

Curso de  
**Consumo  
de API REST  
con JavaScript**





# JuanDC



# Requisitos

- Fundamentos de HTML y JavaScript.



- Comprensión del JS Engine.



- Asincronismo con JS.





**ESCUELA DE**



[www.platzi.com/web](http://www.platzi.com/web)

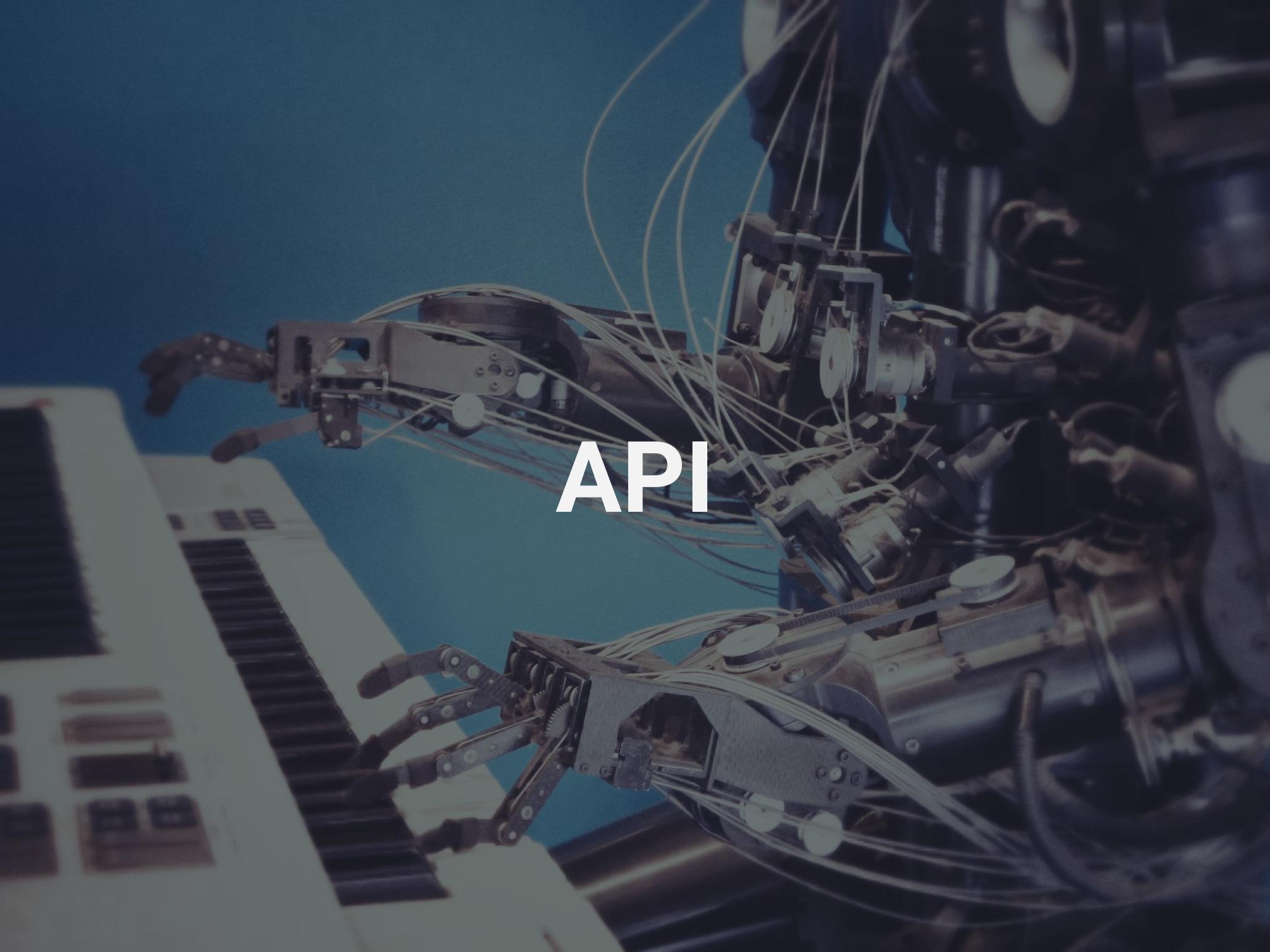


# Qué es una **API REST**



A photograph of the interior of a Tesla Model S. The steering wheel is black with the Tesla logo in the center. To the right is a large, curved touchscreen display showing a map and vehicle information. The car's interior is dark, with leather seats and a sleek design. The background through the windshield shows a blurred city street.

# Interfaz



# API

A close-up photograph of a ginger tabby cat sleeping soundly. The cat's eyes are closed, and its head is nestled on a white, textured, knitted blanket. Its body is curled up, and its front paw is visible, resting on the blanket. The background is a soft, out-of-focus grey.

**REST**

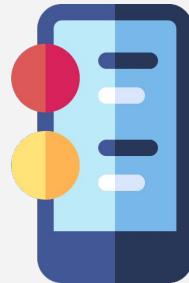
# HTTP con REST



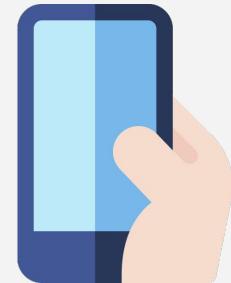
# Comunicación entre usuarios, **frontend y backend**



Frontend en  
HTML y CSS



Interacción  
del usuario



Petición a la  
API con JS



Backend escucha  
la petición

Consulta a la  
base de datos

Entrega la  
información



# Consume tu primera API REST



# HTTP Clients

- XMLHttpRequest 🎅
- Ajax 🐥
- \*Fetch 📝
- Axios, Trae... !

# **Endpoints y query parameters**

# Endpoints

- api.com/breeds
- api.com/categories
- api.com/images
- api.com/images/michi34

# Query parameters

- /categories?search=fun
- /breeds?limit=5&page=2
- /images?category=3
- /images/michi34?format/png



# Qué son los **HTTP Status Codes**



A close-up of Homer Simpson's face from the animated TV show "The Simpsons". He has his signature yellow skin, white eyes with black pupils, and a slightly open mouth. He is wearing a blue shirt under a grey suit jacket. Overlaid on his chin area is the text "2XX" in a large, bold, white sans-serif font.

2XX

A photograph of two white cats with yellow eyes, sitting on a brown corduroy sofa. They are facing each other, looking down at a small, colorful triangular card with a geometric pattern. The card is positioned between the cats' front paws. The image has a dark, slightly grainy texture.

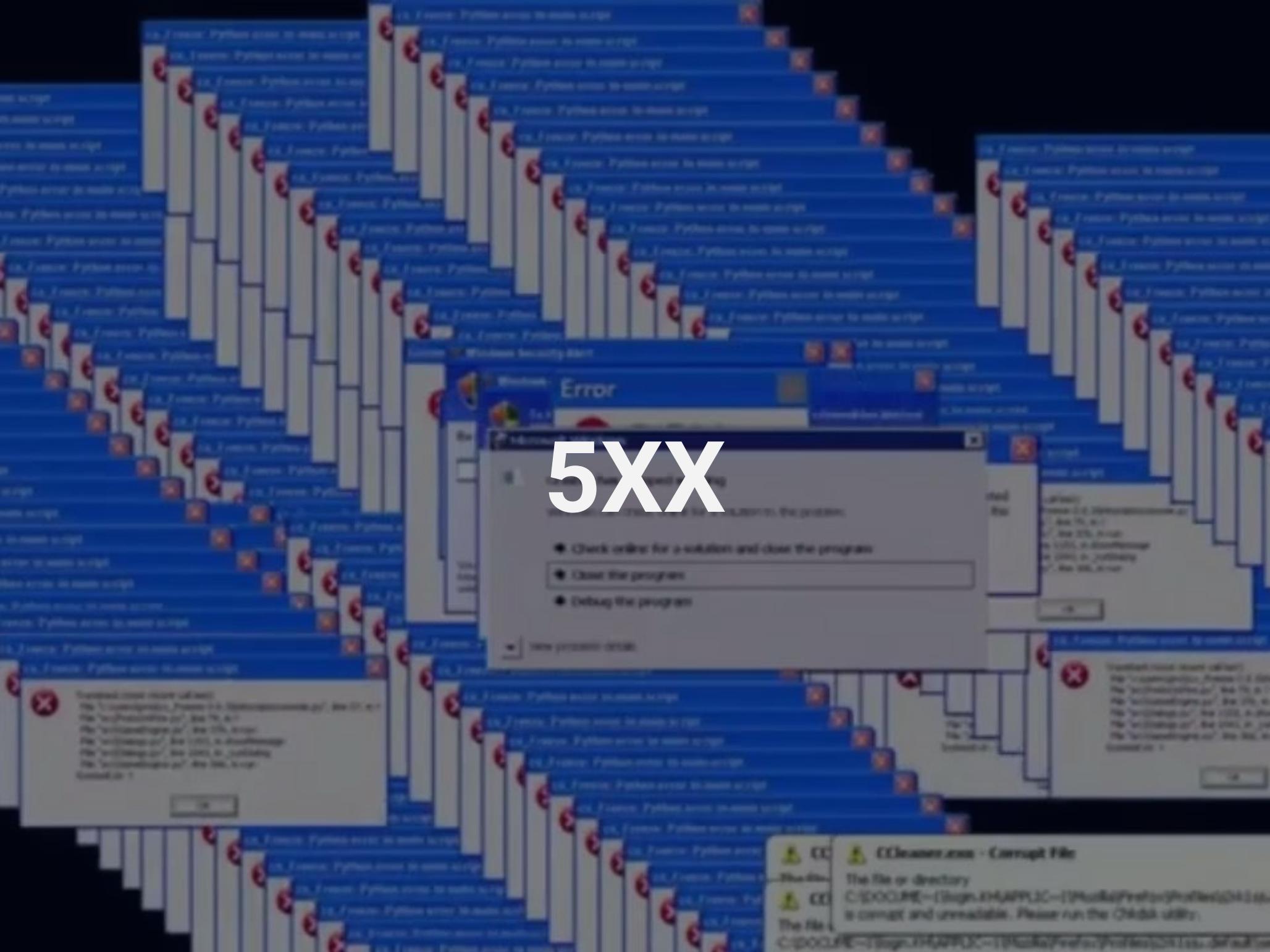
3XX

A colorful illustration of a city skyline at night. In the foreground, there's a pink building with a yellow roof and some green bushes. Behind it is a blue building with a white roof. To the right, there's a yellow building with a red roof and a bridge with orange lights. The sky is dark blue with some clouds.

4XX

404

# 5XX



5XX



5XX





# Qué es una **API KEY**



A dark, moody portrait of Thanos from the Marvel Cinematic Universe. He is shown from the chest up, wearing his signature purple and black suit. His face is partially obscured by shadow, with glowing yellow eyes looking directly at the viewer. The background is a dark, smoky blue.

# Autenticación

# Autorización



# API KEY

- **Query parameter:**  
?api\_key=ABC123
- **Authorization Header:**  
X-API-Key: ABC123

# Alternativas

- Authorization: Basic
- Authorization: Bearer Token
- OAuth 2.0
- Access Key + Secret Key

# ¿Por qué?

- Application-based authentication
- User-based authentication

# Maquetación del proyecto



# Qué son los Métodos HTTP



# Métodos HTTP

- GET 
- POST 
- PUT y PATCH 
- DELETE !

\*

# GET y POST

\*

# **Manipulación dinámica del DOM**

**PUT, PATCH  
y DELETE**

**Qué son los headers**

Elements    Console    Sources    Application    Network    Performance    Memory    »

Styles    Computed  
Filter

element.style  
}

\*, ::after, ::before {  
 box-sizing: border-box;

Inherited from ul, user agent menu, dir {  
 display: block;  
 list-style-type: none;  
 -webkit-margin-left: 1em;  
 -webkit-margin-bottom: 1em;  
 -webkit-margin-top: 0px;  
 -webkit-margin-right: 0px;  
 -webkit-padding-left: 40px;

Inherited from div, body {

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <div id="app">
      <todo-application>
        <#shadow-root (open)>
          <link rel="stylesheet" type="text/css" href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">
          <style>...</style>
          <nav class="navbar navbar-expand-md navbar-dark bg-dark">...</nav>
          <main class="container">
            <todo-form>
              <style>...</style>
              <div class="card todo-form">...</div>
            </todo-form>
            <hr>
            <todo-list ref="list">
              <style>...</style>
              <h2>Tasks:</h2>
              <ul ref="todos" class="list-group">
                <todo-task ref="task-1517176192142" id="task-1517176192142">...</todo-task> == $0
                <todo-task ref="task-1517176320397" id="task-1517176320397">...</todo-task>
                <todo-task ref="task-1517176329096" id="task-1517176329096">...</todo-task>
                <todo-task ref="task-1517176334849" id="task-1517176334849">...</todo-task>
              </ul>
            </todo-list>
          </main>
        </todo-application>
    </div>
```

# Body



# Headers



# Headers HTTP

- Content Type\*
- Authorization\*
- Cookies
- Location
- Etc.



# Header de autorización



# API KEY

- **Query parameter:**  
?api\_key=ABC123
- **Authorization Header:**  
X-API-Key: ABC123

\*

# Header de Content-Type

\*

# Content Types

- application
- audio
- image
- multipart
- text
- video
- vnd
- ...

# Application

- application/json
- application/xml
- application/zip
- **application/x-www-form-urlencoded**

# Audio

- audio/mpeg
- audio/x-ms-wma
- audio/vnd.rn-realaudio
- audio/x-wav

# Image

- image/gif
- image/jpeg
- image/png
- image/x-icon
- image/svg+xml

# Multipart

- multipart/mixed
- multipart/alternative
- multipart/related
- multipart/form-data**

# Text

- text/css
- text/csv
- text/html
- text/plain
- text/xml

# Video

- video/mpeg
- video/mp4
- video/quicktime
- video/webm

# VND

- application/vnd.ms-excel
- application/vnd.ms-powerpoint
- application/msword

\*

# FormData

\*

# Multipart

- multipart/mixed
- multipart/alternative
- multipart/related
- multipart/form-data**



# Librerías para consumir APIs



# HTTP Clients

- Axios\*
- Trae.js
- node-fetch (para Node.js)
- request (para Node.js)
- Etc.



# Otras propiedades de fetch



# Mode

- cors
- no-cors
- same-origin

# Caché

- default
- no-store
- reload
- no-cache
- force-cache
- only-if-cached

# Redirect

- follow
- error
- manual



# Reto: tu propio clon de fetch



# Fetch por dentro

- new **Request()**
- new **Headers()**
- new **Response()**



# El mundo más allá de REST



# sendBeacon

- No espera una respuesta del servidor,
- Es buena idea para analytics.

# GraphQL

- Empowered clients.
- All requests on  
the same endpoint.

```
query {  
  getMichi("id": 325) {  
    id  
    url  
  }  
}  
}
```

```
mutation {
  createMichi("id": 413) {
    id
    image {
      url
    }
  }
}
```

# Web Sockets

- Dejar el "túnel abierto".
- Útil para aplicaciones *real-time* (comunicación "instantánea").

```
const socket = io.connect('https://...');

socket.on('mensaje', function (data) {
    console.log(data)
});
```

# Web 3.0

- Dapps: aplicaciones descentralizadas.



Persistencia

Lógica

Cliente

Base de Datos

Almacenamiento

Lógica de Negocios

Web App

Mobile

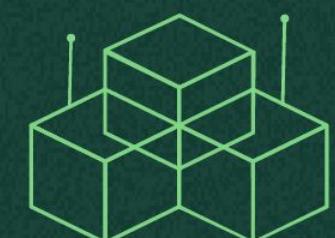
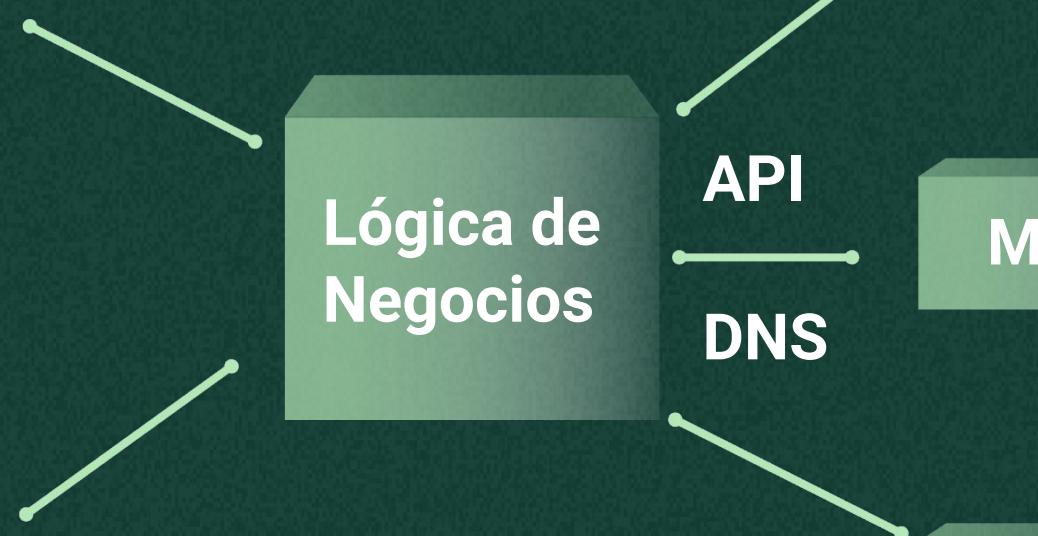
Web App

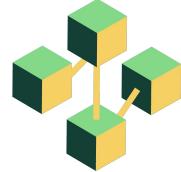
API

DNS

RTC

RTC





# Aplicaciones tradicionales

Persistencia



Lógica

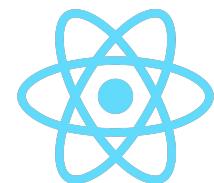


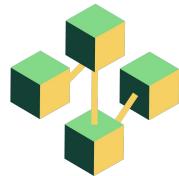
{REST:API}



DNS  
Domain Name System

Cliente





# Aplicaciones descentralizadas

## Persistencia



## Lógica



JSON-RPC



ENS

## Cliente



whisper





# ESCUELA DE



BLOCKCHAIN Y CRIPTOMONEDAS

[www.platzi.com/cripto](http://www.platzi.com/cripto)



Curso de  
**Consumo  
de API REST  
con JavaScript**

