

Jossep Ernsto Paredes Rivera

1. **\*\*Explora una API real\*\*** (10 min)

- Ve a: <https://jsonplaceholder.typicode.com/>
- Abre las herramientas de desarrollador (F12)
- Haz estas peticiones y observa:
  - GET /posts (lista todos)
  - GET /posts/1 (obtiene uno específico)
  - POST /posts (crea nuevo - usa cualquier título/body)

**\*\*Anota\*\***: ¿Qué códigos HTTP ves? ¿Qué patrones observas en las URLs?

**\*\*Entregable\*\***: Documento con observaciones de la API y capturas de pantalla del navegador mostrando las peticiones.

1. GET /posts

- **\*\*Código HTTP: \*\*** 200 OK
- **\*\*Respuesta: \*\*** Lista de posts en formato JSON (array de objetos)
- **\*\*Patrón de URL: \*\*** /recurso (plural)

```

> fetch("https://jsonplaceholder.typicode.com/posts/1")
  .then((response) => response.json())
  .then((json) => console.log(json));
< ▶ Promise {<pending>}
  ▼ {userId: 1, id: 1, title: 'sunt aut facere repellat provident occaecati excepturi optio reprehenderit', body: 'quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto')
    body: "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
    id: 1
    title: "sunt aut facere repellat provident occaecati excepturi optio reprehenderit"
    userId: 1
    ▶ [[Prototype]]: Object

```

Name	Headers	Preview	Response	Initiator	Timing	Cookies
rum	▼ General					
1	Request URL: https://jsonplaceholder.typicode.com/posts/1 Request Method: GET Status Code: 200 OK Remote Address: 104.21.16.1:443 Referrer Policy: strict-origin-when-cross-origin					
	▼ Response Headers					
	Access-Control-Allow-Credentials: true Age: 2675 Alt-Svc: h3="443"; ma=86400 Cache-Control: max-age=43200 Cf-Cache-Status: HIT Cf-Ray: 97cb3e6fc95bdda6-IAD Content-Encoding: zstd Content-Type: application/json; charset=utf-8 Date: Wed, 10 Sep 2025 01:45:05 GMT Etag: W/"124-yiKdLzqO5gfBrJfrcdJ8Yq0LGnU" Expires: -1					
2 requests 1.4 kB transferred 292 B resources	Nel	{"report_to":"heroku-nel","response_headers":["Vi				

## 2. GET /posts

- **Código HTTP:** 200 OK
- **Respuesta:** Un solo post (objeto JSON)
- **Patrón de URL:** /recurso/{id}

```
> fetch('https://jsonplaceholder.typicode.com/posts')
.then((response) => response.json())
.then((json) => console.log(json));
c ▶ Promise (pending)

▼ [100] [{"id":1,"title":"sunt aut facere repellat provident occaecati excepturi optio reprehenderit", "body":"quia et suscipit\qsuscipit recusandae consequuntur utrum est autem sunt rem exumet architecta"}, {"id":2,"title":"qui est esse", "body":"est rerum tempore vitae\sequi sint nihil reprehendo\eripere non debitis possimus qui neque nisi nulla"}, {"id":3,"title":"ea molestias quasi exercitationem repellat qui ipsa sit aut", "body":"et justo sed que laore\voluptates accaeati omnis e\molestiae porro eius odio et labore et velit aut"}, {"id":4,"title":"eum et est occaecati", "body":"ullam et saepe reiciendis voluptates adipisci\vit - ipsum laore\quis sunt voluptates rerum illo velit"}, {"id":5,"title":"necium aut odio", "body":"rem\underline{andae} veniam quae\erat sunt sed\ullam aut fugiat voluptatibus qui\vest aut tenetur dolor nemo"}, {"id":6,"title":"dolores eum magni eos aperiam quia", "body":"ut adipiscite corporis harum nihil\qui provident -\al\voluptate dolores velit et dolorum\ue molestiae"}, {"id":7,"title":"magnam facilis autem", "body":"dolore placeat quibusdam ea quo vitae\magni quis e\ excepturi ut qui\usunt ut sequi eos ea sed quas"}, {"id":8,"title":"dolorem dolor est ipsum", "body":"dignissimos aperiam dolorem qui eum\facilis quibus\voluptas ut commodi dolor voluptatum modi aut vitae"}, {"id":9,"title":"necium laore omnis dolorem tempora et accusantium", "body":"consectetur autem nescium laore dolorem\quia a\ut aut quod aut provident voluptas autem voluptas"}, {"id":10,"title":"expedit mollitia id quia eum", "body":"quo et expedita modi cum officia vel magni\vololorib\it\quos veniam quod sed accusamus veritatis error"}, {"id":11,"title":"et ea vero quia laudantium autem", "body":"delectus reiciendis molestiae occaecati non minima\luptates ut commodi qui incidunt\vit autem commodi"}, {"id":12,"title":"in quibusdam tempore odit est dolorem", "body":"Itaque id aut magni\representant\quia et ea odit e\ia id voluptatem\incidunt ea est distinctio odio"}, {"id":13,"title":"omnium ut in voluptate mollitia et saepe quo anim", "body":"aut dicta possimus sint mollitia voluptas commodi -\sumenda consetetur porro architecto ipsum ipsum"}, {"id":14,"title":"voluptatem eligendi optio", "body":"fuga et accusamus dolorum perferendis illo voluptas\molestiae beata\sed aut voluptas totum sit illum"}, {"id":15,"title":"eveniet quod temporibus", "body":"reprehenderit qui placeat\velit minima officia \duas\unde qui delectus\vol\facilis harum fugiat\vitae"}, {"id":16,"title":"sint suscipit perspiciatis velit dolorum rerum ipsa laboribus odio", "body":"suscipit nam nisi quo aperiam aut\asperiores eos f\ magni mollitia accusamus ea nisi voluptate dicta"}, {"id":17,"title":"fugit voluptas sed molestiae voluptatem provident", "body":"eos voluptas et aut odit natus earum\aspernatur fuga\ aut\qui nihil ratione nemo velit ut aut id quo"}, {"id":18,"title":"voluptate et itaque vero tempora molestiae", "body":"eveniet quo qui\laborum totum consequatur non dolo\repudiandae\vest voluptates vel debitis et magni"}, {"id":19,"title":"dolor sint quo a velit explicabo quia nam", "body":"eos qui et ipsum ipsum suscipit aut\sed omnis non -\molestiae aut et\ae rem suscipit\vol\us impedit esse"}, {"id":20,"title":"adipisci placeat illum aut reiciendis qui", "body":"illum quis cupiditate provident sit magnus\sed -\et\ne\adipisci quo iste expedita sit quos voluptas"}, {"id":21,"title":"doloribus ad provident suscipit at", "body":"qui consequuntur ducimus possimus quisque amet \e\erum consequatur expedita quidem eumque explicabo"}, {"id":22,"title":"aperietis ea ipsum voluptatem modi minima quia sint", "body":"repellat aliquid praesentium dolorum quoniam totum\accusandae veniam\tempore et tenetur expedita sunt"}, {"id":23,"title":"dolor sint quo a velit explicabo quia nam", "body":"eos qui et ipsum ipsum suscipit aut\sed omnis non -\molestiae aut et\ae rem suscipit\vol\us impedit esse"}, {"id":24,"title":"autem hic labore sunt dolores incidunt", "body":"eris et ex nulla\omnis voluptas quia qui\voluptate\sed\id dignissimos aut sed asperiores deserunt"}, {"id":25,"title":"rem alias distinctio non quis", "body":"ullam consequatur ut\omnis quis sit vel consequunt -\nstrum\molestiae illo tempore quia et distinctio"}, {"id":26,"title":"est et quis odit qui non", "body":"similique esse doloribus nihil accusamus\omnis dol\is cum ut laudantium\omnis aut molestiae vel vero"}, {"id":27,"title":"quasi id et eos tenetur aut quo autem", "body":"eum sed dolores ipsa sint possimus debitis occae\ctibus rerum sed inventore temporibus consequatur"}, {"id":28,"title":"delectus ullam et corporis nulla voluptas sequi", "body":"non et quae\et dolore\sed\recusand\ut voluptatibus voluptatem\incidunt nostrum eum"}, {"id":29,"title":"justo eius quod necessitatibus culpa ea", "body":"odit magnam ut saepe sed non qui\tempore atque n\m\ repudiandae odit magni similique sed cum maiores"}, {"id":30,"title":"a quo magni similique perferendis", "body":"alios dolor consequuntur\di\omnis non enim\et\ora quia autem rem\ea provident perspicatis quia"}]
```

Filter								
<div>20,000 ms40,000 ms60,000 ms80,000 ms100,000 ms120,000 ms140,000 ms160,000 ms180,000 ms200,000 ms220,000 ms240,000 ms</div>								
<div>⌵</div>								
<div>Name</div>								
<div>rum</div>								
<div>1</div>								
<div>posts</div>								
<div>⌵</div>								
<div>Request URL</div>								
<div>Request Method</div>								
<div>Status Code</div>								
<div>Remote Address</div>								
<div>Referrer Policy</div>								
<div>⌵</div>								
<div>Access-Control-Allow-Credentials</div>								
<div>Age</div>								
<div>Alt-Svc</div>								
<div>Cache-Control</div>								
<div>CF-Cache-Status</div>								
<div>CF-Ray</div>								
<div>Content-Encoding</div>								
<div>Content-Type</div>								
<div>Date</div>								
<div>Etag</div>								
<div>Expires</div>								
<div>Nel</div>								
<div>3 requests9.2 kB transferred27.8 kB resources</div>								

### 3. POST /posts

- \*\*Código HTTP: \*\* 201 Created

- \*\*Respuesta: \*\* Objeto creado con ID asignado



- Principio clave 2: Uso de métodos HTTP estándar (GET, POST, PUT, DELETE)
- Principio clave 3: Stateless (sin estado entre peticiones)

### **Recursos vs Representaciones**

- Recurso = Entidad conceptual (ej: un usuario, un producto)
- Representación = Forma concreta (ej: JSON, XML)
- Ejemplo: /users/123
  - El recurso es: el usuario con id 123
  - La representación es: el JSON devuelto

### **Códigos HTTP esenciales**

- 200: OK (éxito)
- 201: Created (recurso creado)
- 400: Bad Request (petición inválida)
- 404: Not Found (no encontrado)
- 500: Internal Server Error (error del servidor)

### **Esquema de API REST para EcoMarket**

#### **Recurso principal: products**

Método	Endpoint	Descripción	
-----	-----	-----	

| GET | /products | Lista todos los productos |

| GET | /products/{id} | Obtiene un producto específico |

| POST | /products | Crea un nuevo producto |

| PUT | /products/{id} | Actualiza un producto |

| DELETE | /products/{id} | Elimina un producto |

### **Justificación**

- Se usa 'products' en plural siguiendo convención REST.
- PUT reemplaza todo el recurso, no crea si no existe (devuelve 404).
- DELETE devuelve 404 si el producto no existe.
- URLs son sustantivos, no verbos.
- Se consideran casos extremos (IDs inexistentes, datos inválidos).

### **Bitácora de Decisiones Arquitectónicas (EcoMarket)**

- Recurso principal: products
- Endpoints: GET, POST, PUT, DELETE sobre /products y /products/{id}
- Validaciones: nombre no vacío, price >= 0, stock >= 0
- Formato de errores: siempre JSON {"error": "mensaje"}
- PUT y DELETE devuelven 404 si el producto no existe
- Decisión difícil: PUT no crea, solo actualiza
- Mejoras futuras: paginación, filtros, autenticación, persistencia

### **Reflexión de Diseño**

- Decisión más difícil: Elegir si PUT debe crear o solo actualizar. Decidimos que solo actualiza y devuelve 404 si no existe.
- Códigos HTTP: Elegimos 200 para éxito, 201 para creación, 400 para errores de validación, 404 para no encontrado, 204 para borrado exitoso.

- Como frontend developer, me gustaría que los errores incluyeran un campo "code" para facilitar el manejo en el cliente.