

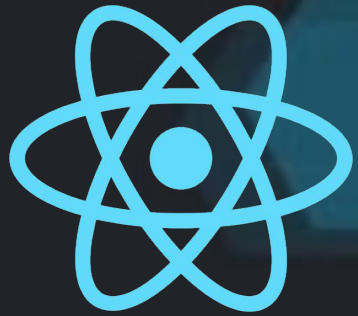


Integrando consultas entre un  
Front End de React y un Back End  
de Node

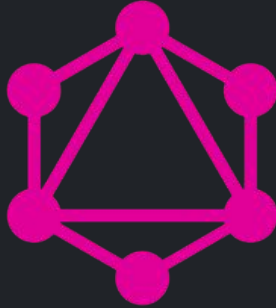
# SERGIO GARZÓN



Desarrollador de Software- Desarrollador de Videojuegos - Docente de Programación



React




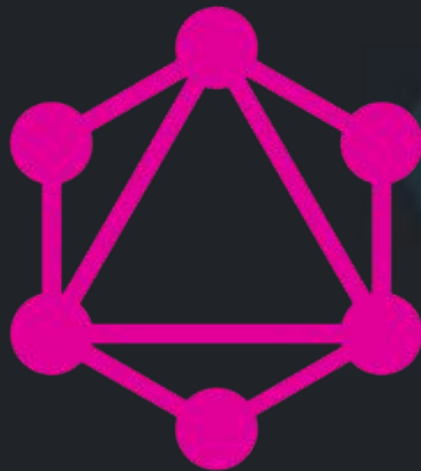
GraphQL



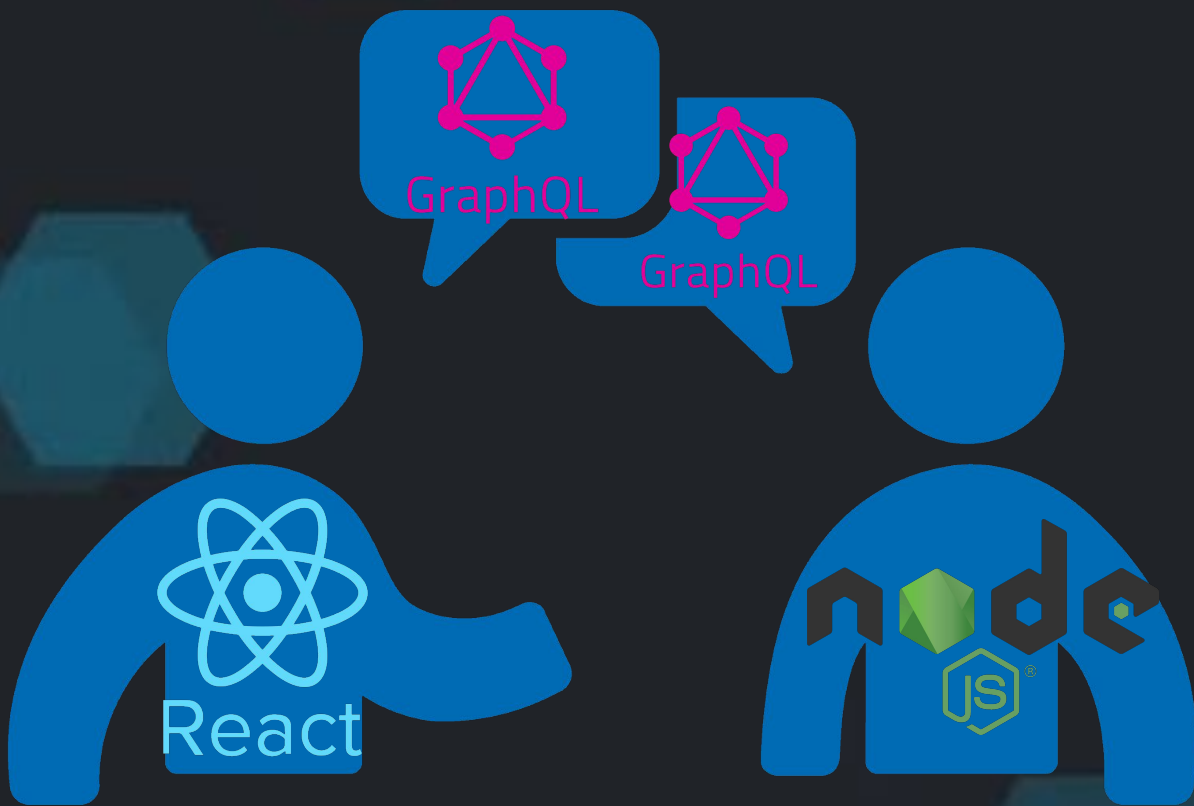
Cómo lograr una integración efectiva entre ambas capas, con GraphQL, para optimizar la comunicación y estructura de nuestras aplicaciones.

## GraphQL:

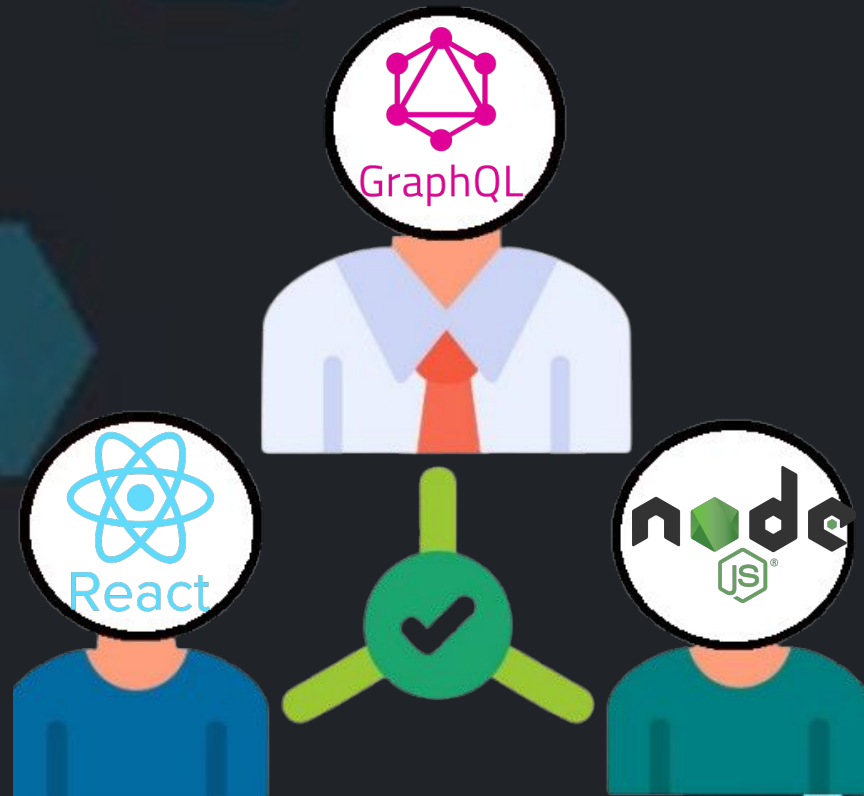
- Desarrollado por  (2015)
- Código abierto (Open source)
- Lenguaje de consultas para lectura y mutación de datos en APIs
- Alternativa de REST API
- Graph (Grafo) QL (Lenguaje de consulta)
- Es declarativo



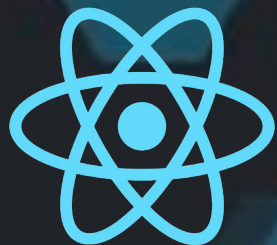
# GraphQL



GraphQL es un lenguaje de comunicación



GraphQL es un intermediario



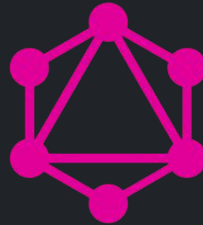
React



Cliente: Puede ser cualquier framework o librería

## GraphQL:

- El cliente especifica que que el servidor devuelva lo que necesite.
- Reduce over-fetching (muchos datos) y under-fetching (pocos datos)



GraphQL



## GraphQL:

- Especificamos y nos devuelve de manera correcta lo que queremos
- Una aplicación puede devolver id, título, fecha de finalización, pero podemos decirle que no devuelva la fecha de finalización.

```
{  
  aplicacion {  
    id  
    titulo  
  }  
}
```

```
{  
  "datos": {  
    "aplicacion": [{  
      "id": 1,  
      "titulo": "BeerJS"  
    },  
    {  
      "id": 1,  
      "titulo": "BeerJS 100"  
    }  
  ]  
}
```

## REST API:

- El cliente hace una solicitud get (por ejemplo a /usuarios) y trae todo.
- Obviamente está más estandarizado, es más fácil de entender.

{REST API}

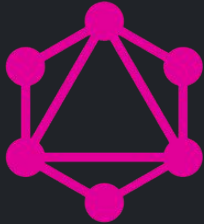
## Transferencia de datos:

Ambos utilizan JSON para la transferencia de datos

A stylized graphic of the text "{JSON}" in blue. The letter 'O' is replaced by a 3D sphere with a black and white gradient, giving it a metallic or glossy appearance. The entire graphic is centered on a dark blue background with faint, larger hexagonal shapes in the background.

## Mutaciones para modificación de datos

- Es mucho mejor en GraphQL (más flexibles, más eficientes que REST API)
- Único endpoint



GraphQL

{REST API}

# Mutaciones en GraphQL

Sin devolución de información

```
mutation {  
  crearUsuario ( nombre : "valor" , email : "valor email") {  
    id  
    nombre  
    email  
  }  
}
```

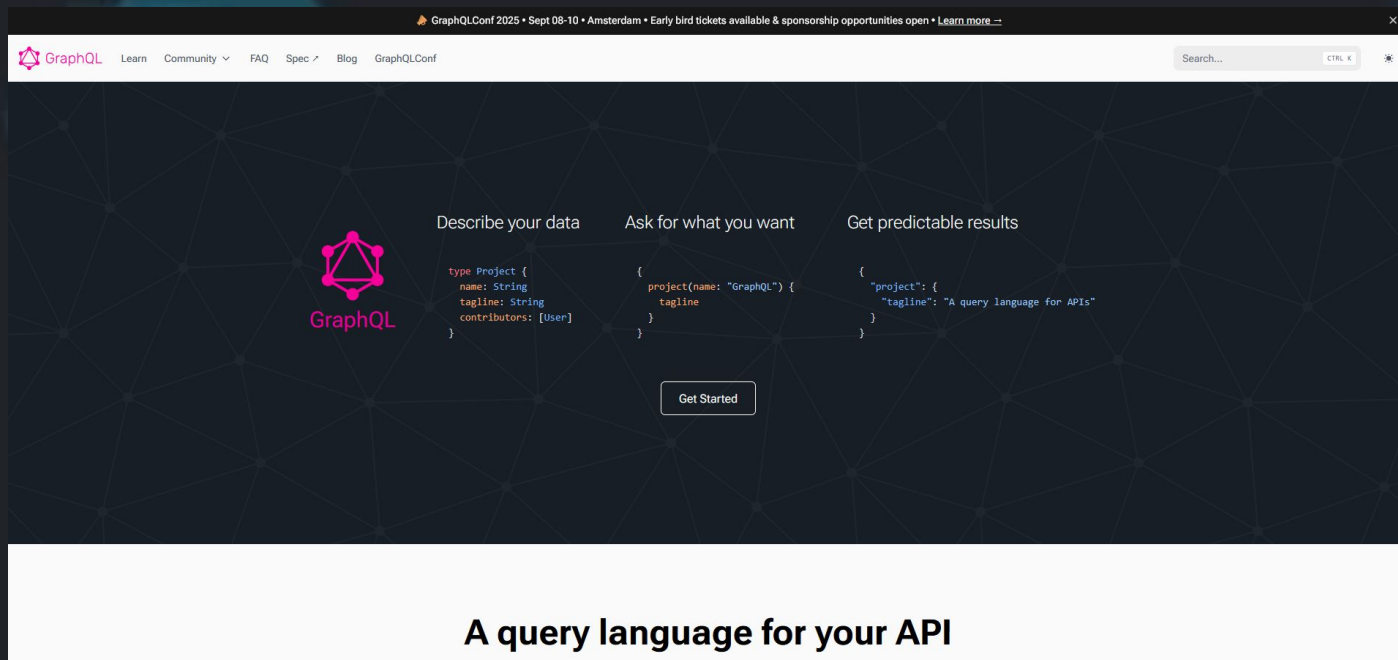
# Mutaciones en GraphQL

Con devolución de información

```
{  
  "datos" : {  
    "crearUsuario" : {  
      "id" : 1,  
      "nombre" : "valor",  
      "email" : "valor email"  
    }  
  }  
}
```

# GraphQL:


Sitio web: <https://graphql.org/>



# GraphQL:

Para instalar GraphQL, en la página tenemos el comando “npm install graphql –save”

GraphQLConf 2025 • Sept 08-10 • Amsterdam • Early bird tickets avail

 GraphQL

LearnCommunity ▾FAQSpec ↗BlogGraphQLConf

**GraphQL.js Tutorial**

Getting Started

Running Express + GraphQL

GraphQL Clients

Basic Types

Passing Arguments

Object Types

Mutations and Input Types

Authentication & Middleware

**Advanced Guides**

Constructing Types

**API Reference**

GraphQL.js Tutorial > Getting Started

## Getting Started With GraphQL.js

### Prerequisites

Before getting started, you should have Node v6 installed, although the examples sh  
guide, we won't use any language features that require transpilation, but we will use  
[functions](#), so if you aren't familiar with them you might want to read up on them first

To create a new project and install GraphQL.js in your current directory:

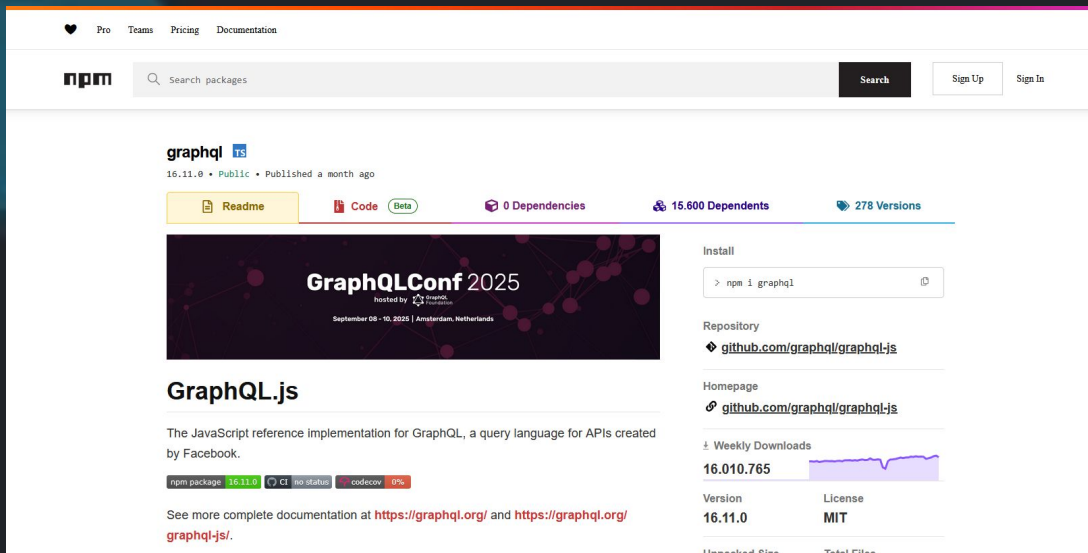
```
npm init
npm install graphql --save
```

### Writing Code



# GraphQL:

También podemos encontrarlo en la página de npm (<https://www.npmjs.com/>)



The screenshot shows the npm page for the **graphql** package. At the top, there's a navigation bar with links for Pro, Teams, Pricing, and Documentation. Below this is the npm logo and a search bar. The package name **graphql** is displayed with a TypeScript (TS) badge. Below the name, it says "16.11.0 • Public • Published a month ago". There are buttons for "Readme", "Code" (with a "Beta" badge), "Dependencies" (0), "Dependents" (15,600), and "Versions" (278). A large banner for "GraphQLConf 2025" is featured, hosted by GraphQL Foundation, from September 08 - 10, 2025, in Amsterdam, Netherlands. Below the banner, the title "GraphQL.js" is shown, followed by the description: "The JavaScript reference implementation for GraphQL, a query language for APIs created by Facebook." There are badges for "npm package 16.11.0", "CI (no status)", and "codecov 100%". Links to documentation are provided: "https://graphql.org/" and "https://graphql.org/graphql-js/". On the right side, there's an "Install" section with a command box showing `> npm i graphql`. Below that, the "Repository" is listed as [github.com/graphql/graphql-js](https://github.com/graphql/graphql-js), and the "Homepage" is also the same. A "Weekly Downloads" chart shows 16,010,765 downloads. A table lists the "Version" as 16.11.0 and the "License" as MIT. At the bottom, there are labels for "Unpacked Size" and "Total Files".



npm install graphql

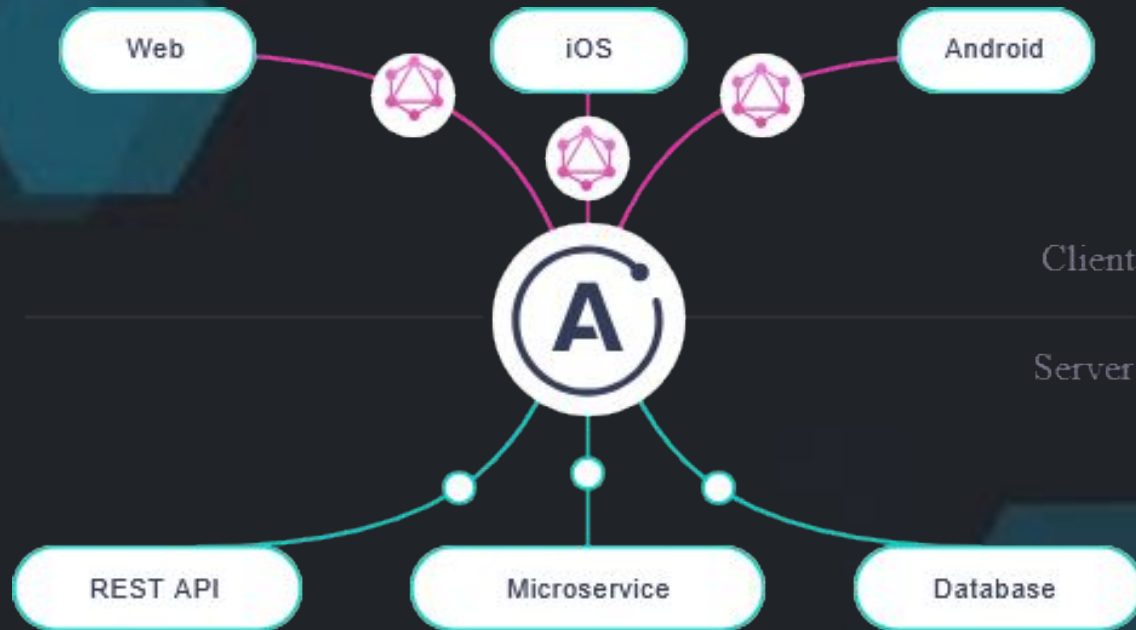
## Apollo-Client:

Podemos instalar varias librerías para que nos ayuden a consumir la API GraphQL, la más popular es Apollo-Client



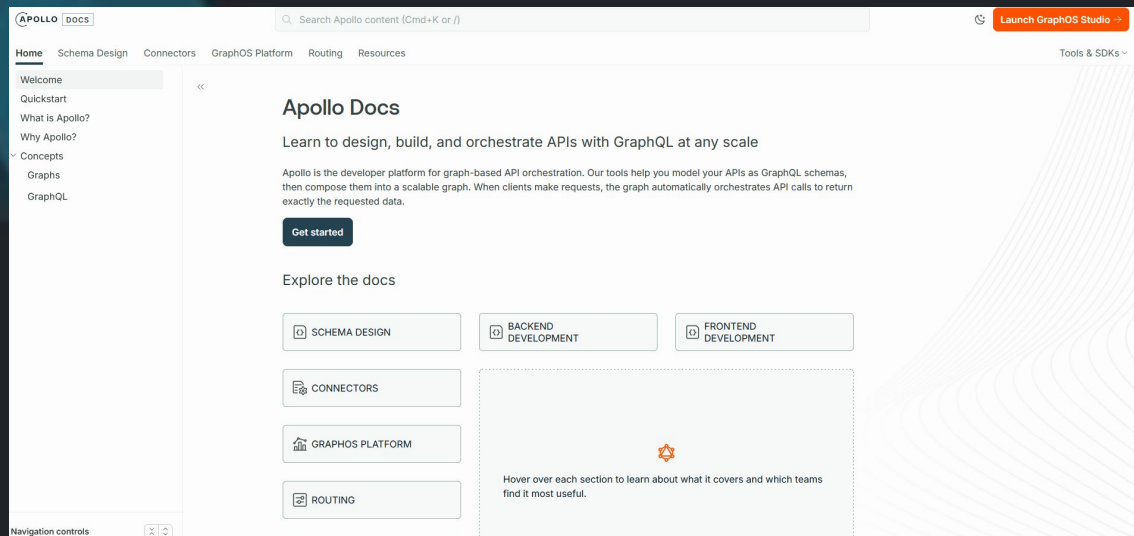
## Apollo-Client:

GraphQL realiza la petición, apollo la recibe y gestiona esa solicitud con el backend



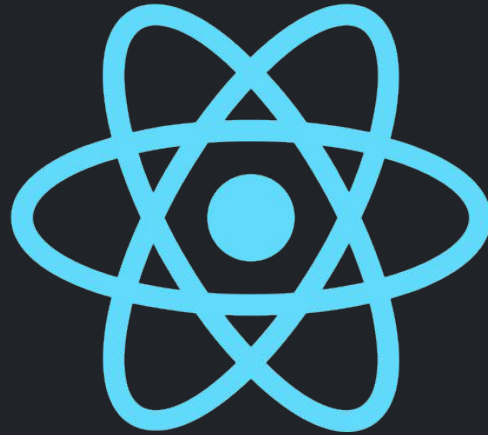
# Apollo-Client:

Sitio web: <https://www.apollographql.com/docs>



`npm install -g apollo-client`

APLICACIÓN FRONT END DE REACT CONVENCIONAL

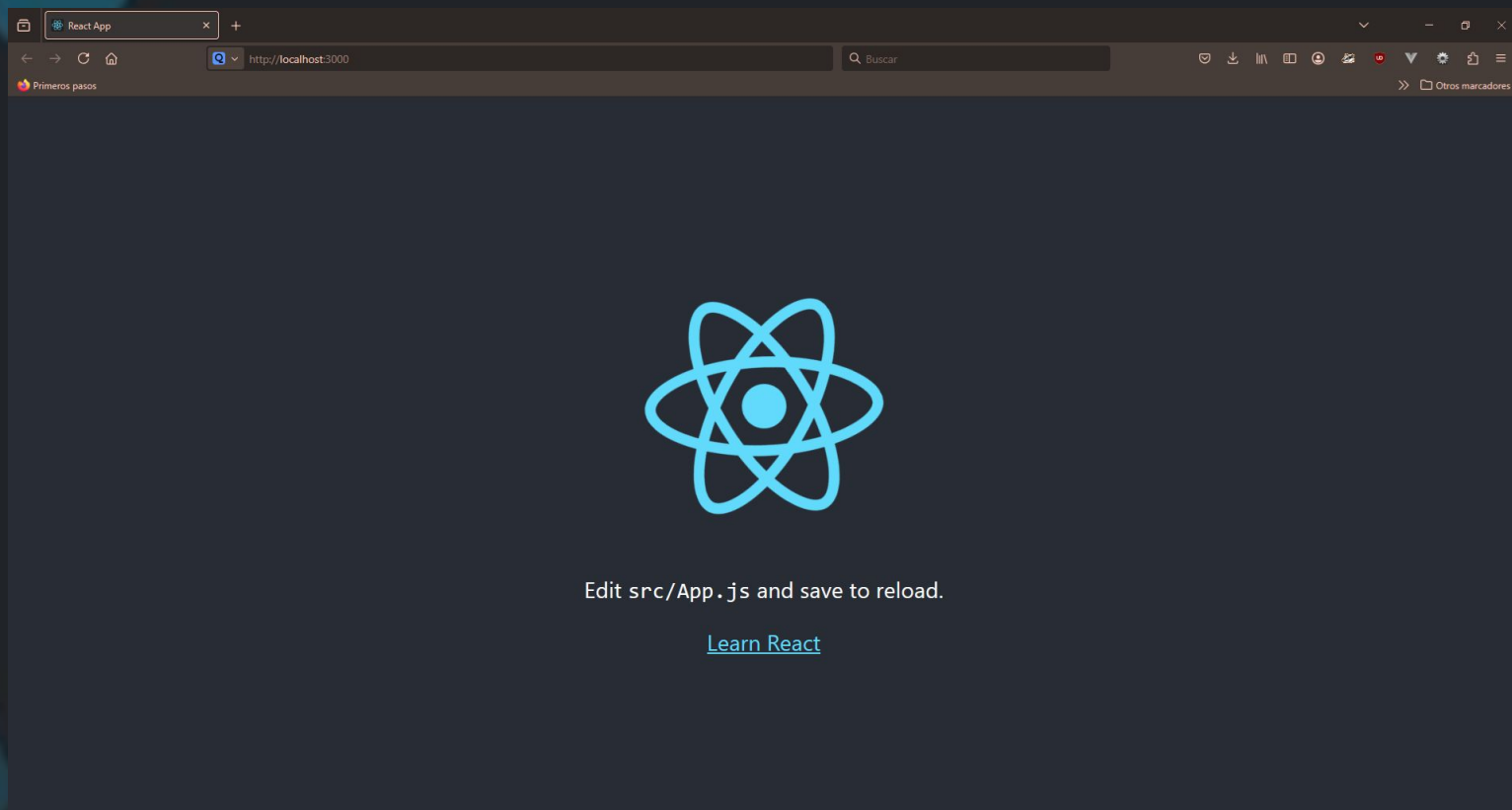


React

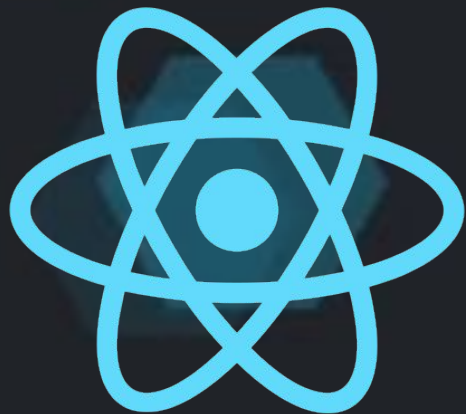


`npx create-react-app proyectobeerjs`

# APLICACIÓN FRONT END DE REACT CONVENCIONAL



# APLICACIÓN FRONT END DE REACT CON EL TEMPLATE DE VITE



React

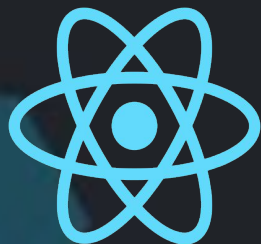


VITE



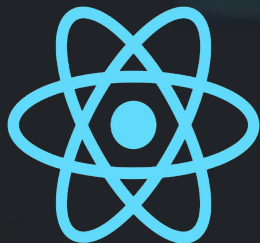
`npm create vite@latest proyectobeerjs -- --template react`

## APLICACIÓN DE REACT CONVENCIONAL Y CON VITE



React

Más lento y menor configuración



React



VITE

Más rápido y un poquito de configuración



## OTRA ALTERNATIVA QUE VIENE ES REACTQL

Ya viene incorporado GraphQL en React, y también incorpora TypeScript



+

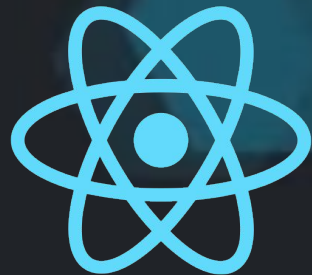


+



VIENE OTRA ALTERNATIVA QUE ES REACTQL

Y también viene Apollo Client

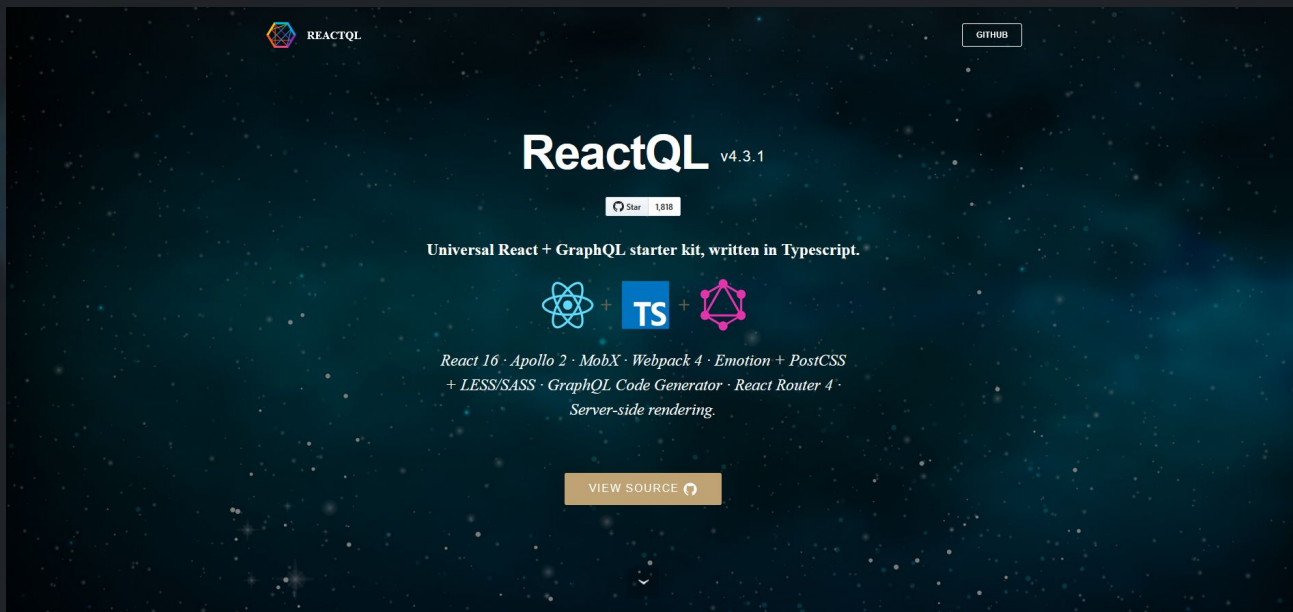


React



# VIENE OTRA ALTERNATIVA QUE ES REACTQL

Sitio web: <https://reactql.js.org/>



```
npm install -g reactql
```

# VIENE OTRA ALTERNATIVA QUE ES REACTQL



REACTQL

Message from GraphQL server: *Hello from graph.cool!*

Currently loading?: nope

- [Home](#)
- [About](#)
- [Contact](#)

Changed route: about

Runtime info:

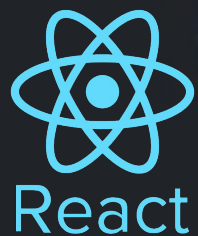
- Environment: *development*
- Running: *In the browser*

Stylesheet examples:

Styled by CSS

Styled by SASS

## React + GraphQL + Apollo Client + NodeJS + ExpressJS:



# Ejemplo de código en React con GraphQL

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import { ApolloClient, InMemoryCache } from '@apollo/client';

const client = new ApolloClient({
  uri: 'http://localhost:3000',
  cache: new InMemoryCache(),
});

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <ApolloProvider client={client}>
      <App />
    </ApolloProvider>
  </React.StrictMode>
);

reportWebVitals();
```

```
import React from 'react';
import { gql, useQuery } from '@apollo/client';

const GET_USERS = gql`
  query GetUsers {
    users {
      id
      title
    }
  }
`;

function UserList() {

  const { loading, error, data } = useQuery(GET_USERS);

  if (loading) return <p>Cargando usuarios...</p>;
  if (error) return <p>Error al cargar usuarios: {error.message}</p>;

  return (
    <div>
      <h2>Lista de Usuarios</h2>
      {data.users && data.users.length > 0 ? (
        <ul>
          {data.users.map((user) => (
            <li key={user.id}>
              <strong>ID:</strong> {user.id} - <strong>Título:</strong> {user.title}
            </li>
          ))}
        </ul>
      ) : null}
    </div>
  );
}
```



¿¿¿Preguntas???

# Muchas Gracias



[Sergio Gabriel Garzón](#)



[sergiog90arg](#)