

Documento de Ejemplos de Consultas (Querys)

Introducción

Este documento sirve como una guía práctica y de referencia para los desarrolladores y usuarios del sistema. Se presenta una colección de ejemplos de consultas (querys) que ilustran cómo interactuar con la base de datos y el API del sistema para realizar tareas comunes. Estos ejemplos abarcan una variedad de casos de uso, desde consultas básicas hasta operaciones más complejas, y están diseñados para proporcionar una comprensión clara y concisa de cómo utilizar eficientemente nuestras interfaces de programación.

Al realizar operaciones de mutación El primer argumento en la mutación siempre debe ser el id del elemento que se va a modificar. Este id no se altera durante la operación de edición. Tras especificar el id, se deben incluir los campos y valores que se desean actualizar. Estos campos son los que efectivamente se modificarán en la base de datos.

Querys

buscarLibrosTitulo(titulo: String): [Libro]

```
query {  
  buscarLibrosTitulo(titulo: "El ritmo de la guerra"){  
    id  
    titulo  
    autor{nombre}  
    categoria{nombre}  
  }  
}
```

buscarTodosLibros: [Libro]

```
query {  
  buscarTodosLibros{  
    id  
    titulo  
    autor{  
      id  
      nombre  
    }  
    categoria{  
      id  
      nombre  
    }  
    disponible  
  }  
}
```

```
}
```

obtenerAutor(id: ID!): Autor

```
query {  
  obtenerAutor(id: 10000){  
    nombre  
  }  
}
```

obtenerTodosAutores: [Autor]

```
query {  
  obtenerTodosAutores{  
    id  
    nombre  
  }  
}
```

obtenerCategoria(id: ID!): Categoria

```
query {  
  obtenerCategoria(id: 10001){  
    id  
    nombre  
  }  
}
```

obtenerTodasCategorias: [Categoria]

```
query {  
  obtenerTodasCategorias {  
    id  
    nombre  
  }  
}
```

obtenerReservas(usuarioId: ID!): [Reserva]

obtenerTodasReservas: [Reserva]

Mutation

agregarLibro(titulo: String!, autorId: ID!, categoriaId: ID!): Libro

```
mutation {  
  agregarLibro(titulo: "Juramentada", autorId: 10000, categoriaId: 10001 ) {  
    id  
    titulo  
  }  
}
```

editarLibro(id: ID!, titulo: String, autorId: ID, categoriaId: ID, disponible: Boolean): Libro

El id se pondrá el original no se podrá modificar

Valores que se modificaran

```
mutation {  
  editarLibro(id: 10004, titulo: "Esquirla del amanecer", autorId: 10000,  
    categoriaId: 10001, disponible: false){  
    id  
    titulo  
  }  
}
```

eliminarLibro(id: ID!): Boolean

```
mutation {  
  eliminarLibro(id: 10004)  
}
```

agregarAutor(nombre: String!): Autor

```
mutation {  
  agregarAutor(nombre: "Brandon Sanderson")  
  {  
    id  
    nombre  
  }  
}
```

editarAutor(id: ID!, nombre: String): Autor

```
mutation {  
  editarAutor(id: 10001, nombre: "Brandon Sanderson"){  
    id  
    nombre  
  }  
}
```

eliminarAutor(id: ID!): String

```
mutation {  
  eliminarAutor(id: 10005)  
}
```

agregarCategoria(nombre: String!): Categoria

```
mutation {  
  agregarCategoria(nombre: "miego"){  
    nombre  
  }  
}
```

editarCategoria(id: ID!, nombre: String): Categoria

```
mutation{  
  editarCategoria(id:10001, nombre: "Accion"){  
    nombre  
  }  
}
```

```
}
```

eliminarCategoria(id: ID!): Boolean

```
mutation{
```

```
  eliminarCategoria(id: 10006)
```

```
}
```

hacerReserva(libroId: ID!, usuarioId: ID!): Reserva

cancelarReserva(id: ID!): Boolean