

Documentación del Microservicio: Creación de Usuarios mediante GraphQL

1. Datos Generales

- **Nombre del estudiante:** Joan Riveros Soria
- **Carrera / Materia:** Ingeniería de Sistemas / Tecnologías Web II
- **Fecha:** 3 de mayo de 2025
- **Proyecto integrador:** Repositorio de Documentos Históricos Digitalizados del Bi-Centenario
- **Tecnologías utilizadas:** FastAPI (backend), Node.js + GraphQL (microservicio)
- **Base de datos realizada en:** pgAdmin4

2. Objetivo del Microservicio

Este microservicio permite crear usuarios mediante una mutación GraphQL. Al recibir los datos desde el cliente, se conecta al backend del proyecto integrador desarrollado en FastAPI y registra al usuario utilizando el endpoint REST `/users/register`.

3. Análisis del Proyecto Integrador

- **Endpoint elegido:** POST `/users/register`
- **Justificación de la elección:** El endpoint de usuarios es fundamental para gestionar el acceso e identidad en el sistema. Además, permite validar completamente el flujo de integración entre servicios REST y GraphQL.
- **Estructura técnica del endpoint:**
 - Método: POST
 - Formato de datos: JSON
 - Ejemplo de cuerpo:

```
{  
  "email": "joan@ucb.edu.bo",  
  "password": "123456"  
}
```

4. Diseño del Microservicio

- Tecnología elegida: GraphQL
- **Librerías principales:** express, express-graphql, graphql, axios

- **Justificación:** GraphQL permite una integración flexible y moderna, ofreciendo una interfaz declarativa al cliente que encapsula la lógica REST del backend. Facilita el desacoplamiento del microservicio respecto al sistema principal.

5. Diagrama del Flujo de Integración

Cliente (GraphQL)

↓ mutation

Microservicio Node.js (GraphQL)

↓ POST /users

Backend FastAPI (Python)

↓ Usuario creado

Respuesta final al cliente

6. Implementación Técnica

- Se definió el tipo `User` y la mutación `createUser` en el esquema GraphQL.
- Se utilizó `axios.post()` para realizar una solicitud al backend FastAPI con los datos del usuario.
- Se devolvió la respuesta formateada al cliente a través de GraphQL.
- Se manejaron errores como duplicidad de email y se reportaron correctamente en el cliente.

7. Pruebas Realizadas

- Se realizaron pruebas con **GraphQL Playground** y **Postman**.
- Se probaron diferentes escenarios:
 - Creación de usuarios válidos
 - Error por email ya registrado
 - Query básica para confirmar disponibilidad del servicio
- Se validó el flujo completo: Cliente → GraphQL → FastAPI → PostgreSQL

Capturas:

- Mutación exitosa:
 - Caso 1: usuario 2, joan@ucb.edu.bo.

GraphiQL

```

1 mutation {
2   createUser(email: "joan@ucb.edu.bo", password: "123456") {
3     id
4     email
5     is_active
6   }
7 }

```

```

{
  "data": {
    "createUser": {
      "id": 2,
      "email": "joan@ucb.edu.bo",
      "is_active": true
    }
  }
}

```

	id [PK] integer	email character varying	hashed_password character varying	is_active boolean	is_admin boolean
1	1	joan.riveros@ucb.edu.bo	5e88482ed421a69c992032232d256f23	true	false
2	2	joan@ucb.edu.bo	\$2b\$12\$Uj1GsprBEn8ZXdw73vE.j.Q1e7OYB7oPMTdTxi7q6h/wDQTz/HC...	true	false

- Caso 2: usuario 3, ana@ucb.edu.bo.

GraphiQL

```

1 mutation {
2   createUser(email: "ana@ucb.edu.bo", password: "123abc") {
3     id
4     email
5     is_active
6   }
7 }
8

```

```

{
  "data": {
    "createUser": {
      "id": 3,
      "email": "ana@ucb.edu.bo",
      "is_active": true
    }
  }
}

```

bi_centenario/postgres@PostgreSQL 16*

bi_centenario/postgres@PostgreSQL 16

Query Query History

```

1 SELECT * FROM public.users
2 ORDER BY id ASC
3
4
5

```

Data Output Messages Notifications

	id [PK] integer	email character varying	hashed_password character varying	is_active boolean	is_admin boolean
1	1	joan.riveros@ucb.edu.bo	5e88482ed421a69c992032232d256f23	true	false
2	2	joan@ucb.edu.bo	\$2b\$12\$Uj1GsprBEn8ZXdw73vE.j.Q1e7OYB7oPMTdTxi7q6h/wDQTz/HCr2	true	false
3	3	ana@ucb.edu.bo	\$2b\$12\$Qy3Em5jCwk9FFyPHwnB0.GG4KE9Et46VjcgJ2s6Rxqvhca6T/W...	true	false

- Caso 3: usuario 4, maria@ucb.edu.bo

GraphiQL

```

1 mutation {
2   createUser(email: "maria@ucb.edu.bo", password: "clave123") {
3     id
4     email
5     is_active
6   }
7 }
8

```

```

{
  "data": {
    "createUser": {
      "id": 4,
      "email": "maria@ucb.edu.bo",
      "is_active": true
    }
  }
}

```

query
mutation
subscription

bi_cenenario/postgres@PostgreSQL 16* x

bi_cenenario/postgres@PostgreSQL 16

Query Query History

```
1 SELECT * FROM public.users
2 ORDER BY id ASC
3
4
5
```

Data Output Messages Notifications

	id [PK] integer	email character varying	hashed_password character varying	is_active boolean	is_admin boolean
1	1	joan.riveros@ucb.edu.bo	5e88482ed421a69c992032232d256f23	true	false
2	2	joan@ucb.edu.bo	\$2b\$12\$Uj1GsprBEn8ZXdw73vE.j.Q1e7OYB7oPMTdTxl7q6h/wDQTz/HCr2	true	false
3	3	ana@ucb.edu.bo	\$2b\$12\$Qy3Em5jCwk9FFYyPHwnBO.GG4KE9Et46VjcgJ2s6Rxqvha6T/W...	true	false
4	4	maria@ucb.edu.bo	\$2b\$12\$W.DFJHtSF08hg8yaLUm0ie6mFolAqITTAThEwNHtCZXR91TaDzV...	true	false

- Caso 4: usuario 5, carlos@ucb.edu.bo.

GraphQL

Prettify Merge Copy History

```
1 mutation {
2   createUser(email: "carlos@ucb.edu.bo", password: "segura456") {
3     id
4     email
5     is_active
6   }
7 }
8
```

```
{
  "data": {
    "createUser": {
      "id": 5,
      "email": "carlos@ucb.edu.bo",
      "is_active": true
    }
  }
}
```

bi_cenenario/postgres@PostgreSQL 16* x

bi_cenenario/postgres@PostgreSQL 16

Query Query History

```
1 SELECT * FROM public.users
2 ORDER BY id ASC
3
4
5
```

Data Output Messages Notifications

	id [PK] integer	email character varying	hashed_password character varying	is_active boolean	is_admin boolean
1	1	joan.riveros@ucb.edu.bo	5e88482ed421a69c992032232d256f23	true	false
2	2	joan@ucb.edu.bo	\$2b\$12\$Uj1GsprBEn8ZXdw73vE.j.Q1e7OYB7oPMTdTxl7q6h/wDQTz/HCr2	true	false
3	3	ana@ucb.edu.bo	\$2b\$12\$Qy3Em5jCwk9FFYyPHwnBO.GG4KE9Et46VjcgJ2s6Rxqvha6T/WP2	true	false
4	4	maria@ucb.edu.bo	\$2b\$12\$W.DFJHtSF08hg8yaLUm0ie6mFolAqITTAThEwNHtCZXR91TaDzV72	true	false
5	5	carlos@ucb.edu.bo	\$2b\$12\$CMV1p1Rdhg5pl925VWHYF.WKHeESUKTug/R8ZhFWfyWDhSX3iJf...	true	false

- Caso 5: usuario 6, luciana@ucb.edu.bo.

```
GraphQL
mutation {
  createUser(email: "luciana@ucb.edu.bo", password: "lucianaPass789") {
    id
    email
    is_active
  }
}
```

```
{
  "data": {
    "createUser": {
      "id": 6,
      "email": "luciana@ucb.edu.bo",
      "is_active": true
    }
  }
}
```

bi_centenario/postgres@PostgreSQL 16*

bi_centenario/postgres@PostgreSQL 16

Query Query History

```
1 SELECT * FROM public.users
2 ORDER BY id ASC
3
4
5
```

Data Output Messages Notifications

	id [PK] integer	email character varying	hashed_password character varying	is_active boolean	is_admin boolean
1	1	joan.riveros@ucb.edu.bo	5e88482ed421a69c99203223d256f23	true	false
2	2	joan@ucb.edu.bo	\$2b\$12\$Uj1GsprBEn8ZXdw73vE.j.Q1e7OYB7oPMTdTxI7q6h/wDQTz/HCr2	true	false
3	3	ana@ucb.edu.bo	\$2b\$12\$Qy3Em5jCwk9FFYyPHwnBO.GG4KE9Et46VjcgJ2s6Rxqvhca6T/WP2	true	false
4	4	maria@ucb.edu.bo	\$2b\$12\$W.DFJHTSF08hg8yaLUm0ie6mFolAQiTtAThEwNHtCZXR91TaDzV72	true	false
5	5	carlos@ucb.edu.bo	\$2b\$12\$CMV1p1Rdhg5pi925VWHYF.WKHeESUKTug/R8ZhFWfyWDhSX3iJf...	true	false

- Caso 6: usuario 7, carla@ucb.edu.bo.

```
GraphQL
mutation {
  createUser(email: "carla@ucb.edu.bo", password: "carlaPass789") {
    id
    email
    is_active
  }
}
```

```
{
  "data": {
    "createUser": {
      "id": 7,
      "email": "carla@ucb.edu.bo",
      "is_active": true
    }
  }
}
```

bi_centenario/postgres@PostgreSQL 16* x

bi_centenario/postgres@PostgreSQL 16

Query Query History

```
1 SELECT * FROM public.users
2 ORDER BY id ASC
3
4
5
```

Data Output Messages Notifications

	id [PK] integer	email character varying	hashed_password character varying	is_active boolean	is_admin boolean
1	1	joan.riveros@ucb.edu.bo	5e88482ed421a69c992032232d256f23	true	false
2	2	joan@ucb.edu.bo	\$2b\$12\$Uj1GsprBEn8ZXdw73vE.j.Q1e7OYB7oPMTdTxI7q6h/wDQTz/HCr2	true	false
3	3	ana@ucb.edu.bo	\$2b\$12\$Qy3Em5jCwk9FFyPHwnBO.GG4KE9Et46VjcgJ2s6Rxqvha6T/W...	true	false
4	4	maria@ucb.edu.bo	\$2b\$12\$W.DFJHtSF08hg8yaLUm0ie6mFolAqITTAthEwNHtCZXR91TaDzV...	true	false
5	5	carlos@ucb.edu.bo	\$2b\$12\$CMV1p1Rdhg5pl925VWHYF.WKHeESUKTug/R8ZhFWfyWDhSX3i...	true	false
6	6	luciana@ucb.edu.bo	\$2b\$12\$akddJxh2S4TmkaygFFwbKOjytoapjo9yZP1tA0IW2TgH9KXkzlyS	true	false
7	7	carla@ucb.edu.bo	\$2b\$12\$RGKuS8ZnwmzoWlJMWuNaud9lrbXj49256qb42tKp2cG4uf3N4o...	true	false

- Error por duplicado
 - Caso: usuario, carla@ucb.edu.bo.

GraphiQL

```
1 mutation {
2   createUser(email: "carla@ucb.edu.bo", password: "carlaPass789") {
3     id
4     email
5     is_active
6   }
7 }
8
```

```
{
  "errors": [
    {
      "message": "Error al crear usuario",
      "locations": [
        {
          "line": 2,
          "column": 3
        }
      ],
      "path": [
        "createUser"
      ]
    }
  ],
  "data": {
    "createUser": null
  }
}
```

- Consola de Node.js con logs
 - Caso: Creación de usuarios.

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  GITLENS

  _values: [Array],
  _valueLookup: null,
  _nameLookup: null
}
},
_subTypeMap: [Object: null prototype] {},
_implementationsMap: [Object: null prototype] {}
}
Microservicio GraphQL funcionando en http://localhost:4000/graphql
Usuario Creado Correctamente, FASTAPI_URL: http://localhost:8000
```

- Caso: Usuario Existente.

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  GITLENS

  _valueLookup: null,
  _nameLookup: null
}
},
_subTypeMap: [Object: null prototype] {},
_implementationsMap: [Object: null prototype] {}
}
Microservicio GraphQL funcionando en http://localhost:4000/graphql
Usuario Creado Correctamente, FASTAPI_URL: http://localhost:8000
Usuario Creado Correctamente, FASTAPI_URL: http://localhost:8000
Error en createUser: { detail: 'Email ya registrado' }
```

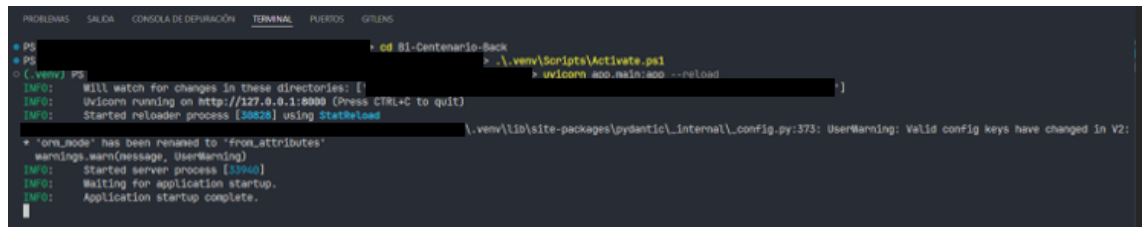
- Query de prueba _empty
 - query { _empty }

```
GraphQL  [Play]  [Prettify]  [Merge]  [Copy]  [History]

1 query {
2   _empty
3 }
4
query

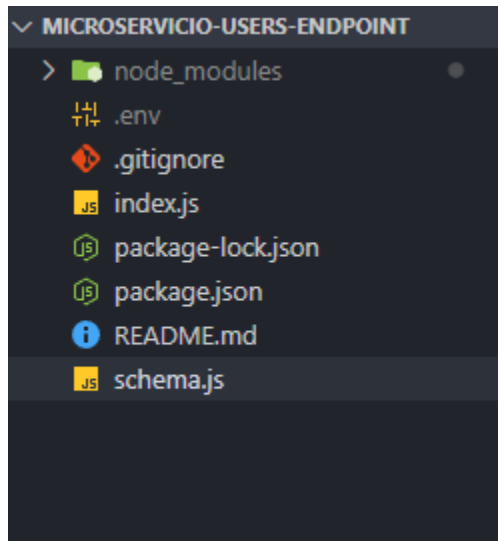
{
  "data": {
    "_empty": "Servidor activo ✓"
  }
}
```

- FastAPI ejecutándose
 - Consola de Visual Studio Code:



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  GRUPOS
• PS      i cd 81-Centenario-Back
• PS      p .\venv\scripts\Activate.ps1
o (.venv) PS      k uvicorn app:main:app --reload
INFO: Will watch for changes in these directories: []
INFO: uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reload process [30828] using StatReloader
* "orm_node" has been renamed to "from_attributes"
warnings.warn(message, UserWarning)
INFO: Started server process [13960]
INFO: Waiting for application startup.
INFO: Application startup complete.
```

- Estructura de archivos del proyecto
 - Árbol de archivos del proyecto:



8. Conclusión

El microservicio cumple con su propósito: permite crear usuarios desde un sistema externo mediante GraphQL, integrándose exitosamente con el backend FastAPI del proyecto principal.

Este desarrollo demuestra la capacidad de desacoplar funcionalidades específicas en microservicios reutilizables, utilizando tecnologías modernas, eficientes y robustas.

9. Repositorio del Proyecto

Link del repositorio de GitHub:

https://github.com/Joan-Riveros/Joan_Riveros_2daEvaluacionBackend