



Ayudantía 3: **Consumo de APIs**

Juan Fernández, Pedro Ríos

Contenidos

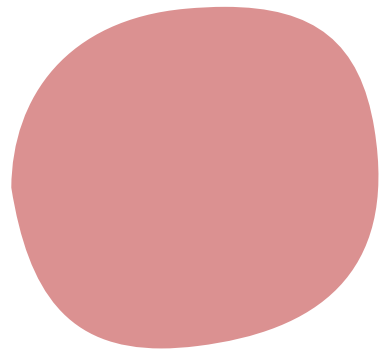
01 API/RESTful

**04 Códigos de estado de
respuesta HTTP**

02 Métodos HTTP

05 Postman

03 Headers y body



¿Qué es una API?





¿Qué es una API?

Es un **intermediario** que define cómo deben interactuar las aplicaciones entre sí, permitiendo acceso controlado a sus funciones y datos.





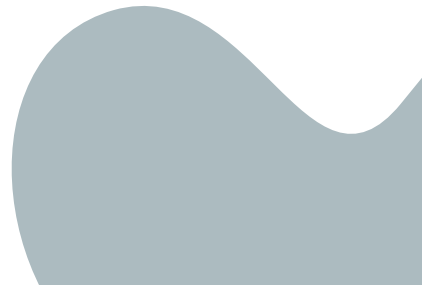
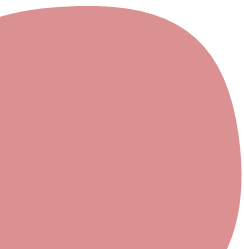
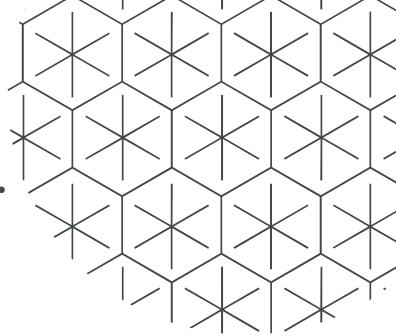
¿Qué es una API?

Una API **expone** una lista de funciones, datos o servicios que una aplicación puede solicitar mediante solicitudes HTTP.



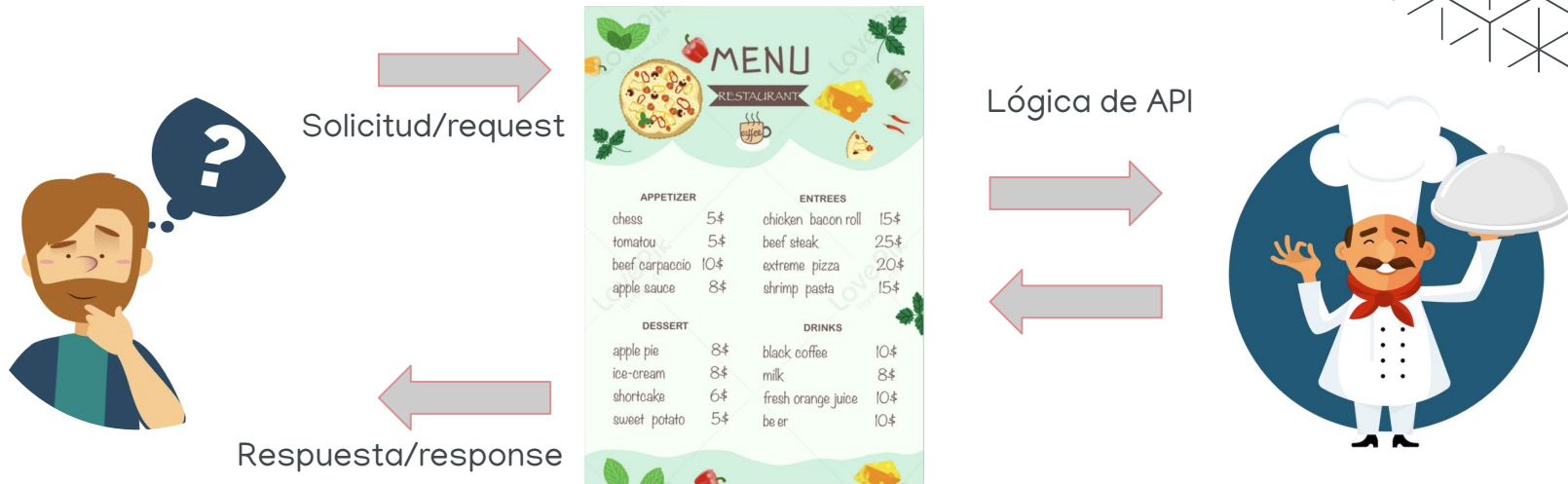
¿Qué es una API?

Pueden pensar que una API es como un menú de un restaurante.



¿Qué es una API?

Pueden pensar que una API es como un menú de un restaurante.



¿Qué es una API?

Pueden pensar que una API es como un menú de un restaurante.



De esta forma, las aplicaciones pueden pedir y compartir datos o funcionalidades entre sí.

¿Qué es REST?

REST (REpresentational State Transfer) es un conjunto de principios y convenciones que debe cumplir un servicio web.

Los servicios REST utilizan los métodos HTTP, como **GET/POST/PUT/DELETE**, para realizar operaciones sobre los recursos (objetos, documentos, o datos).

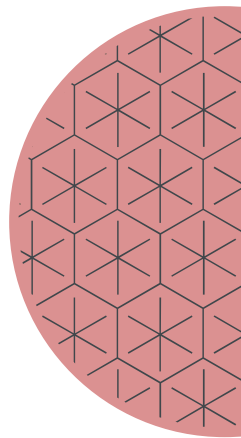
Principios REST

- **Protocolo Cliente–Servidor:** el cliente y servidor deben estar separados y poder evolucionar independientemente.
- **Sin estado:** cada solicitud del cliente al servidor debe contener toda la información necesaria para comprender y procesar la solicitud.
- **Cacheable:** el servicio debe poder indicar si una respuesta es cacheable. El caché del cliente puede reciclar los datos de respuesta para solicitudes similares en el futuro.
- **Interfaz uniforme:** identificación única de recursos mediante URL, manipulación a través de representaciones (como JSON o XML), respuestas autodescriptivas y uso de hipermedia para guiar al cliente.
- **Sistema de capas:** los componentes están separados en capas, evitando la interacción entre capas y restringiendo su visibilidad más allá de la capa inmediata.

Métodos HTTP en REST

Los métodos HTTP son acciones que permiten a las aplicaciones interactuar con recursos en una API REST.

- **GET:** recuperar información de un recurso
- **POST:** crear un nuevo recurso
- **PUT:** actualizar un recurso existente por completo
- **PATCH:** actualizar parcialmente un recurso existente
- **DELETE:** eliminar un recurso



Métodos HTTP en REST

GET

`/pet/{petId}` Find pet by ID

PUT

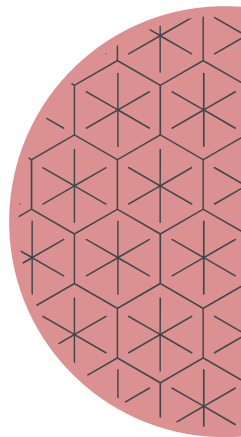
`/pet` Update an existing pet

DELETE

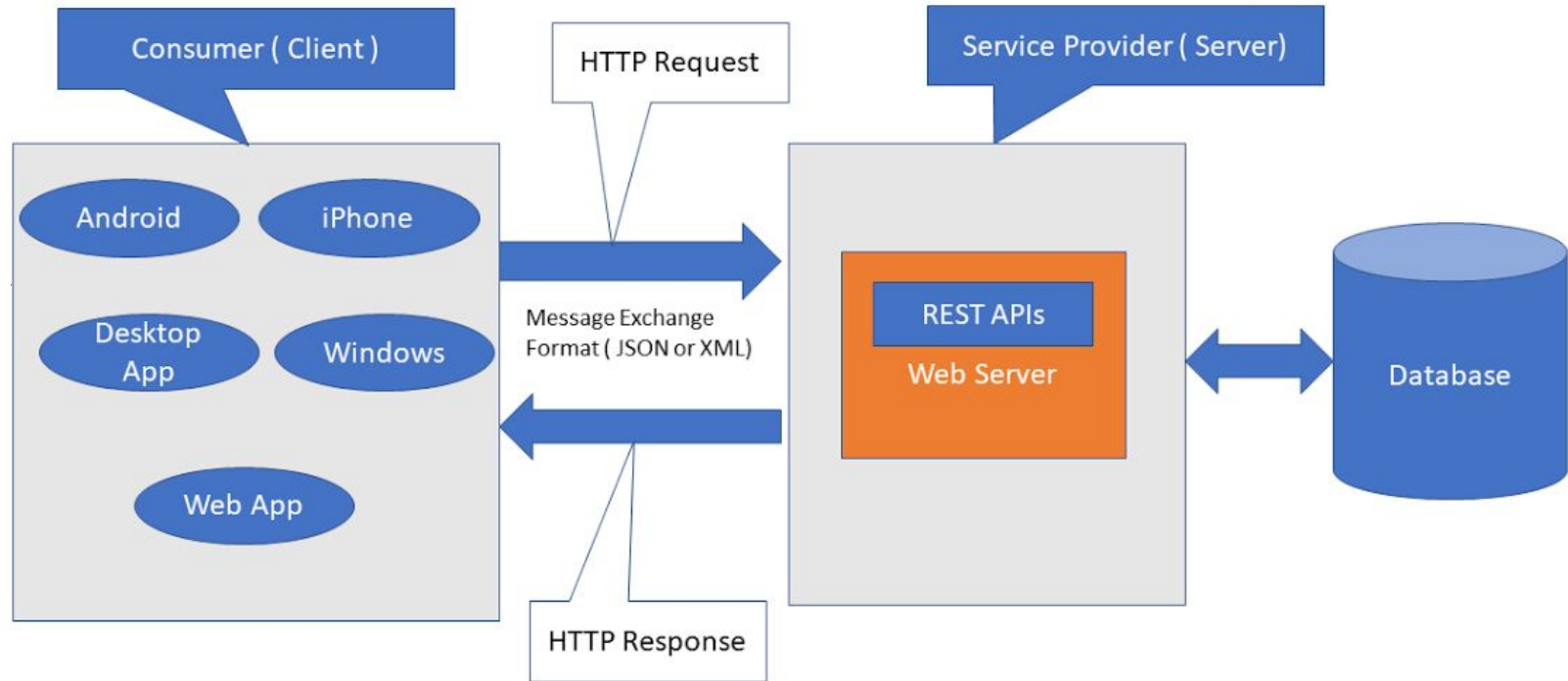
`/pet/{petId}` Deletes a pet

POST

`/pet/{petId}/uploadImage` uploads an image



Arquitectura API RESTful



Contenidos

01 API/RESTful

04 Códigos de estado de
respuesta HTTP

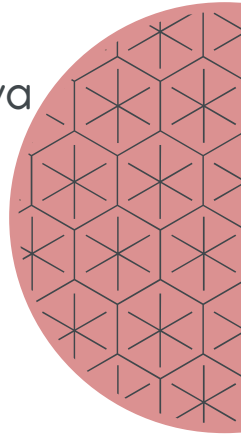
02 Métodos HTTP

05 Postman

03 Headers y body

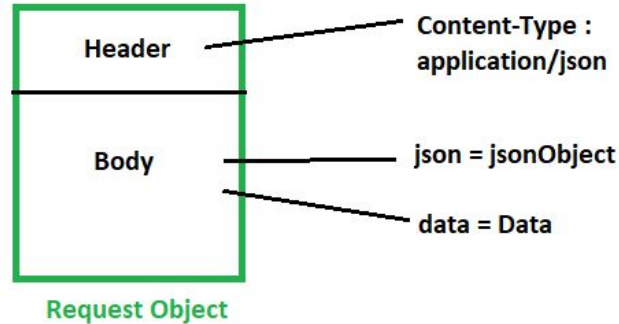
Headers y Body

Los Headers y el Body son componentes fundamentales de las solicitudes y respuestas HTTP que permiten la comunicación efectiva entre el cliente y el servidor en la web.



Headers

Se encuentran en la parte superior de la solicitud o respuesta y contienen metadatos clave que proporcionan información crucial sobre la comunicación entre el cliente (por ejemplo, un navegador web o una aplicación) y el servidor web.

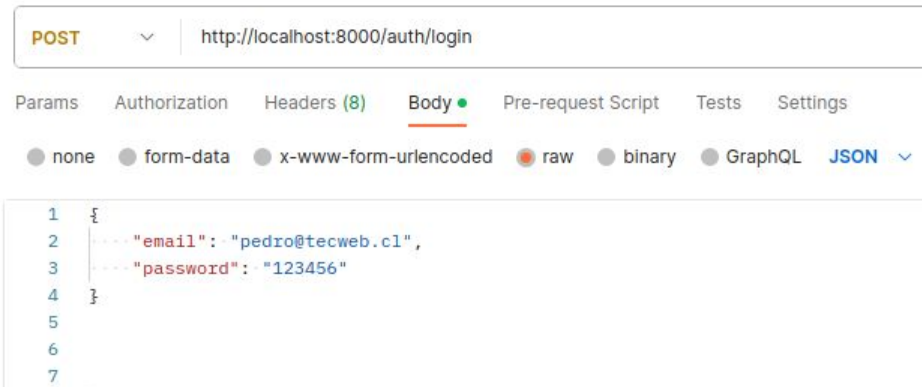


Principales Headers

1. **Content-Type:** Este encabezado especifica el tipo de contenido que se encuentra en el cuerpo de la solicitud o respuesta.
2. **Authorization:** El encabezado de autorización se utiliza para enviar credenciales de autenticación al servidor. Esto es esencial cuando se necesita autenticar el acceso a recursos protegidos, como contraseñas o tokens de acceso.
3. **Accept:** Indica los tipos de contenido que el cliente está dispuesto a aceptar como respuesta. Ayuda al servidor a seleccionar la mejor representación del recurso solicitado si hay múltiples opciones disponibles.

Body

El Body, o cuerpo de la solicitud o respuesta HTTP, es donde se encuentran los datos reales que se envían entre el cliente y el servidor. La utilidad del cuerpo radica en la capacidad de transmitir información más allá de simples metadatos.



Aspectos relevantes

- 1. Envío de datos:** El cuerpo de una solicitud se utiliza para enviar datos desde el cliente al servidor. Por ejemplo, en una solicitud POST para crear una nueva entrada en un blog, el cuerpo puede contener el texto del artículo que se está creando.
- 2. Recepción de datos:** El cuerpo de una respuesta se utiliza para transmitir datos desde el servidor al cliente. Esto puede incluir el contenido de una página web, una imagen, un archivo JSON con resultados de una consulta, etc.

Aspectos relevantes

- 1. Formato de datos:** El cuerpo puede contener datos en diversos formatos, como JSON, XML, HTML, texto sin formato, imágenes, videos y más. La elección del formato depende del tipo de datos que se esté transmitiendo.
- 2. Estado y respuestas:** En las respuestas del servidor, el cuerpo a menudo contiene información sobre el estado de la solicitud, como códigos de estado (por ejemplo, 200 OK o 404 Not Found) y mensajes descriptivos que ayudan a los clientes a comprender la respuesta

Contenidos

01 API/RESTful

04 Códigos de estado de
respuesta HTTP

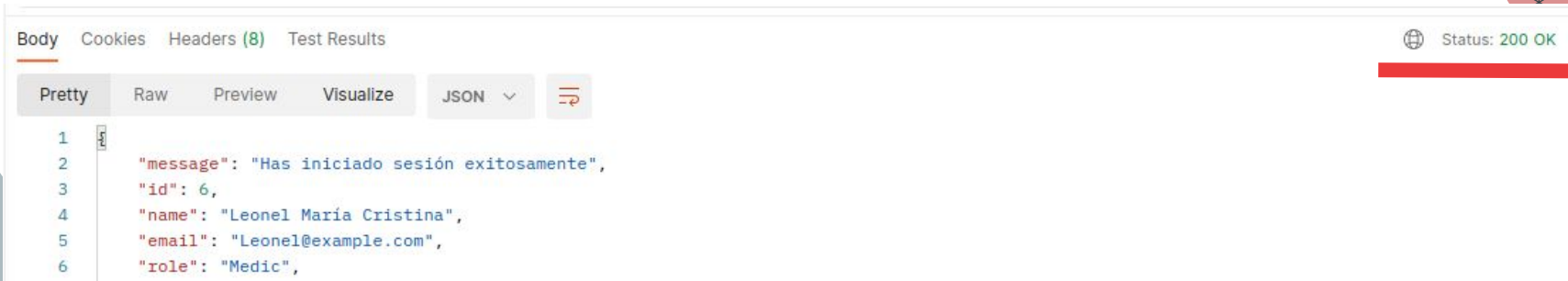
02 Métodos HTTP

05 Postman

03 Headers y body

Códigos de respuesta

Los códigos de respuesta HTTP son una parte fundamental del protocolo HTTP (Hypertext Transfer Protocol), que es el protocolo utilizado para la comunicación entre clientes (como navegadores web o aplicaciones) y servidores web. Estos códigos se utilizan para proporcionar información sobre el resultado de una solicitud realizada por un cliente a un servidor web.



Principales códigos

1xx: Respuestas informativas.

2xx: Respuestas exitosas.

3xx: Redirecciones.

4xx: Errores del cliente.

5xx: Errores del servidor.

Algunos ejemplos comunes de códigos de respuestas incluyen el código 200 (OK, solicitud exitosa), 404 (No encontrado, recurso no disponible), y 500 (Error interno del servidor).

[Lista completa de códigos \[VER\]](#)

Contenidos

01 API/RESTful

04 Códigos de estado de
respuesta HTTP

02 Métodos HTTP

05 Postman

03 Headers y body

Postman

Postman es una herramienta popular utilizada por desarrolladores y equipos de desarrollo de API para probar, documentar y automatizar solicitudes a APIs. Ofrece una interfaz de usuario intuitiva que permite a los usuarios crear y enviar solicitudes HTTP a servidores, inspeccionar las respuestas, y colaborar en el desarrollo de API



POSTMAN

Postman

Nos conectaremos a una API pública y la consumiremos usando el método GET.



POSTMAN

