



重庆味道 耙耙牛肉

老沧州火锅鸡

万达店

API 3.0 HTTP



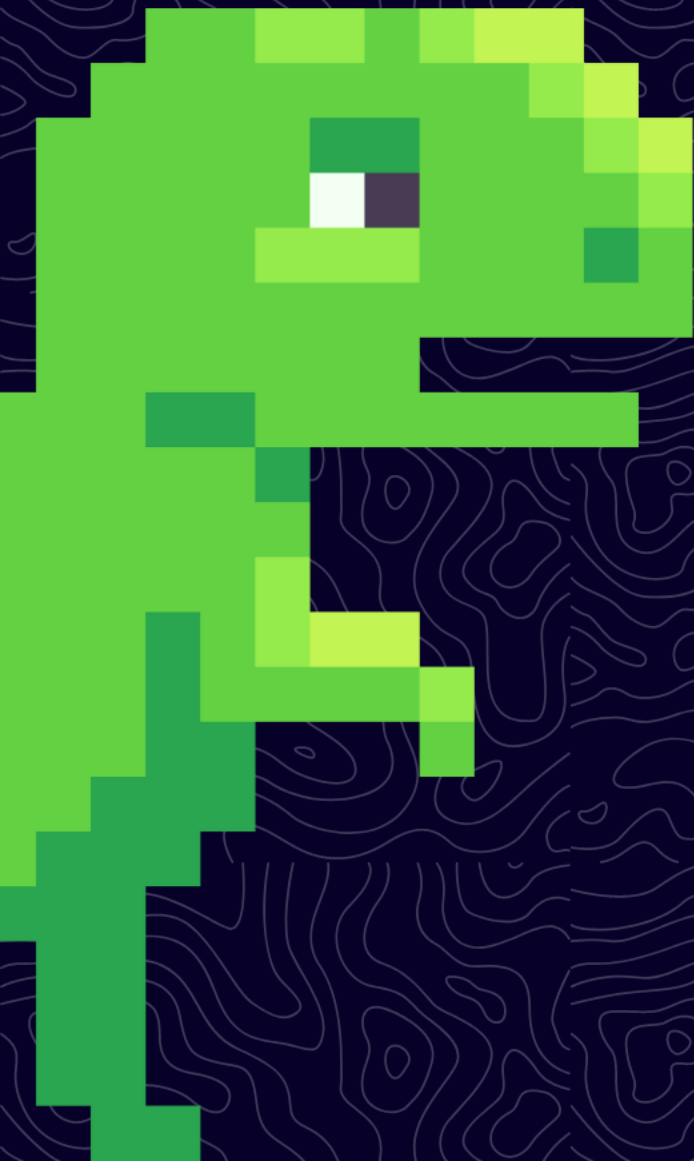
Request

START





HISTORIA DE LAS APIs



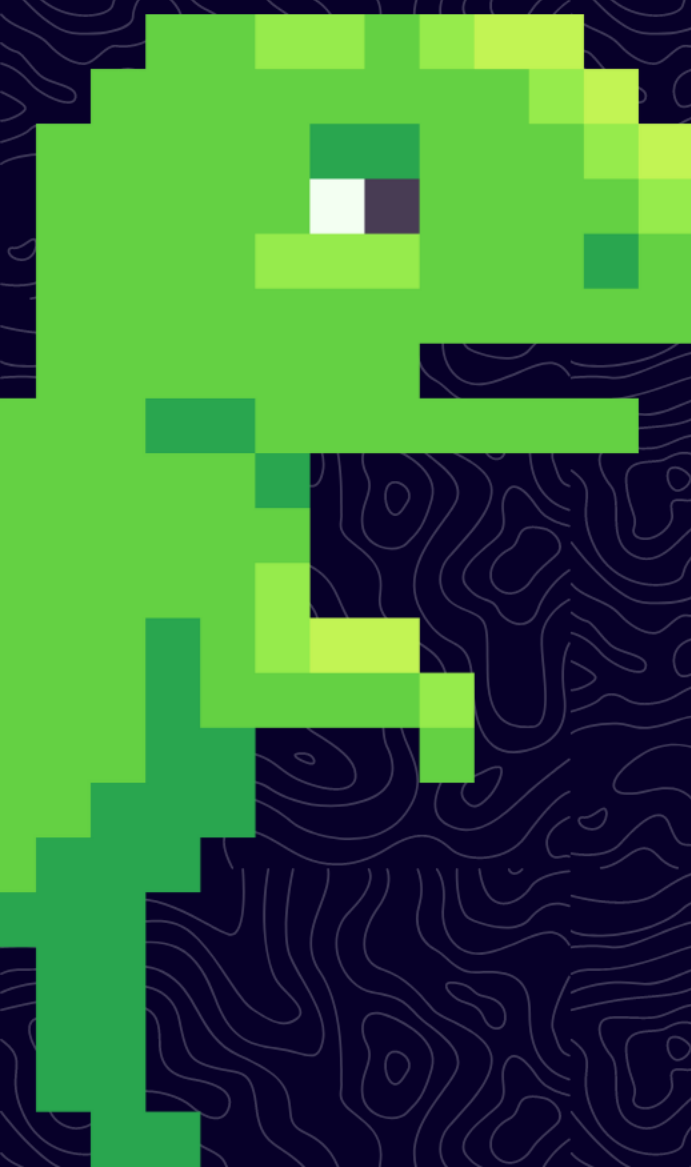


- ☆ Las API surgieron en los comienzos de la informática, mucho antes que la computadora personal.
- ☆ En esa época, una API normalmente se usaba como biblioteca para los sistemas operativos.
- ☆ Después de casi 30 años, las API se expandieron más allá de los entornos locales.
- ☆ A principios del año 2000, ya eran una tecnología importante para la integración remota de datos.





¿QUÉ ES
UNA API?

Application Process Interface

- 
- 
- 
- ☆ **Permiten** que sus **productos y servicios se comuniquen con otros**, sin necesidad de saber cómo están implementados.
 - ☆ Esto **simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero.**



¿CÓMO
FUNCIONAN?

¿Cómo funcionan las API?

☆ La aplicación que **envía la solicitud** se llama **cliente**, y la que **envía la respuesta** se llama **servidor**.



FUNCIONAMIENTO DE LAS API

Funcionamiento de API's - SOAP

- ☆ En estas API el cliente y el servidor intercambian mensajes mediante XML.
- ☆ Se trata de una API menos flexible que era más popular en el pasado.
- ☆ Usan XML para el formato de sus mensajes y reciben solicitudes a través de HTTP.

Funcionamiento de API's - Websocket

- ★ Utiliza objetos JSON para pasar datos.
- ★ Admite la comunicación bidireccional entre las aplicaciones cliente y el servidor.

Funcionamiento de API's - REST

- ☆ El cliente envía las solicitudes al servidor como datos. El servidor utiliza esta entrada del cliente para *iniciar funciones internas y devuelve los datos de salida al cliente.*
- ☆ REST define un conjunto de funciones como GET, PUT, DELETE, etc. que los clientes pueden utilizar para acceder a los datos del servidor. Los clientes y los servidores intercambian datos mediante HTTP.

TIPOS DE API

Tipos de API

☆ API privadas

- ☆ Son internas de una empresa y solo se utilizan para conectar sistemas y datos dentro de la empresa.

☆ API públicas

- ☆ Están abiertas al público y pueden cualquier persona puede utilizarlas. Puede haber o no alguna autorización y coste asociado a este tipo de API.

¿CÓMO PROTEGER UNA API?

¿Cómo proteger una API?

- ☆ **Tokens de autenticación:** comprueban que los usuarios son quienes dicen ser y que tienen los derechos de acceso para esa llamada concreta a la API.
- ☆ **Claves:** identifican la aplicación y se aseguran de que tiene los derechos de acceso necesarios para hacer la llamada a la API en cuestión. Las claves de API no son tan seguras como los tokens, pero permiten supervisar la API para recopilar datos sobre su uso.

A conceptual image featuring two hands, one from the top and one from the bottom, holding a glowing blue sphere. The sphere is covered in binary code (0s and 1s) and has a bright light source inside, creating a lens flare effect. The background is a solid dark blue. The text 'HTTP REQUEST' is overlaid in the center in a white, pixelated font.

HTTP REQUEST

HTTP REQUEST

★ Indican la **acción** que se desea realizar para un recurso determinado.

HTTP REQUEST - Métodos

GET

Solicita una **representación de un recurso específico**. Las peticiones que usan el método GET *sólo deben recuperar datos*.

HEAD

Pide una **respuesta idéntica a** la de una petición **GET**, pero *sin el cuerpo de la respuesta*.

POST

Se utiliza para **enviar una entidad a un recurso en específico**, *causando* a menudo un *cambio en el estado o efectos secundarios en el servidor*.

HTTP REQUEST - Métodos

PUT

El modo PUT **reemplaza todas las representaciones actuales** del recurso de destino con la *carga útil de la petición*.

DELETE

Borra un *recurso en específico*.

HTTP REQUEST

- ☆ Aunque hay varios métodos de HTTP para recuperar datos de un servidor, **las dos más utilizados son GET y POST.**
- ☆ El método GET solicita un recurso del servidor indicado en el campo URI. Si la URI apunta a una base de datos de producción de recursos como un servlet, los *datos serán devueltos dentro del mensaje de respuesta.*
- ☆ El método POST se utiliza para *pasar explícitamente datos al servidor en el propio mensaje de solicitud.* Estos datos **estarán a disposición del recurso URI** especificado en la solicitud.

A conceptual image featuring two hands, one from the left and one from the right, holding a glowing blue sphere. The sphere is covered in binary code (0s and 1s) and has a bright light source inside, creating a lens flare effect. The background is a solid dark blue. The text 'HTTP RESPONSE' is overlaid in the center in a white, pixelated font.

HTTP RESPONSE

HTTP RESPONSE - Códigos

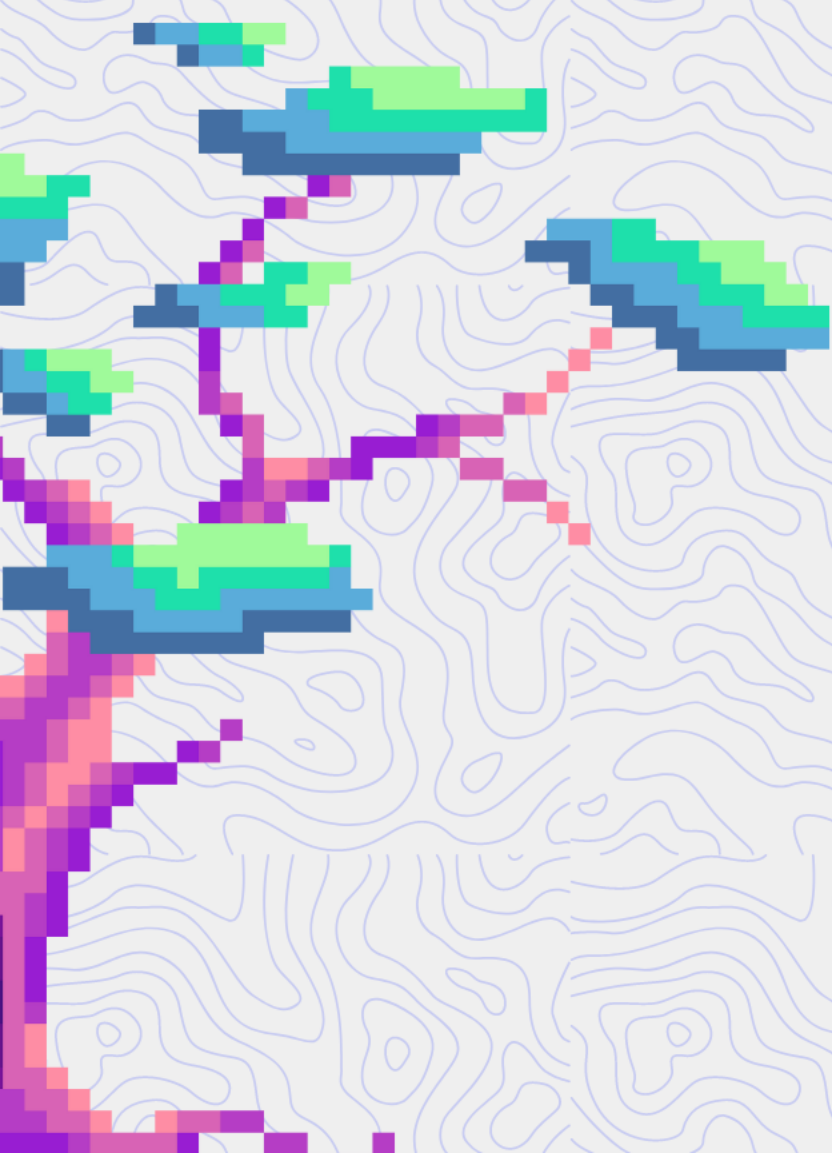
| Código de estado | Significado del valor numérico |
|------------------|---|
| 100-199 | Informativo - La solicitud fue recibida y se está procesando. |
| 200-299 | Éxito - La acción fue recibida con éxito, entendido y aceptado. |
| 300-399 | Redirección - Otras peticiones hay que realizar para completar la petición. |
| 400-499 | Error en Cliente - La solicitud contiene mala sintaxis y no pueden llevarse a cabo. |
| 500-599 | Error de servidor - El servidor no pudo cumplir con una solicitud aparentemente válida. |

HTTP RESPONSE - Respuestas más comunes

| Código | Significado |
|----------------------------------|---|
| OK-200 | La solicitud éxito |
| 302 Movido temporalmente | La solicitud de residencia temporal en un URI. Si el nuevo URI es un ubicación, la ubicación en el campo de encabezado de respuesta dará a la nueva URL. Este código normalmente se utiliza cuando el cliente se redirige |
| 400 Bad Request | El servidor no puede comprender la solicitud debido a un error de sintaxis. |
| 401 | La solicitud no autorizada requiere autenticación y / o autorización. |
| 403 Forbidden | El servidor entiende la petición, pero por alguna razón se niega a cumplirlo. El servidor puede o no puede revelar la razón por la que se niega la petición. |
| 404 Not Found | El servidor no ha encontrado nada que coinciden con el URI de solicitud. |
| 500-Internal Server Error | "El servidor encontró un inesperado condición que le impidió el cumplimiento de la petición. |



BIBLIOGRAFÍA



★ <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>

★ <https://aws.amazon.com/es/what-is/api/>

★ <https://www.ibm.com/docs/en/cics-ts/5.3?topic=concepts-http-protocol>

★ <https://www.ibm.com/docs/en/cics-ts/5.3?topic=protocol-http-requests>

★ <https://developer.mozilla.org/es/docs/Web/HTTP/Methods>

★ <https://sites.google.com/site/conceptoprogramacion/request-response>