual de la ejecución de las peticiones HTTP sobre la API-GraphQL del API Gateway. iv. Soporte visual de las acciones evidenciadas en la base de datos, tras la ejecución de las peticiones sobre el API Gateway

1. **Mutations i**
   1. **Creación de una categoría**

**Evidencia de petición a GraphQL por medio de Postman**

****

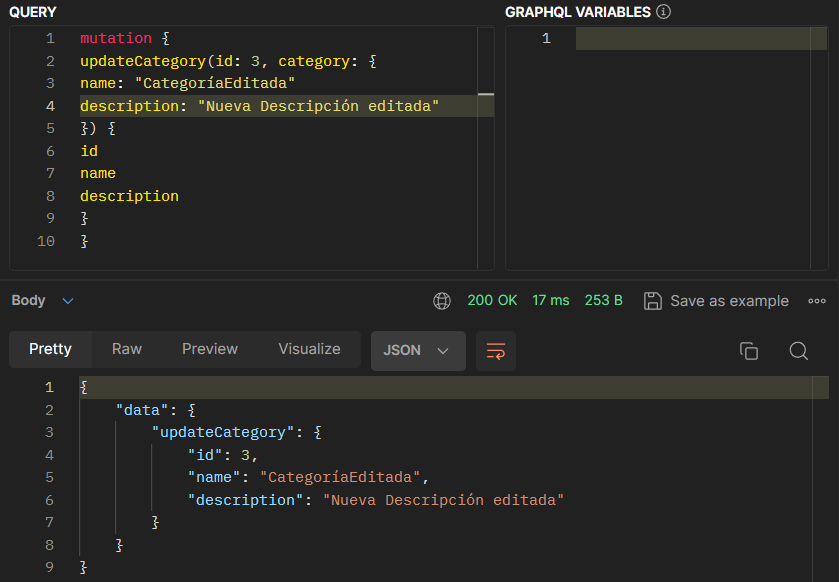
****

**Evidencia en base de datos**

****

* 1. **Modificación de una categoría**

**Evidencia de petición a GraphQL por medio de Postman**

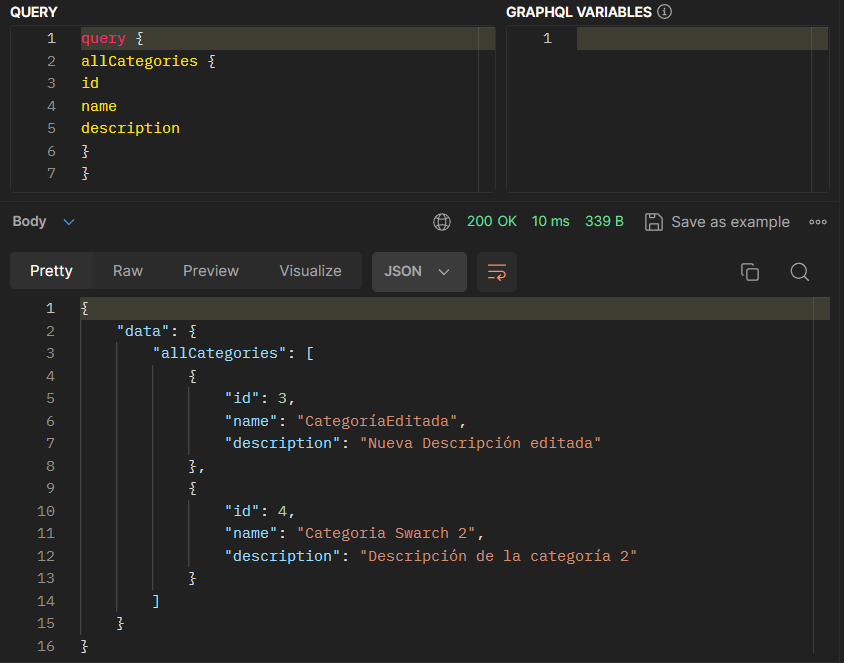
****

**Evidencia en base de datos**

****

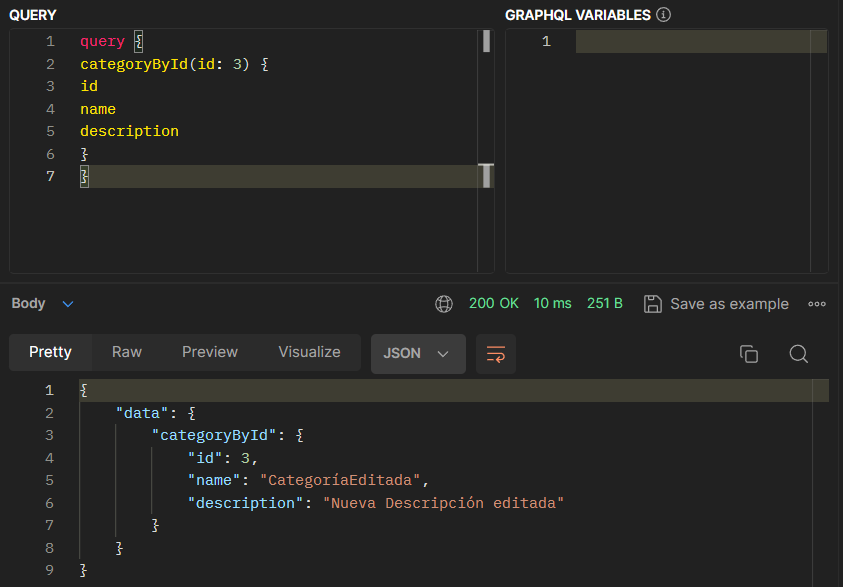
1. **Queries**
   1. **Consulta de todas las categorías**

**Evidencia de petición a GraphQL por medio de Postman**

****

* 1. **Consulta de una categoría en específico**

**Evidencia de petición a GraphQL por medio de Postman**

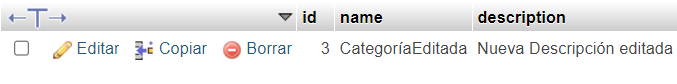


1. **Mutations ii**
   1. **Remoción de una categoría**

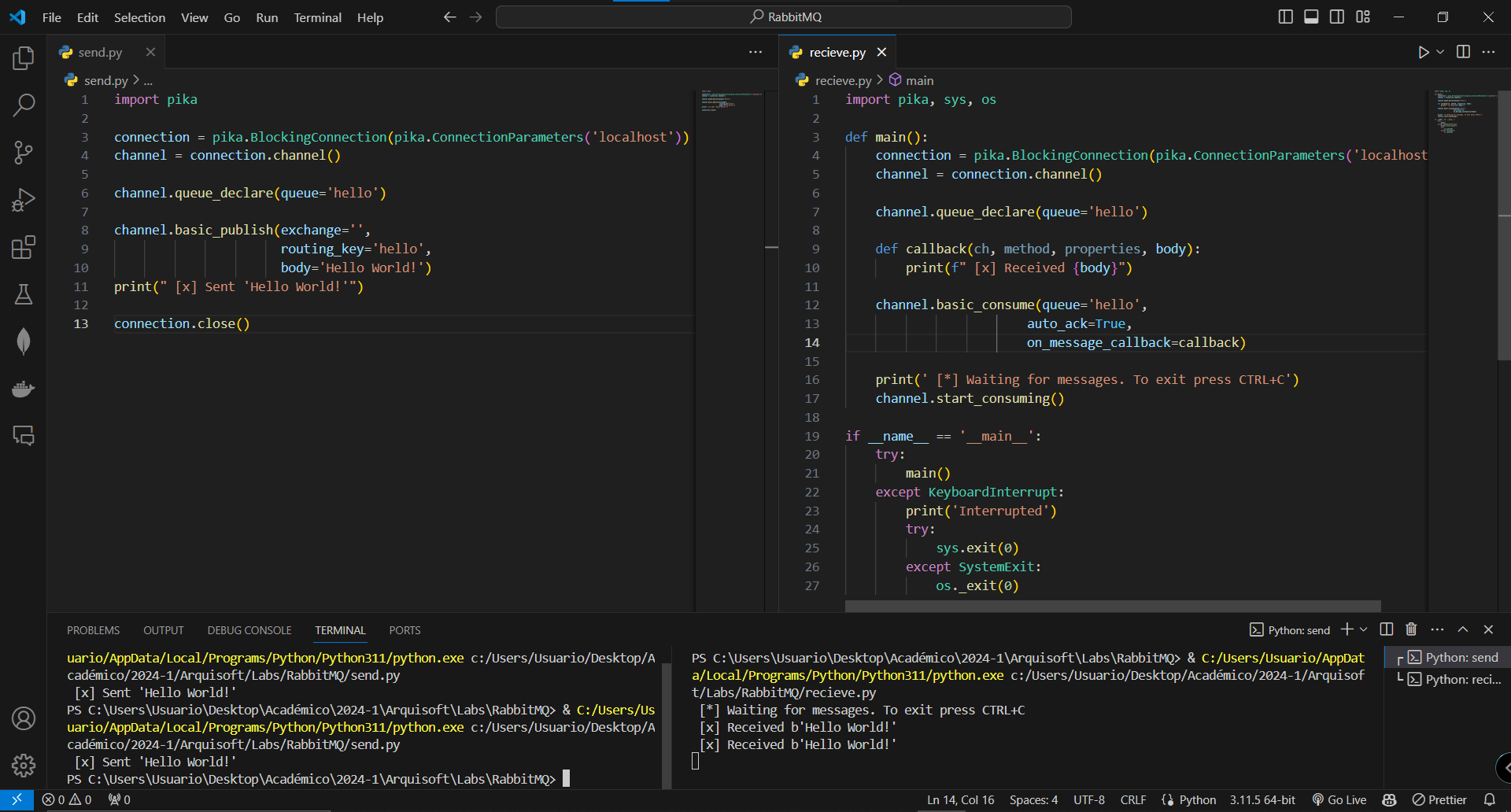
**Evidencia de petición a GraphQL por medio de Postman**

****

**Evidencia en base de datos**

****

### v. Soporte visual del desarrollo del ejercicio de RabbitMQ.

****En la captura anterior se pueden evidenciar los dos programas *send.py*  y *recieve.py*, los cuáles, en conjunto realizan el proceso de enviar y recibir un mensaje, que en este caso es “Hello World”, se evidencia que cada vez que se ejecuta el primer programa de la izquierda (el responsable de enviar el mensaje), este se ve reflejado en la terminal que está ejecutando el ciclo infinito del programa de la derecha (el responsable de recibir el mensaje).