

**UNIVERSIDAD MAYOR, REAL Y PONTIFICIA DE
SAN FRANCISCO XAVIER DE CHUQUISACA**

FACULTAD DE TECNOLOGIA



PRACTICA EN CLASE 4

MATERIA: MICROSERVICIOS

UNIVERSITARIO: SOLORZANO ARANCIBIA DIEGO SANTIAGO
JANCKO GALLARDO PABLO ALEJANDRO

CARRERA: ING. EN CIENCIAS DE LA COMPUTACION

Ejemplos de consultas en GraphQL Playground

1. Crear una mesa

The screenshot shows the GraphQL Playground interface. The URL is `http://localhost:4000/graphql`. The operation is a mutation to create a table:

```
1 mutation {
2   createMesa(nro_mesa: "2001", nombre_escuela: "Escuela Técnica") {
3     id
4     nro_mesa
5   }
6 }
```

The response is:

```
{
  "data": {
    "createMesa": {
      "id": "2",
      "nro_mesa": "2001"
    }
  }
}
```

The response status is 200, with a response time of 34.0ms and a size of 53B. The bottom panel shows the 'Variables' tab with an empty object and a 'JSON' button.

2. Crear un padron

The screenshot shows the GraphQL Playground interface. The URL is `http://localhost:4000/graphql`. The operation is a mutation to create a padron:

```
1 mutation {
2   createPadron(
3     nombres: "Diego y Pablo"
4     apellidos: "Solorzano y Jancko"
5     numero_documento: "87654321"
6     fotografia: "foto.png"
7     mesaId: 2
8   ) {
9     id
10    nombres
11    mesa {
12      nro_mesa
13    }
14  }
15 }
```

The response is:

```
{
  "data": {
    "createPadron": {
      "id": "2",
      "nombres": "Diego y Pablo",
      "mesa": {
        "nro_mesa": "2001"
      }
    }
  }
}
```

The response status is 200, with a response time of 26.0ms and a size of 90B. The bottom panel shows the 'Variables' tab with an empty object and a 'JSON' button.

3. Consultar todos los padrones con su mesa

The screenshot shows the GraphQL Sandbox interface. The URL is `http://localhost:4000/graphql`. The query in the Operation tab is:

```
1 query {
2   getPadrones {
3     id
4     nombres
5     apellidos
6     numero_documento
7     mesa {
8       nro_mesa
9       nombre_escuela
10    }
11  }
12 }
```

The Response tab shows the JSON result:

```
{
  "data": {
    "getPadrones": [
      {
        "id": "2",
        "nombres": "Diego y Pablo",
        "apellidos": "Solorzano y Jancko",
        "numero_documento": "87654321",
        "mesa": {
          "nro_mesa": "2001",
          "nombre_escuela": "Escuela Técnica"
        }
      }
    ]
  }
}
```

Below the query editor, there are tabs for Variables, Headers, Pre-Operation Script, and Post-Operation Script. The Variables tab is active, showing a single variable with the value `1`. A button labeled `JSON` is visible next to the variable.

4. Consultar mesas con sus padrones

The screenshot shows the GraphQL Sandbox interface. The URL is `http://localhost:4000/graphql`. The query in the Operation tab is:

```
1 query {
2   getMesas {
3     nro_mesa
4     nombre_escuela
5     padrones {
6       nombres
7       apellidos
8     }
9   }
10 }
```

The Response tab shows the JSON result:

```
{
  "data": {
    "getMesas": [
      {
        "nro_mesa": "2001",
        "nombre_escuela": "Escuela Técnica",
        "padrones": [
          {
            "nombres": "Diego y Pablo",
            "apellidos": "Solorzano y Jancko"
          }
        ]
      }
    ]
  }
}
```

Below the query editor, there are tabs for Variables, Headers, Pre-Operation Script, and Post-Operation Script. The Variables tab is active, showing a single variable with the value `1`. A button labeled `JSON` is visible next to the variable.

Ejercicio de Sistema de Biblioteca

a. Esquema GraphQL

o Define los tipos Libro y Prestamo.

```
Libro.js U Prestamo.js U typeDefs.js U resolvers.js U index.js U
biblioteca > src > entity > Prestamo.js > <unknown> > columns > fecha_devolucion
1 const { EntitySchema } = require("typeorm");
2
3 module.exports = new EntitySchema({
4   name: "Prestamo",
5   tableName: "prestamo",
6   columns: {
7     id: {
8       type: Number,
9       primary: true,
10      generated: true,
11    },
12    usuario: {
13      type: String,
14    },
15    fecha_prestamo: {
16      type: String, // puedes usar Date si prefieres
17    },
18    fecha_devolucion: {
19      type: String,
20    },
21  },
22  relations: {
23    libro: {
24      type: "many-to-one",
25      target: "Libro",
26      joinColumn: true,
27      eager: true,
28    },
29  },
30 });
```

```
Libro.js U Prestamo.js U typeDefs.js U resolvers.js U index.js
biblioteca > src > entity > Libro.js > <unknown> > columns > id
1 const { EntitySchema } = require("typeorm");
2
3 module.exports = new EntitySchema({
4   name: "Libro",
5   tableName: "libro",
6   columns: {
7     id: {
8       type: Number,
9       primary: true,
10      generated: true,
11    },
12    titulo: {
13      type: String,
14    },
15    autor: {
16      type: String,
17    },
18    isbn: {
19      type: String,
20    },
21    anio_publicacion: {
22      type: Number,
23    },
24  },
25  relations: {
26    prestamos: {
27      type: "one-to-many",
28      target: "Prestamo",
29      inverseSide: "libro",
30    },
31  },
32 });
```

o Define consultas (Query) para:

- Obtener todos los libros (getLibros).

```
biblioteca > src > schema > resolvers.js > resolvers
1 const { AppDataSource } = require("../data-source");
2 const Libro = require("../entity/Libro");
3 const Prestamo = require("../entity/Prestamo");
4
5 const resolvers = {
6   Query: {
7     getLibros: async () => {
8       return await AppDataSource.getRepository(Libro).find({ relations: ["prestamos"] });
9     },
10  },
11};
```

```
const typeDefs = gql`
  type Libro {
    id: ID!
    titulo: String!
    autor: String!
    isbn: String!
    anio_publicacion: Int!
    prestamos: [Prestamo!]
  }
`
```

```
type Query {
  getLibros: [Libro!]
}
```

- Obtener todos los préstamos (getPrestamos).

```
type Prestamo {
  id: ID!
  usuario: String!
  fecha_prestamo: String!
  fecha_devolucion: String!
  libro: Libro!
}
```

```
getPrestamos: async () => {
  return await AppDataSource.getRepository(Prestamo).find({ relations: ["libro"] });
},
```

```
getPrestamos: [Prestamo!]
```

- Obtener un préstamo por su id (getPrestamoById).

```
getPrestamoById(id: ID!): Prestamo
```

```
getPrestamoById: async (_, { id }) => {
  return await AppDataSource.getRepository(Prestamo).findOne({
    where: { id },
    relations: ["libro"],
  });
},
```

o Define mutaciones (Mutation) para:

- Crear un libro (createLibro).

```
type Mutation {
  createLibro(
    titulo: String!
    autor: String!
    isbn: String!
    anio_publicacion: Int!
  ): Libro
}
```

```
Mutation: {
  createLibro: async (_, { titulo, autor, isbn, anio_publicacion }) => {
    const repo = AppDataSource.getRepository(Libro);
    const libro = repo.create({ titulo, autor, isbn, anio_publicacion });
    return await repo.save(libro);
  },
}
```

- Registrar un préstamo (createPrestamo).

```
createPrestamo(
  usuario: String!
  fecha_prestamo: String!
  fecha_devolucion: String!
  libroId: ID!
): Prestamo
```

```
createPrestamo: async (
  _,
  { usuario, fecha_prestamo, fecha_devolucion, libroId }
) => {
  const repoPrestamo = AppDataSource.getRepository(Prestamo);
  const repoLibro = AppDataSource.getRepository(Libro);

  const libro = await repoLibro.findOneBy({ id: libroId });
  if (!libro) throw new Error("Libro no encontrado");

  const prestamo = repoPrestamo.create({
    usuario,
    fecha_prestamo,
    fecha_devolucion,
    libro,
  });

  return await repoPrestamo.save(prestamo);
},
```

b. Consultas GraphQL (Queries)

o Listar todos los préstamos con el libro correspondiente.

Operation

1

2

3

4

5

6

7

8

9

10

11

12

13

```
query {
  getPrestamos {
    id
    usuario
    fecha_prestamo
    fecha_devolucion
    libro {
      titulo
      autor
    }
  }
}
```

...

Run

Response

200

7.00ms

205B

205B

7.00ms

200

```
{
  "data": {
    "getPrestamos": [
      {
        "id": "1",
        "usuario": "María López",
        "fecha_prestamo": "2025-09-04",
        "fecha_devolucion": "2025-09-20",
        "libro": {
          "titulo": "Cien Años de Soledad",
          "autor": "Gabriel García Márquez"
        }
      }
    ]
  }
}
```

Practica 4 / Obtener todos los prestamos

Save

Share

POST

http://localhost:4000/graphql

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "query": "{ getPrestamos { id usuario fecha_prestamo fecha_devolucion libro { id titulo autor } } }"
3 }
4
```

Body

Cookies

Headers (8)

Test Results

200 OK

11 ms

669 B

Save Response

JSON

Preview

Visualize

```
1 {
2   "data": {
3     "getPrestamos": [
4       {
5         "id": "1",
6         "usuario": "María López",
7         "fecha_prestamo": "2025-09-04",
8         "fecha_devolucion": "2025-09-20",
9         "libro": {
10          "id": "1",
11          "titulo": "Cien Años de Soledad",
12          "autor": "Gabriel García Márquez"
13        }
14      }
15    ]
16  }
17 }
```


o Listar todos los libros junto con el historial de préstamos realizados.

Operation

1

2

3

4

5

6

7

8

9

10

11

```
query {
  getLibros {
    id
    titulo
    prestamos {
      usuario
      fecha_prestamo
    }
  }
}
```

...

Run

Response

200

20.0ms

141B

```
{
  "data": {
    "getLibros": [
      {
        "id": "1",
        "titulo": "Cien Años de Soledad",
        "prestamos": [
          {
            "usuario": "María López",
            "fecha_prestamo": "2025-09-04"
          }
        ]
      }
    ]
  }
}
```

Practica 4 / Obtener todos los libros

Save

Share

POST

http://localhost:4000/graphql

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Cookies

Beautify

```
1 {
2   "query": "{ getLibros { id titulo autor isbn anio_publicacion prestamos { id usuario fecha_prestamo fecha_devolucion } } }"
3 }
4
```

Body

Cookies

Headers (8)

Test Results

200 OK

17 ms

773 B

Save Response

{ JSON

Preview

Visualize

```
1 {
2   "data": {
3     "getLibros": [
4       {
5         "id": "1",
6         "titulo": "Cien Años de Soledad",
7         "autor": "Gabriel García Márquez",
8         "isbn": "9780307474728",
9         "anio_publicacion": 1967,
10        "prestamos": [
11          {
12            "id": "1",
13            "usuario": "María López",
14            "fecha_prestamo": "2025-09-04"
```

c. Mutaciones GraphQL

o Crear un libro nuevo:

- **titulo:** Cien Años de Soledad
- **autor:** Gabriel García Márquez
- **isbn:** 9780307474728
- **anio_publicacion:** 1967.

Operation

Run

```
1 mutation {
2   createLibro(
3     titulo: "Cien Años de Soledad"
4     autor: "Gabriel García Márquez"
5     isbn: "9780307474728"
6     anio_publicacion: 1967
7   ) {
8     id
9     titulo
10  }
11 }
```

Response

200 37.0ms 69B

```
{
  "data": {
    "createLibro": {
      "id": "1",
      "titulo": "Cien Años de Soledad"
    }
  }
}
```

Practica 4 / Crear Libro

Save Share

POST http://localhost:4000/graphql Send

Params Authorization Headers (8) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

cookies Beautify

```
1 {
2   "query": "mutation { createLibro(titulo: \"Cien Años de Soledad\", autor: \"Gabriel García Márquez\", isbn: \"9780307474728\",
3     anio_publicacion: 1967) { id titulo autor isbn anio_publicacion } }"
4 }
```

Body Cookies Headers (8) Test Results 200 OK 32 ms 419 B Save Response

{ JSON Preview Visualize

```
1 {
2   "data": {
3     "createLibro": {
4       "id": "2",
5       "titulo": "Cien Años de Soledad",
6       "autor": "Gabriel García Márquez",
7       "isbn": "9780307474728",
8       "anio_publicacion": 1967
9     }
10  }
11 }
```


o Crear un préstamo con los siguientes datos:

- usuario: María López
- fecha_prestamo: 2025-09-04
- fecha_devolucion: 2025-09-20
- libroId: 1 (libro recién creado).

Operation

Run

```
1 mutation {
2   createPrestamo(
3     usuario: "María López"
4     fecha_prestamo: "2025-09-04"
5     fecha_devolucion: "2025-09-20"
6     libroId: 1
7   ) {
8     id
9     usuario
10    libro {
11      titulo
12    }
13  }
14 }
```

Response

200 20.0ms 108B

```
{
  "data": {
    "createPrestamo": {
      "id": "1",
      "usuario": "María López",
      "libro": {
        "titulo": "Cien Años de Soledad"
      }
    }
  }
}
```

Practica 4 / Crear Prestamo

Save Share

POST http://localhost:4000/graphql Send

Params Authorization Headers (8) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "query": "mutation { createPrestamo(usuario: \"María López\", fecha_prestamo: \"2025-09-04\", fecha_devolucion: \"2025-09-20\", libroId: 1) { id usuario libro { titulo } fecha_prestamo fecha_devolucion } }"
3 }
4
```

Body Cookies Headers (8) Test Results

200 OK 30 ms 438 B Save Response

JSON Preview Visualize

```
1 {
2   "data": {
3     "createPrestamo": {
4       "id": "2",
5       "usuario": "María López",
6       "libro": {
7         "titulo": "Cien Años de Soledad"
8       },
9       "fecha_prestamo": "2025-09-04",
10      "fecha_devolucion": "2025-09-20"
11    }
12  }
13 }
```

d. Desafío extra

o Crea una consulta que devuelva los préstamos filtrados por nombre de usuario.

Operation

1

2

3

4

5

6

7

8

9

10

```
query {
  getPrestamosByUsuario(usuario: "María López") {
    id
    fecha_prestamo
    libro {
      titulo
    }
  }
}
```

Run

Response

20014.0ms121B

```
{
  "data": {
    "getPrestamosByUsuario": [
      {
        "id": "1",
        "fecha_prestamo": "2025-09-04",
        "libro": {
          "titulo": "Cien Años de Soledad"
        }
      }
    ]
  }
}
```

Practica 4 / Filtrar prestamo por usuario

Save

Share

POST

http://localhost:4000/graphql

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

4

```
{
  "query": "{ getPrestamosByUsuario(usuario: \"María López\") { id fecha_prestamo fecha_devolucion libro { titulo autor } } }"
```

Body

Cookies

Headers (8)

Test Results

200 OK

8 ms

608 B

Save Response

JSON

Preview

Visualize

1

2

3

4

5

6

7

8

9

10

11

12

13

14

```
{
  "data": {
    "getPrestamosByUsuario": [
      {
        "id": "1",
        "fecha_prestamo": "2025-09-04",
        "fecha_devolucion": "2025-09-20",
        "libro": {
          "titulo": "Cien Años de Soledad",
          "autor": "Gabriel García Márquez"
        }
      }
    ]
  }
}
```