

Víctor Asenjo Carvajal  
Alexandru Dumitru Maroz  
Peilin Ni  
Alvaro Rodríguez Rubio

# Víctor Asenjo Carvajal

# Alexandru Dumitru Maroz

# Peilin Ni

# Alvaro Rodríguez Rubio

# CONTENIDOS DE LA PRESENTACIÓN

---

¿Qué es?

01



Evolución histórica

02



Características relevantes

03



04

Ejemplos de uso



05

Puntos fuertes



06

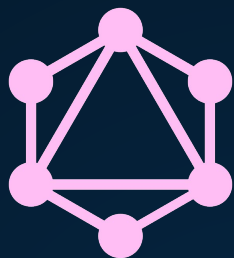
Puntos débiles



## ¿QUÉ ES?

---

GraphQL es un lenguaje de código abierto para consultar bases de datos desde aplicaciones del lado del cliente.

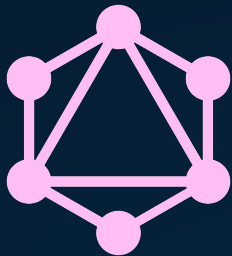


# ¿QUÉ ES?

---

# ¿QUÉ ES?

---

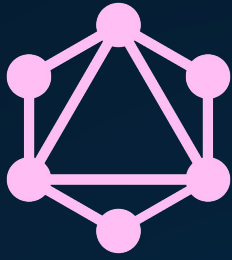


CLIENTE

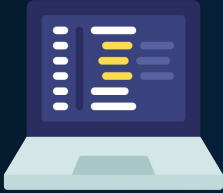


Lenguaje de consultas

# ¿QUÉ ES?

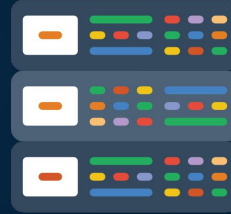


CLIENTE



Lenguaje de consultas

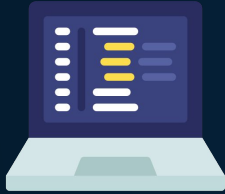
SERVIDOR



Entorno de ejecución  
para servir las consultas

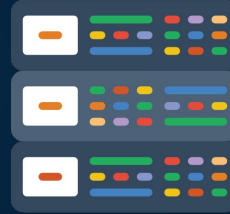
# ¿QUÉ ES?

CLIENTE



Lenguaje de consultas

SERVIDOR



Entorno de ejecución  
para servir las consultas



Base de datos (SQL, NoSQL), API Rest, GraphQL

# EVOLUCIÓN HISTÓRICA

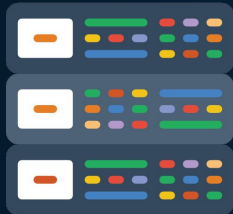
---





# ¿CÓMO FUNCIONA?

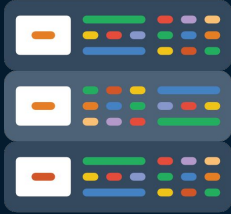
---



1

Describe tus datos en backend usando un schema y un sistema de tipos.

# ¿CÓMO FUNCIONA?



1

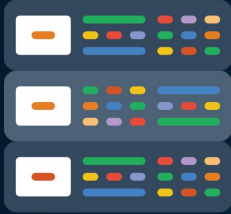
Describe tus datos en backend usando un schema y un sistema de tipos.



2

El cliente hace la petición de con el lenguaje de consultas GraphQL

# ¿CÓMO FUNCIONA?



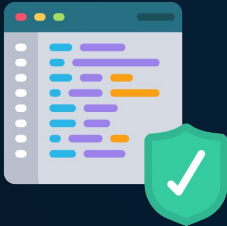
1

Describe tus datos en backend usando un schema y un sistema de tipos.



2

El cliente hace la petición de con el lenguaje de consultas GraphQL



3

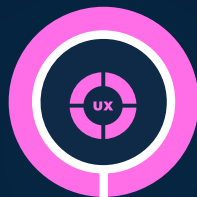
El cliente obtiene exactamente lo que necesita (ni más, ni menos).

# CARACTERÍSTICAS RELEVANTES

---



PERSONALIZACIÓN



ESCALABILIDAD



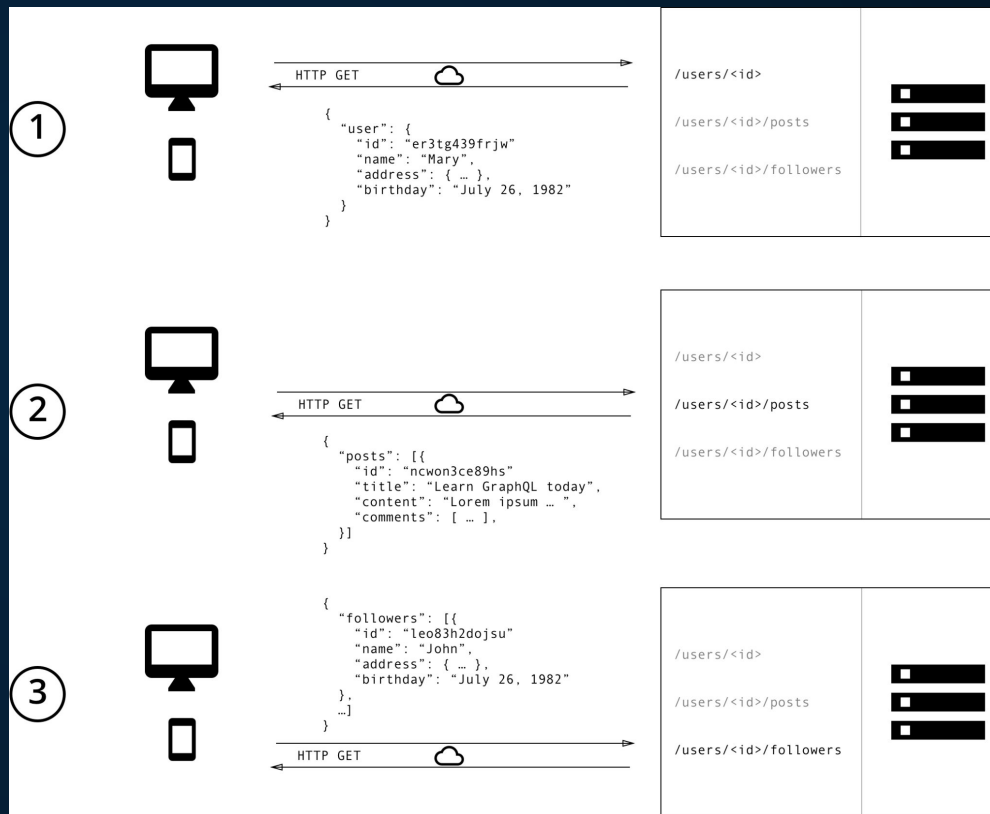
COMPATIBILIDAD

# PERSONALIZACIÓN



La API de GraphQL permite crear esquemas para agrupar distintas tablas de datos y poderlas consultar al unísono. Por la parte del cliente, solo recibe la información más relevante para él, que haya preseleccionado al hacer la consulta.

## REST API



# PERSONALIZACIÓN



PERSONALIZACIÓN

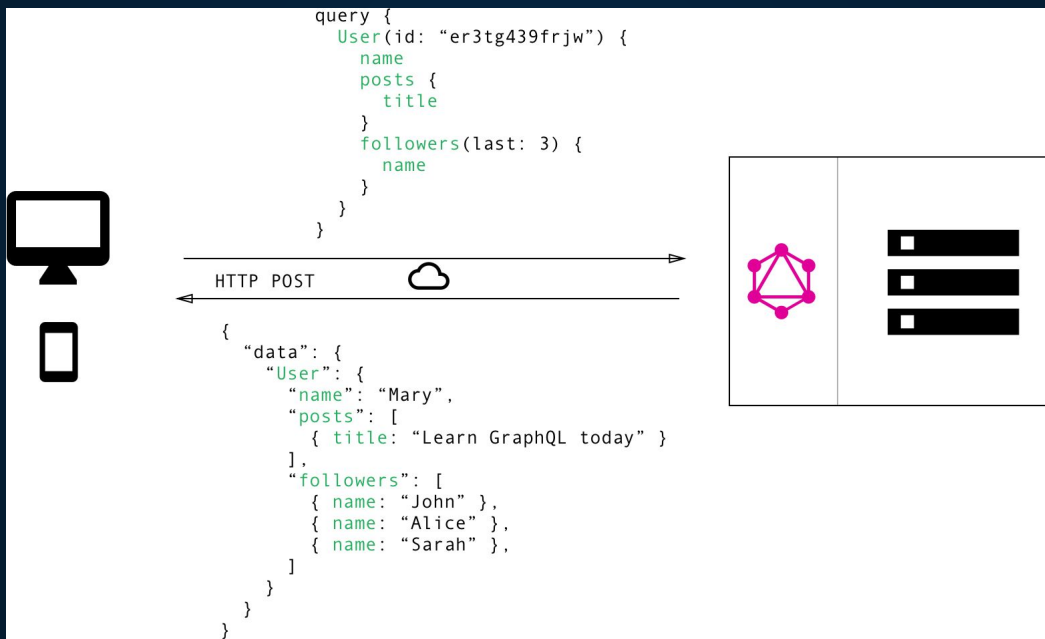


ESCALABILIDAD



COMPATIBILIDAD

GRAPHQL





Por la naturaleza del funcionamiento de GraphQL, es indiferente si vamos añadiendo más tablas o variables al esquema, lo que provoca que las querries antiguas se mantengan en funcionamiento.

# ESCALABILIDAD

```
type Film {  
  title: String  
  episode: Int  
  releaseDate: String  
  
}
```

```
type Film {  
  title: String  
  episode: Int  
  releaseDate: String  
  openingCrawl: String  
+ director: String  
  
}
```

```
type Film {  
  title: String  
  episode: Int  
  releaseDate: String  
+ openingCrawl: String  
  
}
```

# ESCALABILIDAD



PERSONALIZACIÓN



ESCALABILIDAD



COMPATIBILIDAD

```
type Film {  
  title: String  
  episode: Int  
  releaseDate: String  
  openingCrawl: String  
  director: String  
+ directedBy: Person  
}  
  
+ type Person {  
+   name: String  
+   directed: [Film]  
+   actedIn: [Film]  
+ }
```

```
type Film {  
  title: String  
  episode: Int  
  releaseDate: String  
  openingCrawl: String  
- director: String  
  directedBy: Person  
}  
  
type Person {  
  name: String  
  directed: [Film]  
  actedIn: [Film]  
}
```

```
type Film {  
  title: String  
  episode: Int  
  releaseDate: String  
  openingCrawl: String  
  director: String @deprecated  
  directedBy: Person  
}  
  
type Person {  
  name: String  
  directed: [Film]  
  actedIn: [Film]  
}
```





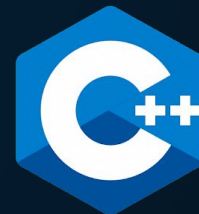
# COMPATIBILIDAD



PYTHON



GraphQL es compatible con gran variedad de lenguajes de programación o bases de datos. Por lo tanto, se puede adaptar a cualquier sistema ya existente.





# PUNTOS FUERTES



**Agrega nuevos campos sin  
afectar a versiones anteriores**



**Datos requeridos**



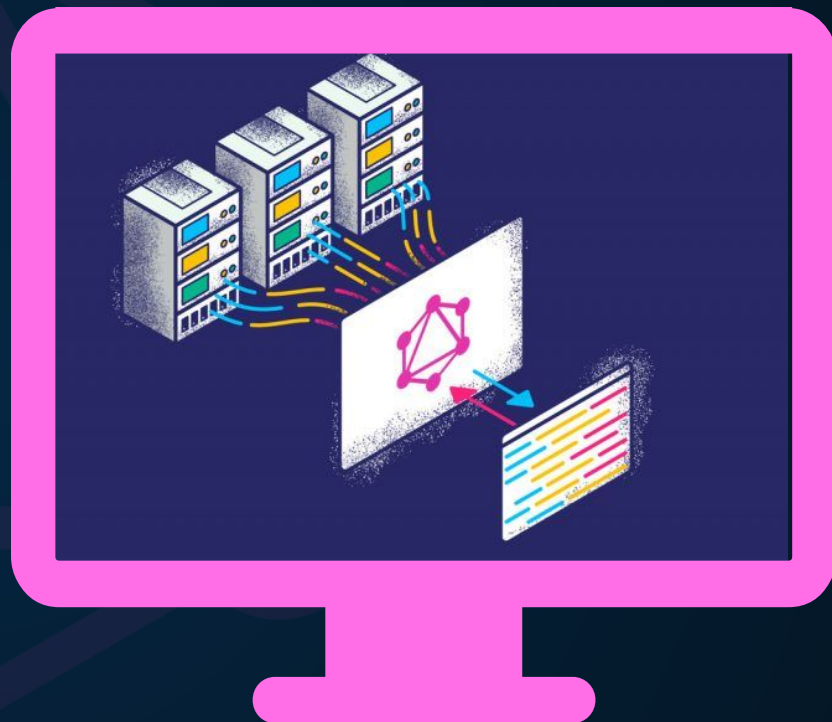
**Independiente de la base de datos**



**Flexible y eficiente**



**Añade alias a las variables**



# PUNTOS DÉBILES



**Formato limitado: respuestas en formato JSON**



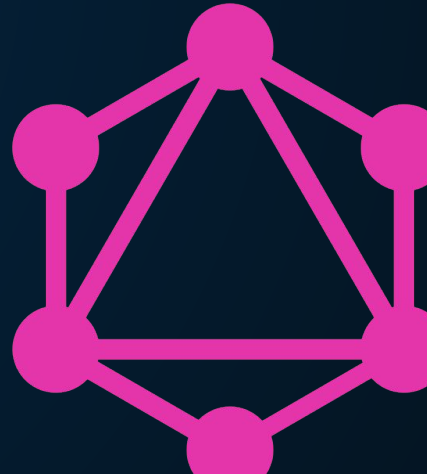
**Errores difíciles de gestionar**

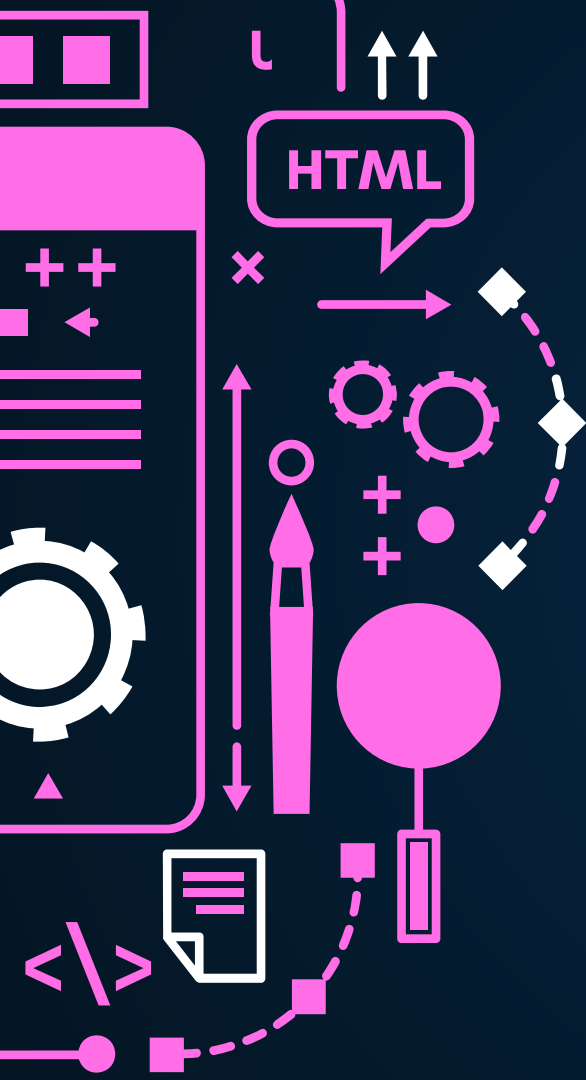


**Almacenamiento en caché muy complejo**



**Dificultad al limitar peticiones**





**¡Gracias por  
vuestra atención!**

# REFERENCIAS

- <https://graphql.org/>
- <https://medium.com/hackteam/graphql-will-do-to-rest-what-json-did-to-xml-7f060a75cc24>
- <https://docs.github.com/es/graphql/guides/introduction-to-graphql>
- <https://www.redhat.com/es/topics/api/what-is-graphql>
- <https://t.ly/nUlo>
- <https://t.ly/l6Hh>
- <https://t.ly/1jjP>
- <https://t.ly/XLR1>

# El equipo

---



**Alexandru Dumitru Maroz**

Ha hecho:

- Características
- Ejemplos de uso



**Victor Asenjo Carbajal**

Ha hecho:

- ¿Que es?
- ¿Cómo funciona?
- Evolución histórica



**Peilin Ni**

Ha hecho:

- Puntos débiles
- Investigar características



**Alvaro Rodríguez Rubio**

Ha hecho:

- Puntos fuertes
- El equipo