



# NESTJS

**WWW.NESTJS.COM**  
**@NESTFRAMEWORK**



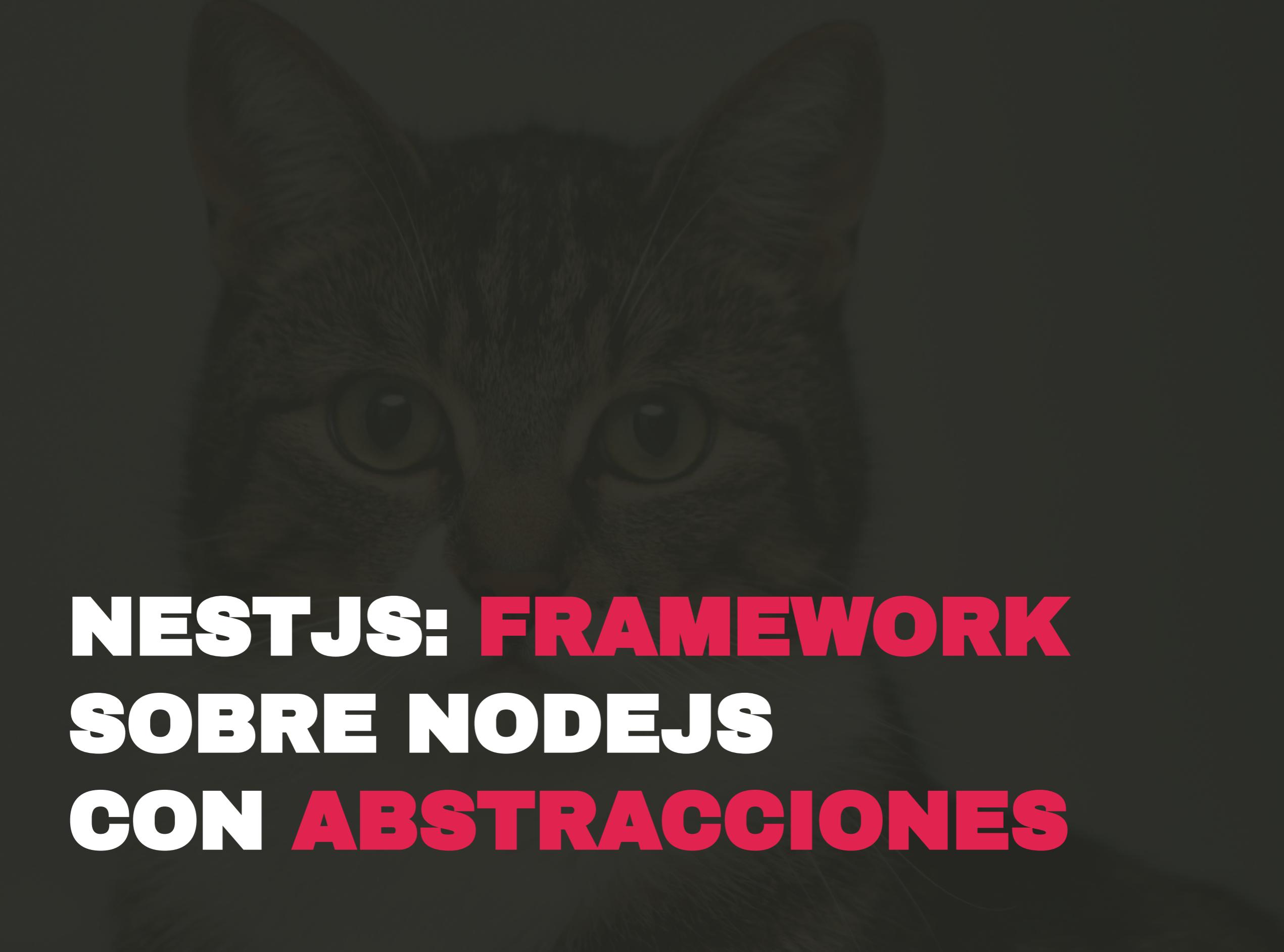
# **BACKEND CON NESTJS**



# NICOLAS MOLINA

**DEV & TEACHER AT PLATZI  
GOOGLE DEVELOPER EXPERT**

**@NICOBYTES**



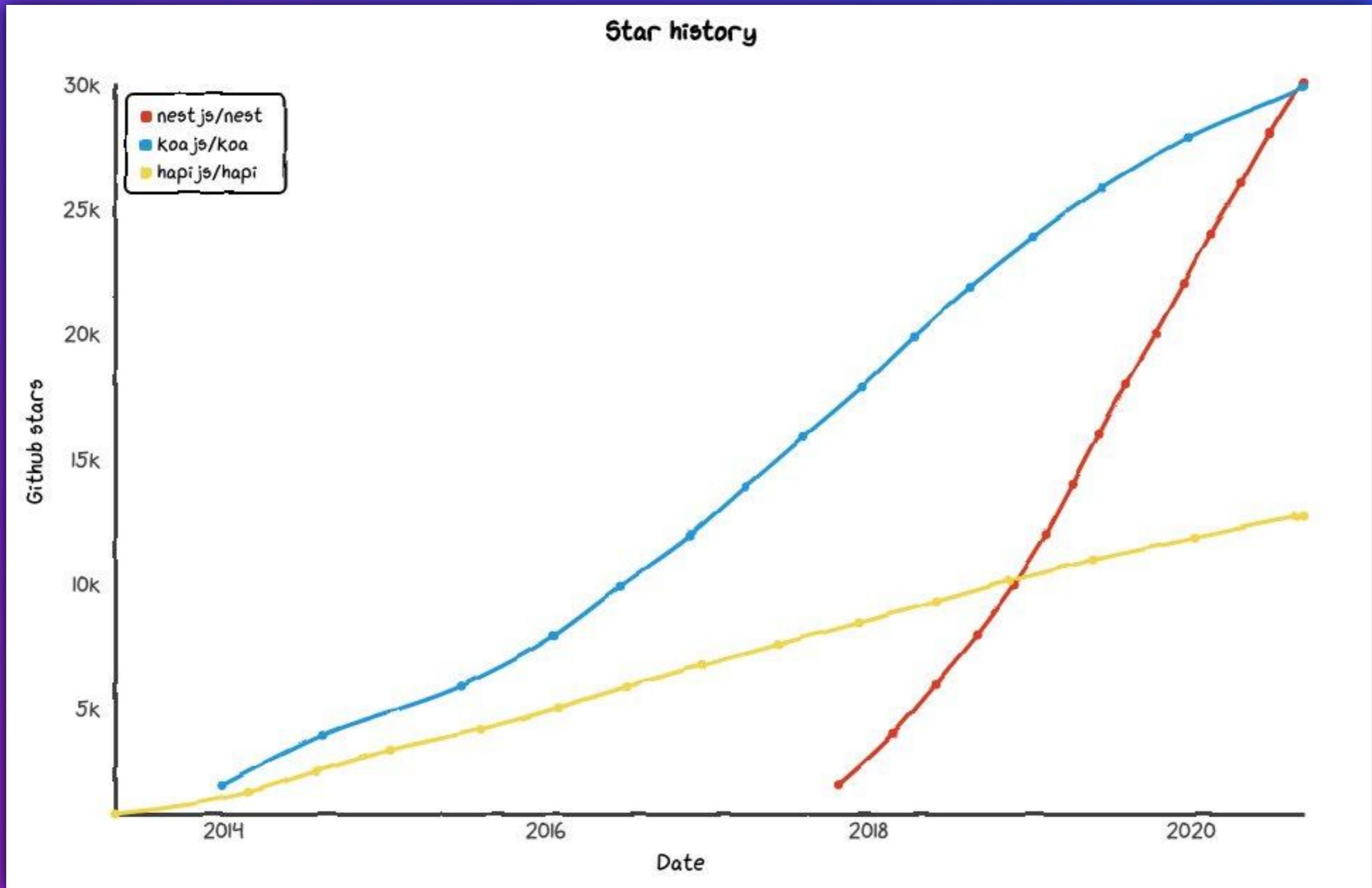
# **NESTJS: FRAMEWORK SOBRE NODEJS CON ABSTRACCIONES**

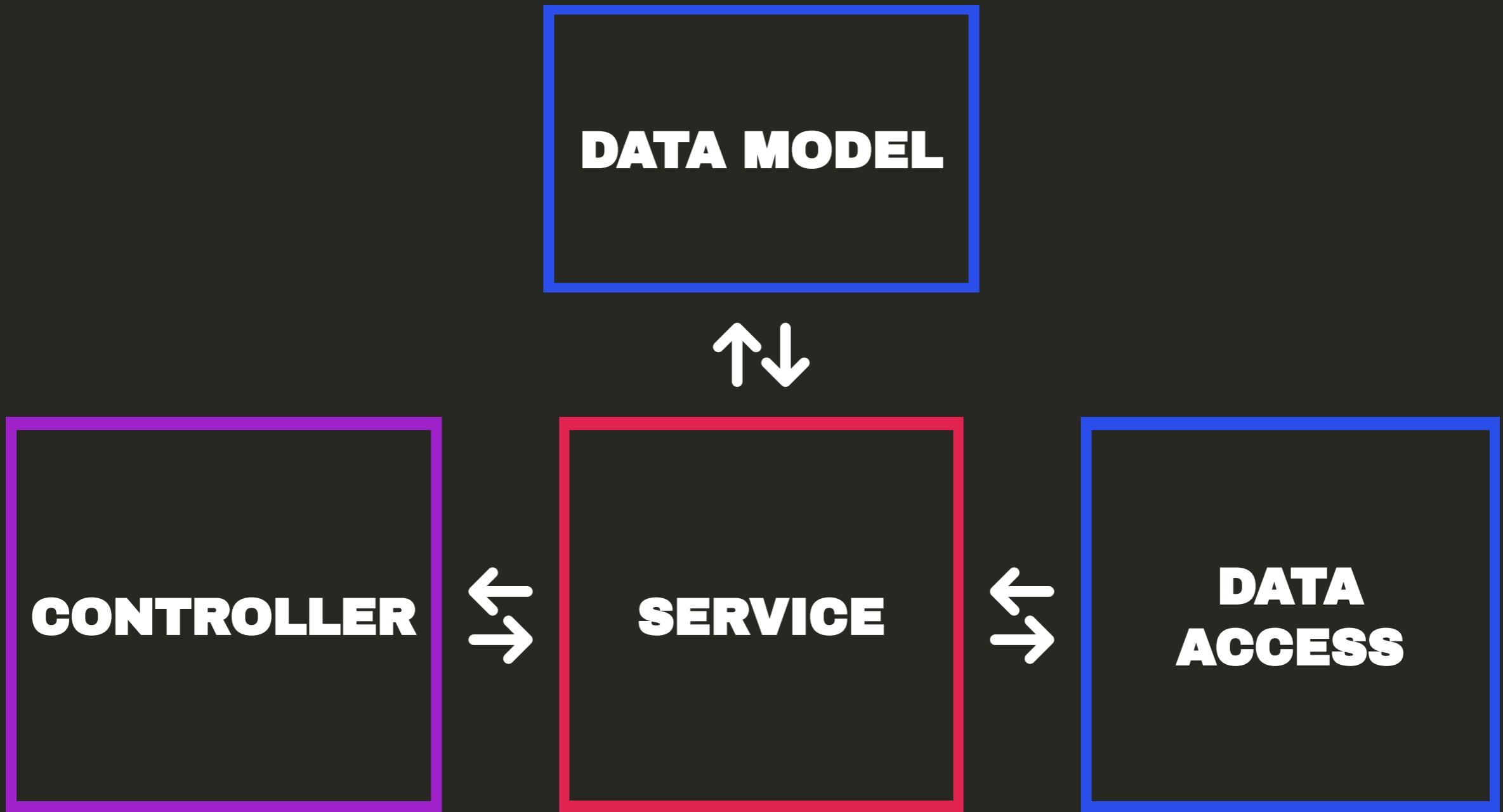
```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello World!');
});

app.listen(port, () => {
  console.log(`Listening at ${port}`);
});
```

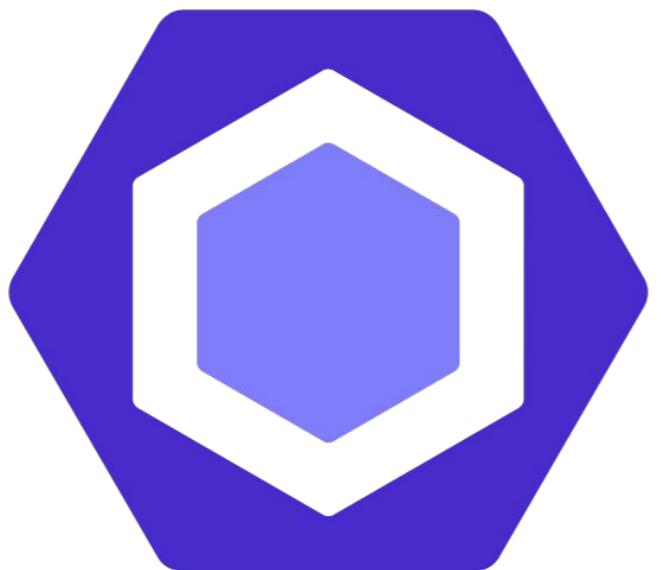
- SOLID
- TYPESCRIPT
- ORIENTADO A OBJETOS
- PROGRAMACIÓN FUNCIONAL
- PROGRAMACIÓN REACTIVA





# INSTALACIÓN

```
node --version
npm i -g @nestjs/cli
nest --version
nest --help
```



ESLint



Prettier

Visual Studio | Marketplace

Sign in  Search

Visual Studio Code > Programming Languages > ESLint

New to Visual Studio Code? [Get it now.](#)



## ESLint

Dirk Baeumer | 13,171,160 installs | ★★★★★ (161) | Free

Integrates ESLint JavaScript into VS Code.

[Install](#) [Trouble Installing?](#)

[Overview](#) [Version History](#) [Q & A](#) [Rating & Review](#)

---

### VS Code ESLint extension

 Azure Pipelines succeeded

Integrates [ESLint](#) into VS Code. If you are new to ESLint check the [documentation](#).

The extension uses the ESLint library installed in the opened workspace folder. If the folder doesn't provide one the extension looks for a global install version. If you haven't installed ESLint either locally or globally do so by running `npm install eslint` in the workspace folder for a local install or `npm install -g eslint` for a global install.

On new folders you might also need to create a `.eslintrc` configuration file. You can do this by either using the VS Code command `Create ESLint configuration` or by running the `eslint` command in a terminal. If you have installed ESLint globally (see above) then run `eslint --init` in a terminal. If you have installed ESLint locally then run `.\node_modules\.bin\eslint --init` under Windows and `./node_modules/.bin/eslint --init` under Linux and Mac.

**Categories**

[Programming Languages](#) [Linters](#)

**Tags**

[ignore](#) [javascript](#) [json](#) [jsonc](#) [linters](#)  
[multi-root ready](#)

**Resources**

[Issues](#) [Repository](#) [License](#) [Changelog](#) [Download Extension](#)

Visual Studio | Marketplace

Sign in  Search

Visual Studio Code > Formatters > Prettier - Code formatter

New to Visual Studio Code? [Get it now.](#)



## Prettier - Code formatter

Prettier | 11,184,888 installs | ★★★★☆ (230) | Free

Code formatter using prettier

[Install](#) [Trouble Installing?](#)

---

[Overview](#) [Version History](#) [Q & A](#) [Rating & Review](#)

### Prettier Formatter for Visual Studio Code

Prettier is an opinionated code formatter. It enforces a consistent style by parsing your code and re-printing it with its own rules that take the maximum line length into account, wrapping code when necessary.

JavaScript · TypeScript · Flow · JSX · JSON  
CSS · SCSS · Less  
HTML · Vue · Angular  
GraphQL · Markdown · YAML  
*Your favorite language?*

 Main passing downloads 81M installs 11M code style prettier [follow prettier](#) 17k

### Categories

Formatters

### Tags

angular commonmark css flow formatter  
graphql html ignore javascript js json jsx  
less markdown md mdx multi-root ready php  
prettier pug ruby scss styled-components  
styled-jsx swift ts typescript vue yaml yml

### Resources

[Issues](#)

### Installation

Visual Studio | Marketplace

Sign in  Search

Visual Studio Code > Other > EditorConfig for VS Code

New to Visual Studio Code? [Get it now.](#)



## EditorConfig for VS Code

**EditorConfig** | 3,178,685 installs | ★★★★★ (58) | Free

EditorConfig Support for Visual Studio Code

[Install](#) [Trouble Installing?](#)

[Overview](#) [Version History](#) [Q & A](#) [Rating & Review](#)

---

### EditorConfig for Visual Studio Code

 GitHub Actions success  Gitter [Join the EditorConfig VSCode chat](#)

This plugin [attempts](#) to override user/workspace settings with settings found in `.editorconfig` files. No additional or vscode-specific files are required. As with any EditorConfig plugin, if `root=true` is not specified, EditorConfig [will continue to look](#) for an `.editorconfig` file outside of the project.

This repository is specific to the [EditorConfig Extension for Visual Studio Code](#). Internally, it uses the [editorconfig npm package](#), which is one of a few [EditorConfig cores](#) available.

See also:

- [Visual Studio Code](#)

**Categories**

Other

**Tags**

config editor editorconfig multi-root ready

**Resources**

Issues Repository Homepage License Changelog Download Extension



¿QUÉ HAY DENTRO  
DE UNA APP?

**PROYECTO: PLATZI  
STORE**

**ORDER**

**USER**

**CUSTOMER**

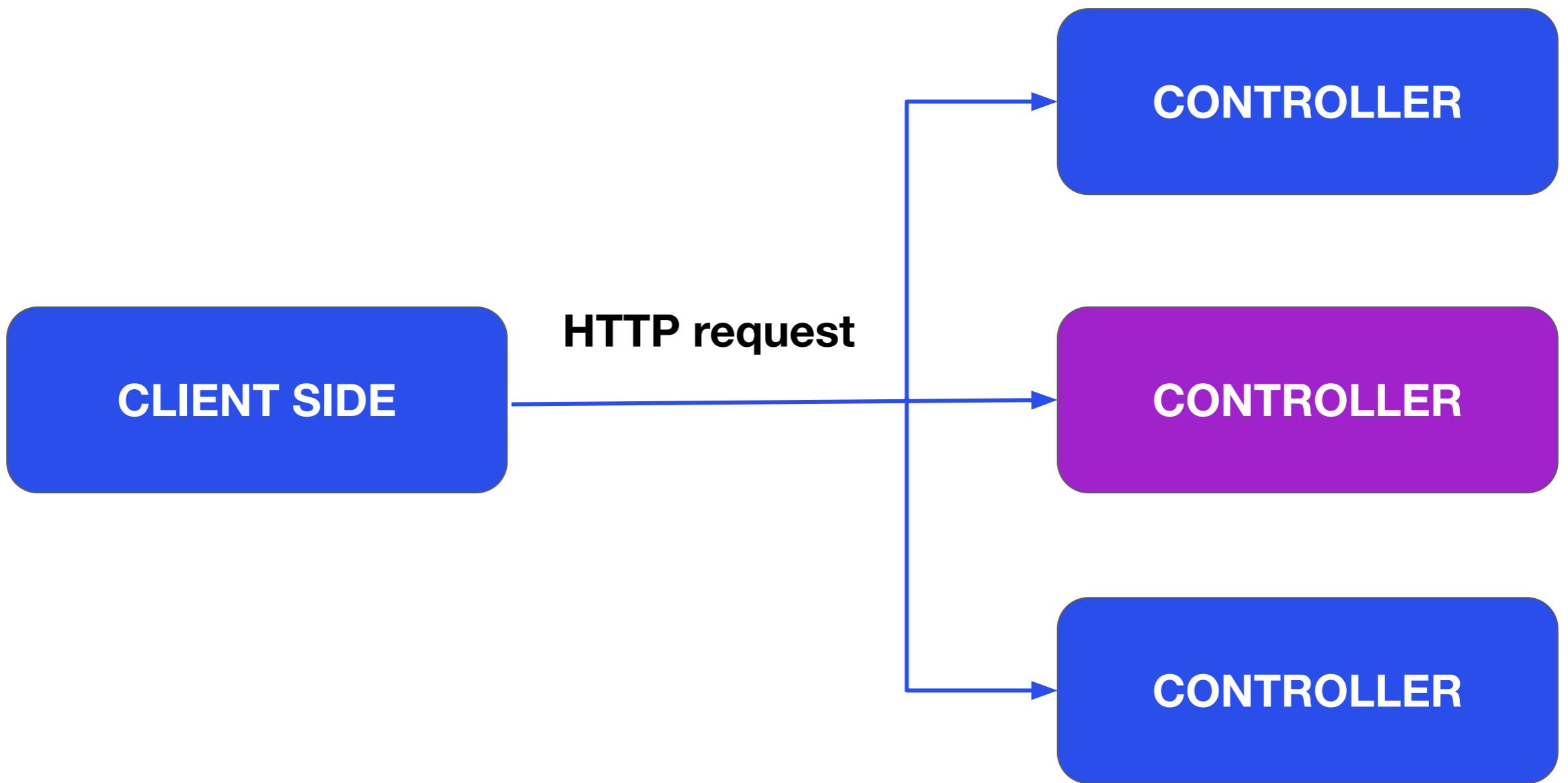
**CATEGORY**

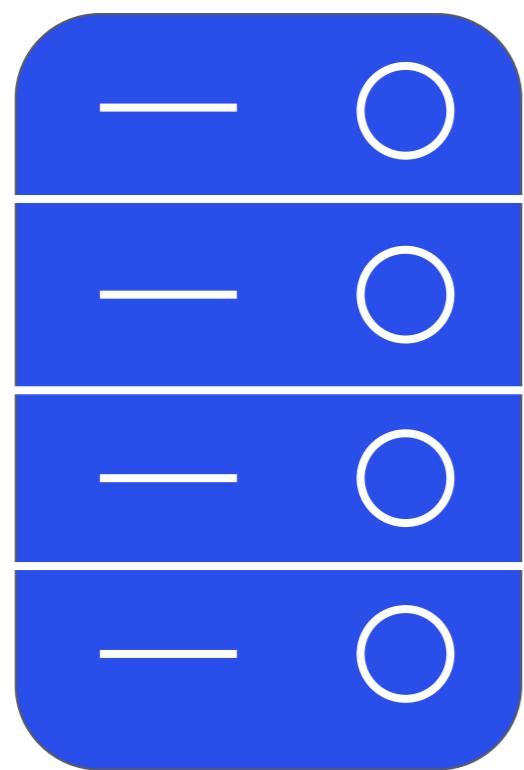
**PRODUCT**

**BRAND**



**REPASO A TS**





**GET**



**PUT**



**POST**



**DELETE**



```
import { Controller, Get } from '@nestjs/common';

@Controller()
export class AppController {

  constructor() {}

  @Get('nuevo-endpoint')
  getNewEndpoint(): string {
    return 'Soy un nuevo endpoint';
  }
}
```

```
import { Controller, Get } from '@nestjs/common';

@Controller()
export class AppController {

  constructor() {}

  @Get('nuevo-endpoint')
  getNewEndpoint(): string {
    return 'Soy un nuevo endpoint';
  }
}
```

# **CONTROLADORES**



# **RECIBIR PARÁMETROS**

api.example.com/tasks/{id}/  
api.example.com/people/{id}/  
api.example.com/users/{id}/tasks/

```
@Get(':id')
findOne(@Param('id') id: string) {
  return `The #${id}`;
}
```

```
@Get(':id')
findOne(@Param('id') id: string) {
  return `The #${id}`;
}
```

# PARÁMETROS QUERY

api.example.com/products  
api.example.com/products?page=1  
api.example.com/products?limit=10&offset=0  
api.example.com/products?region=USA  
api.example.com/products?region=USA&brand=XYZ  
api.example.com/products?region=USA&brand=XYZ&sort=asc

```
@Get()
findAll(@Query() paginationQuery) {
  const { limit, offset } = paginationQuery;
  return `Limit ${limit}, offset: ${offset}`;
}
```

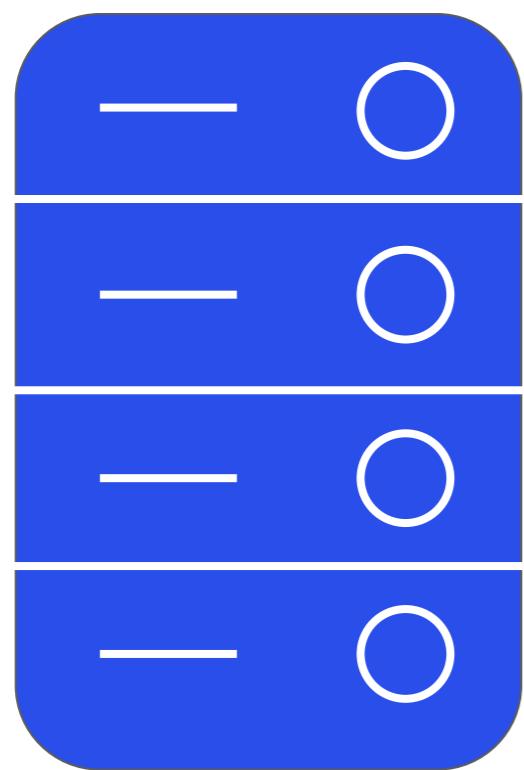
```
@Get()
findAll(@Query() paginationQuery) {
    const { limit, offset } = paginationQuery;
    return `Limit ${limit}, offset: ${offset}`;
}
```



**SINGLE RESPONSIBILITY**



# **POST: MÉTODO PARA CREAR**



**GET**



**PUT**



**POST**



**DELETE**



```
@Post( )
create(@Body( ) payload: any) {
    return payload;
}
```

```
@Post( )
create(@Body( ) payload: any) {
    return payload;
}
```



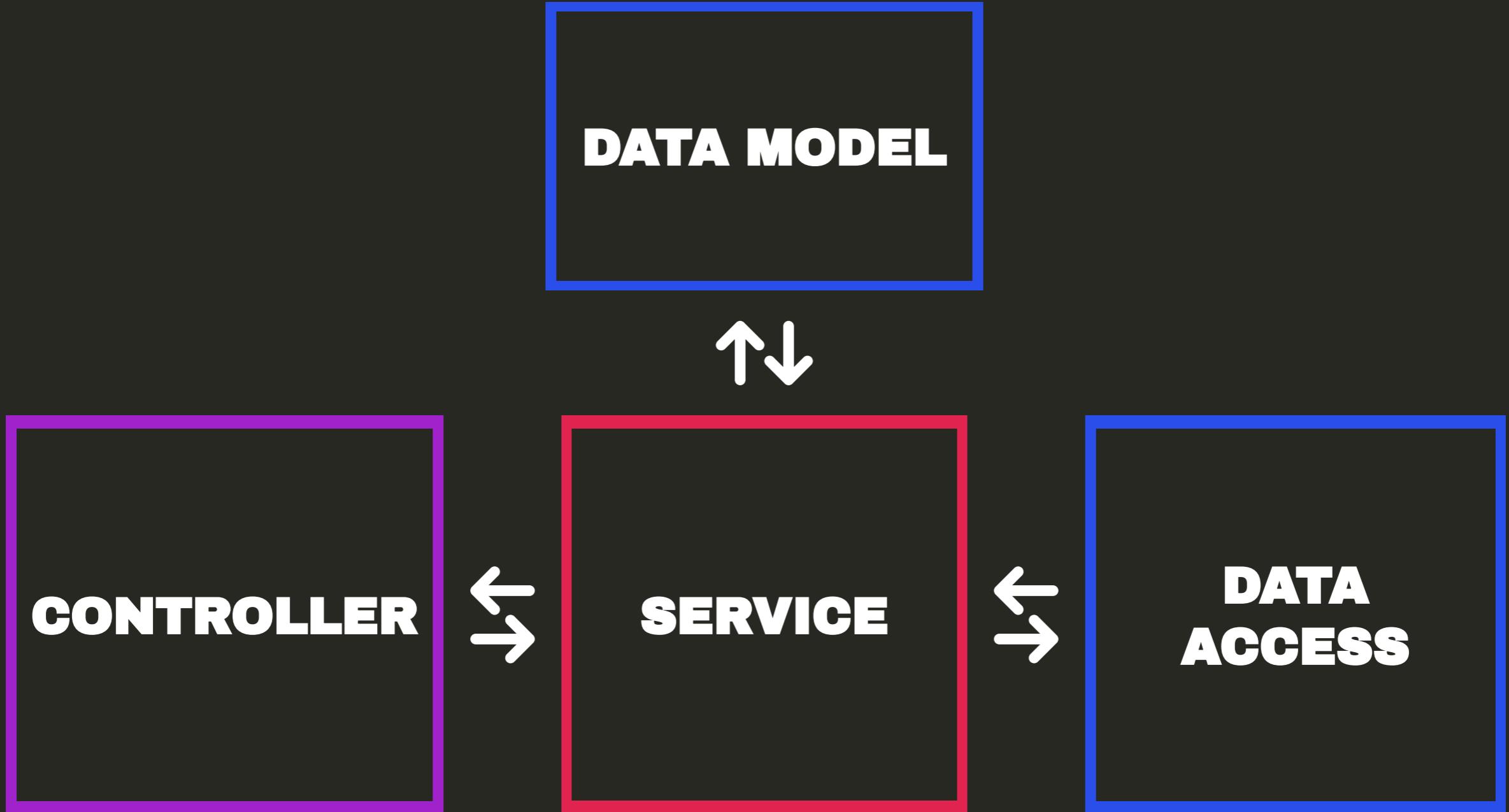
# MÉTODO PUT Y DELETE



# CÓDIGOS DE ESTADO



**CREANDO UN SERVICIO**



```
import { Injectable } from '@nestjs/common';

@Injectable()
export class ProductsService {
    ...
}
```

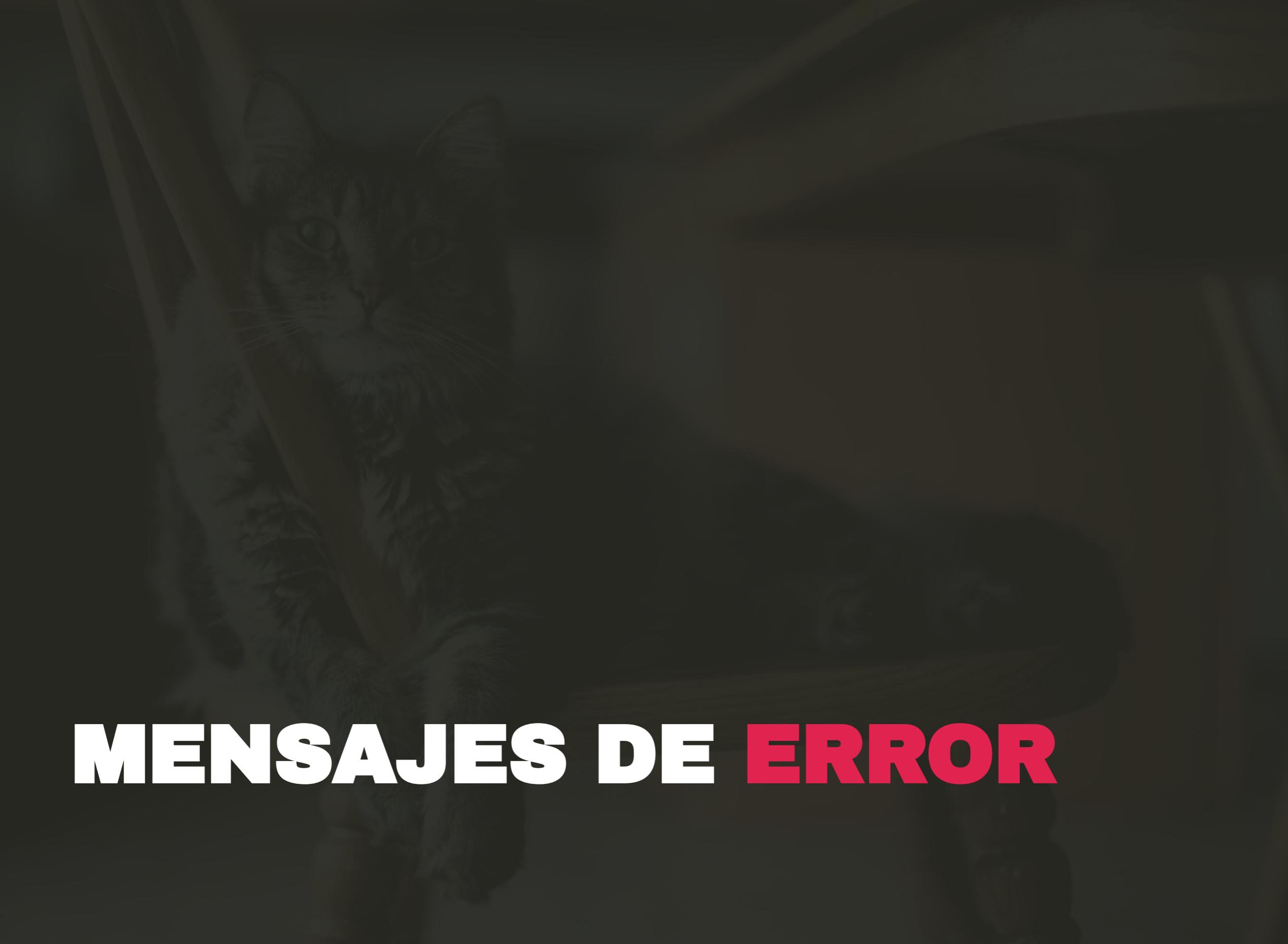
```
import { Injectable } from '@nestjs/common';

@Injectable()
export class ProductsService {

    ...
}
```



# **IMPLEMENTACIÓN DEL SERVICIO**

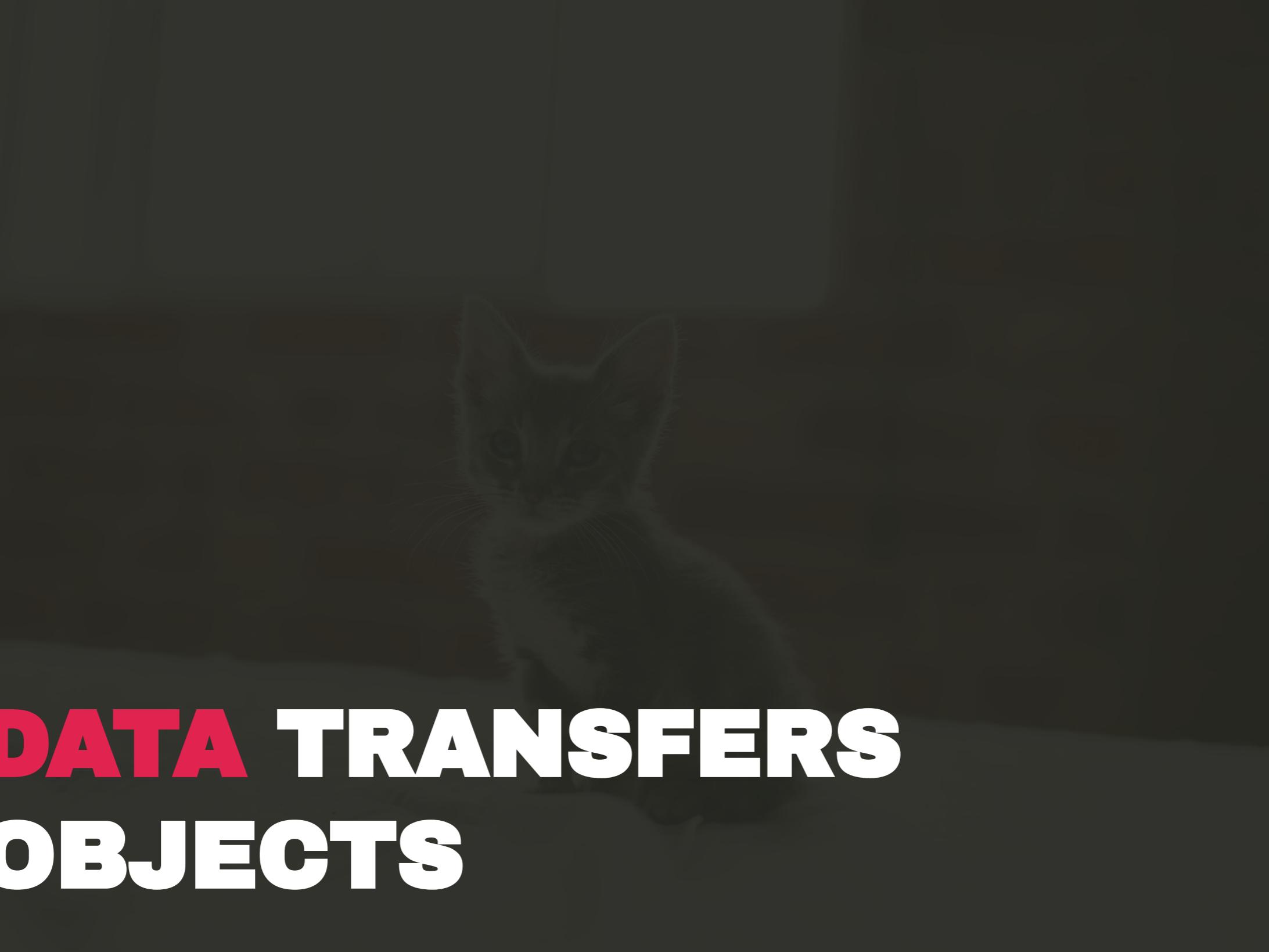


# MENSAJES DE ERROR

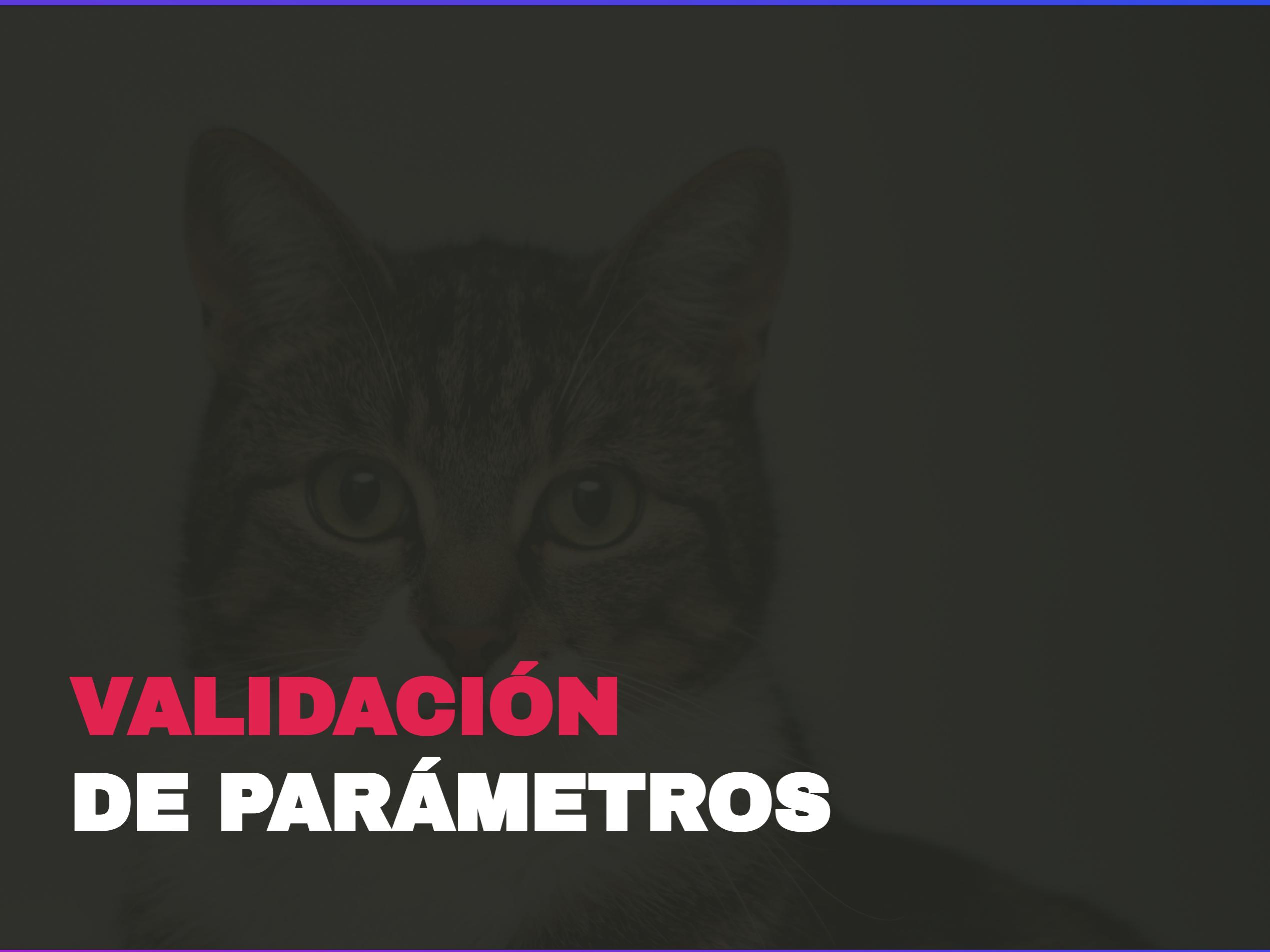


# INTRODUCCIÓN A PIPES

# **CREANDO UN PIPE**

A small, dark-colored kitten is positioned in the center of the slide, looking directly at the viewer. It has large, light-colored eyes and visible whiskers. The background behind the kitten is a dark, solid color.

# **DATA TRANSFERS OBJECTS**



# **VALIDACIÓN DE PARÁMETROS**



# EVITANDO PARÁMETROS INCORRECTOS