APIS

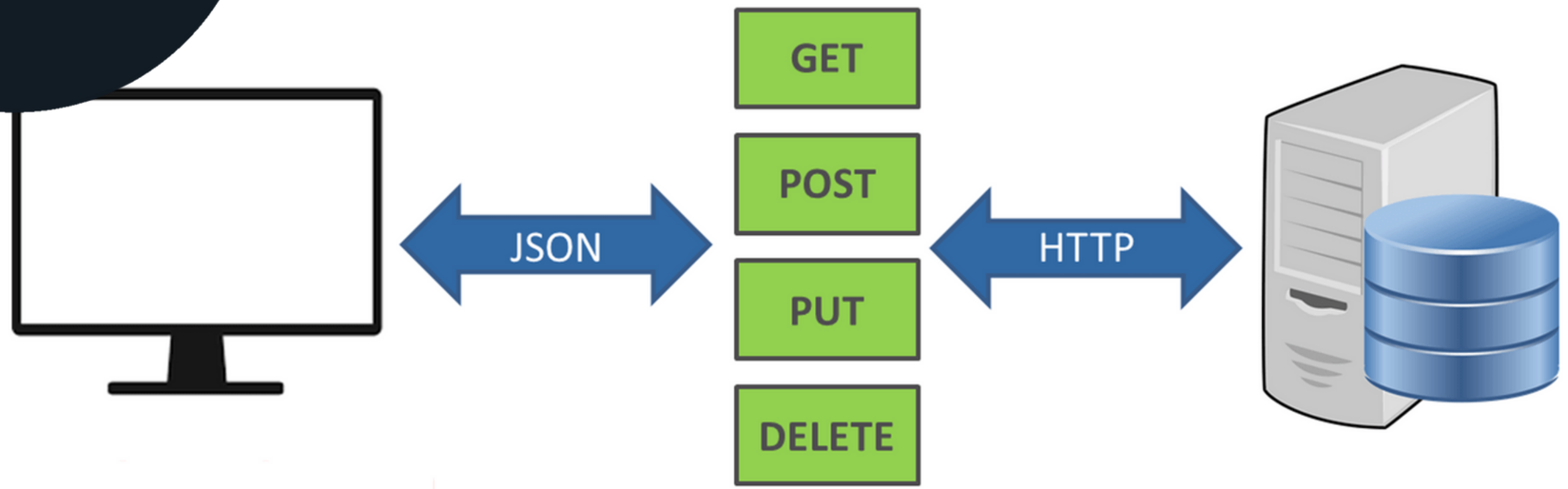
PROGRAMACIÓN III - TUP UTN FRLR

¿Qué es una API?

El término API es una abreviatura de **Application Programming Interfaces**, que en español significa **Interfaz de Programación de Aplicaciones**.

Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.

{ api }

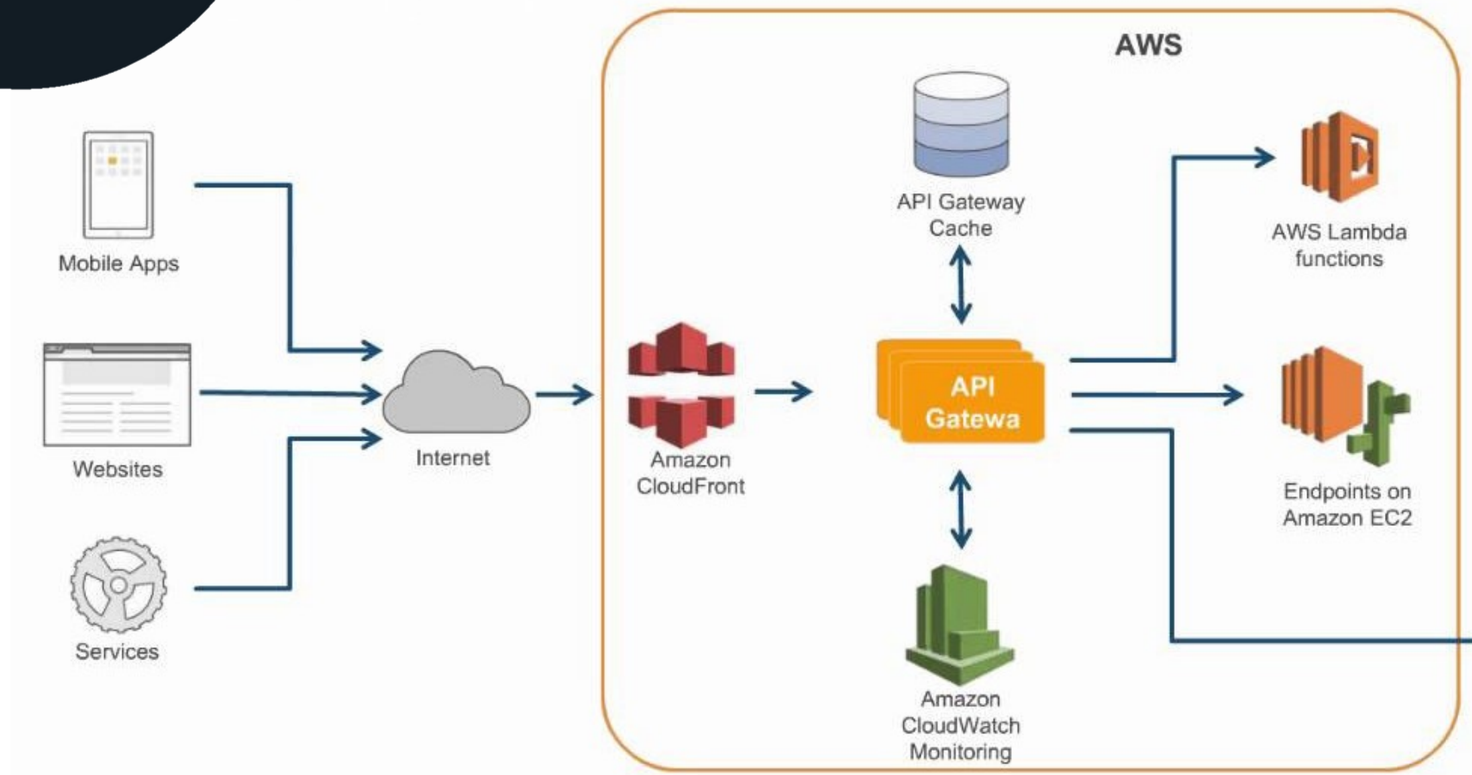


Client sends a **request**

HTTP methods

Server sends a **response**

{ api }

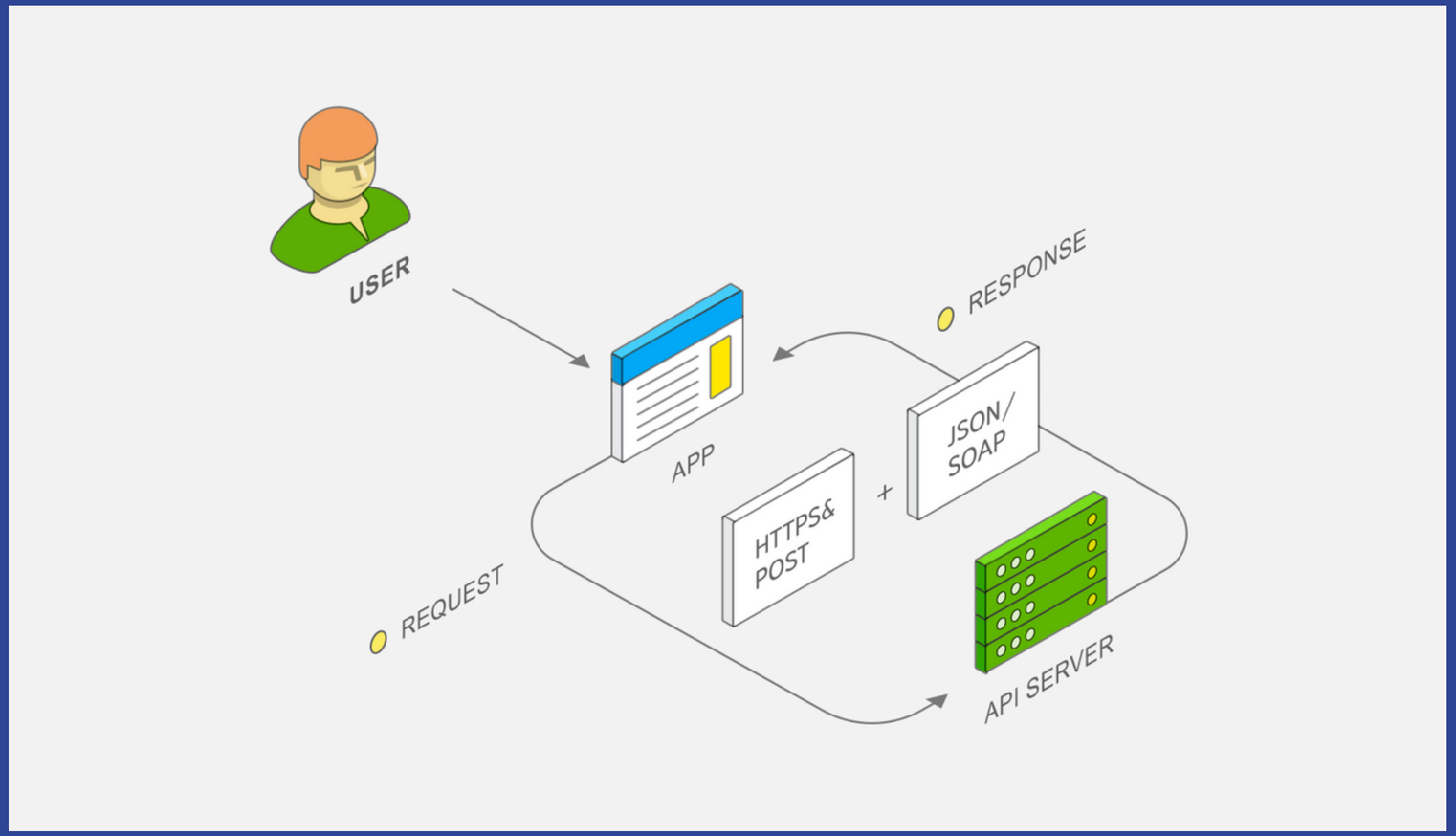

AWS
Mobile Apps
Websites
Services
Internet
Amazon CloudFront
API Gateway Cache
API Gatewa
Amazon CloudWatch Monitoring
AWS Lambda functions
Endpoints on Amazon EC2
Any other publicly accessible endpoint

{ api }

Así pues, podemos hablar de una API como una especificación formal que establece cómo un módulo de un software se comunica o interactúa con otro para cumplir una o muchas funciones.

"Todo dependiendo de las aplicaciones que las vayan a utilizar, y de los permisos que les dé el propietario de la API a los desarrolladores de terceros".

{ api }



Las API pueden tener tanto una como varias funciones, pudiendo llegar a ser auténticos kits de herramientas. Cuando esto pasa, una aplicación puede enviarle una solicitud con una estructura particular, y esta estructura determinará cómo responderá el servicio o el software al que le estés enviando esa solicitud.

{ api }

PRIVADAS - LOCALES

Pueden ser privadas para el uso de una empresa, abiertas sólo para partners, o públicas para que cualquier desarrollador interactuar con ellas o crear sus propias API para que lo hagan.

También pueden ser API locales para aplicaciones que se comunican dentro de un mismo ambiente o dispositivo, o remotas para cuando hay que acceder a otro punto diferente.

{ api }

ENTAJAS

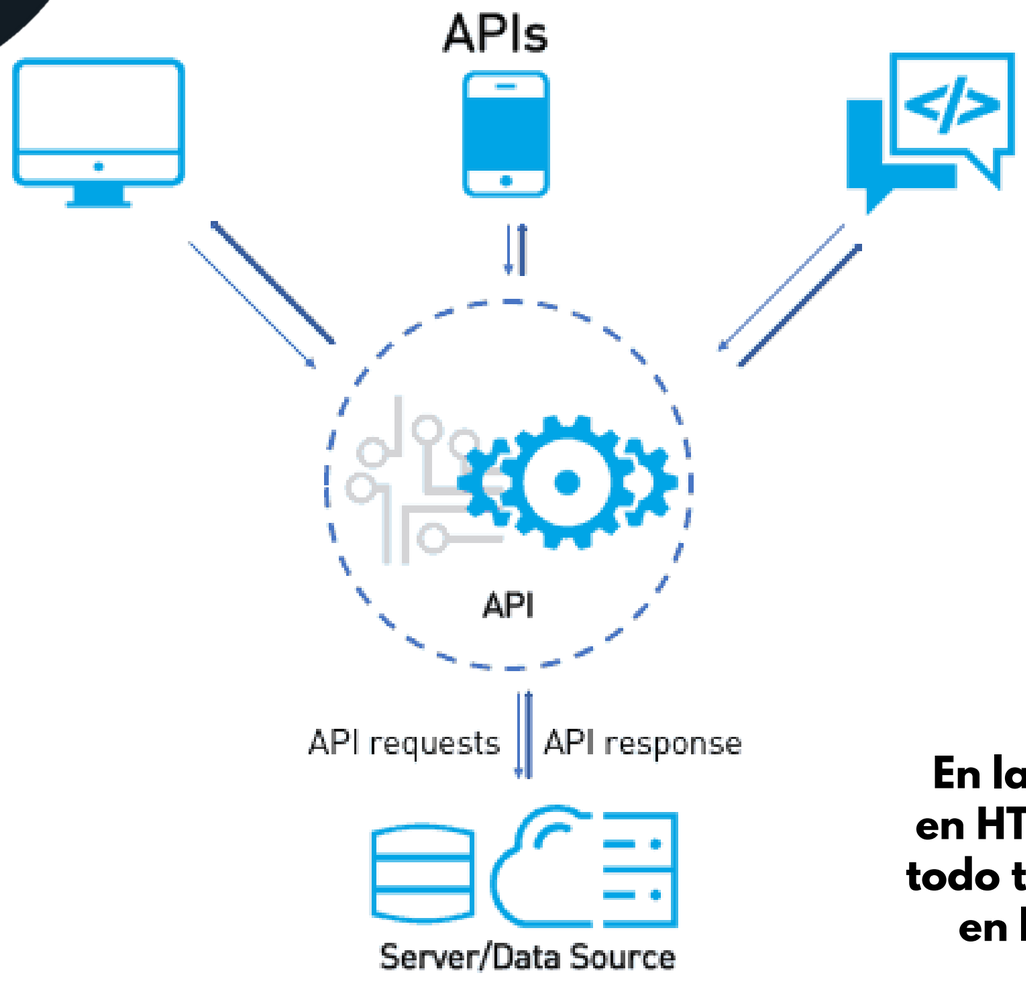
- Una de las principales funciones de las API es poder facilitarle el trabajo a los desarrolladores y ahorrarles tiempo y dinero.

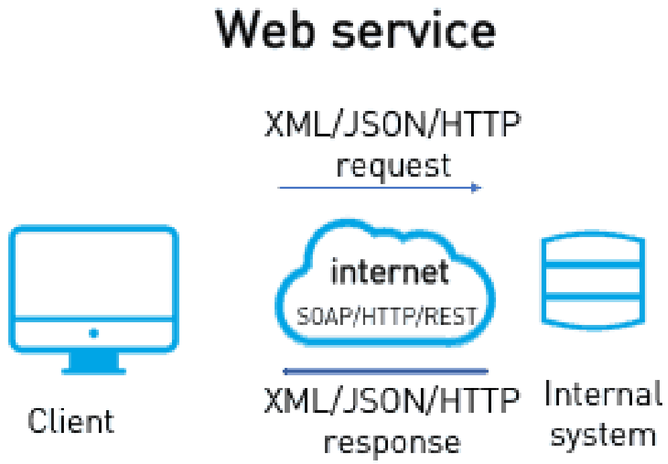
- No será necesario tener que reinventar la rueda con cada servicio que se crea, ya que podrás utilizar piezas o funciones que otros ya han creado.

- También son útiles para cuando lo único que se quiere es utilizar deliberadamente las funciones de determinado servicio para ofrecer ventajas a sus usuarios o atraer a los usuarios de ese servicio a que utilicen tu aplicación.

APIs de Servicios Web

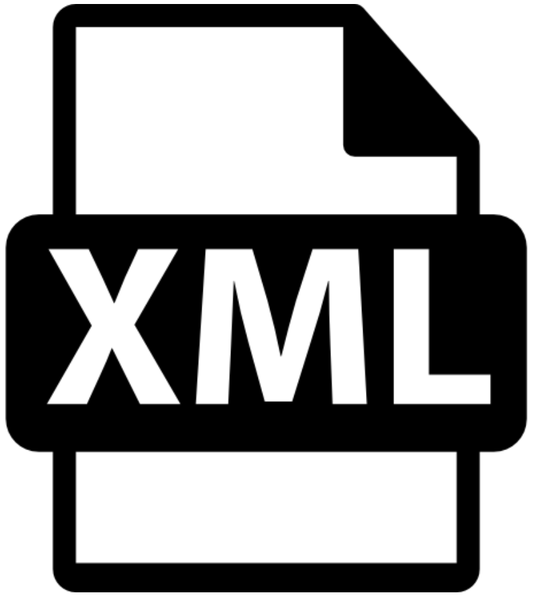
Son las interfaces de desarrollo de aplicaciones que permiten el intercambio de información entre un servicio web (software que da acceso a un servicio concreto a través de una URL) y una aplicación. Normalmente ese intercambio se produce a través de peticiones HTTP o HTTPS (la versión cifrada del protocolo HTTP).





En la petición de la aplicación y respuesta, también en HTTP del servicio web, se contiene información de todo tipo tanto en los metadatos de la cabecera como en los del mensaje, normalmente en dos tipos de formatos muy usados: XML o JSON.

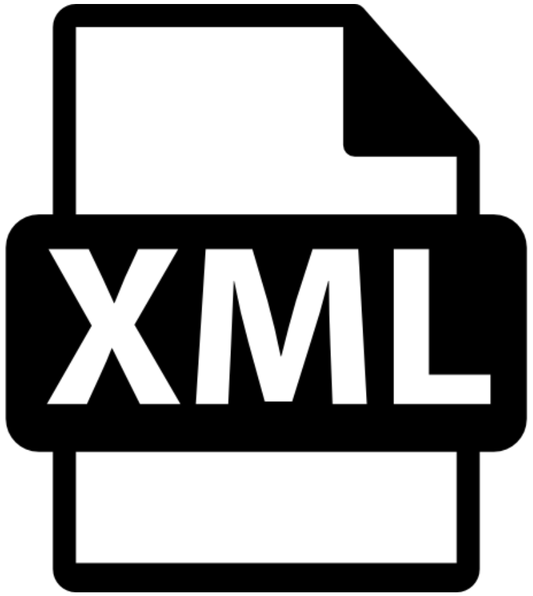
{ api }



TIPOS DE API - WEBSERVICES

SOAP (Simple Object Access Protocol), un protocolo estándar de intercambio de información y datos en XML entre dos objetosDeslizamiento de imágenes.

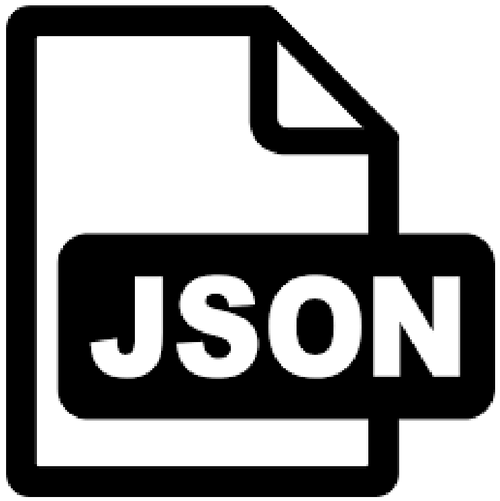
{ api }



TIPOS DE API - WEBSERVICES

XML-RPC, un protocolo de llamada a procedimiento remoto que usa XML como formato de datos y llamadas HTTP como sistema de comunicación;

{ api }



TIPOS DE API - WEBSERVICES

JSON-RPC, mismo protocolo pero en formato JSON;

{ api }

REST API

TIPOS DE API - WEBSERVICES

REST (Representational State Transfer), arquitectura de software para sistemas hipermedia en la World Wide Web; una API REST usa el protocolo HTTP.

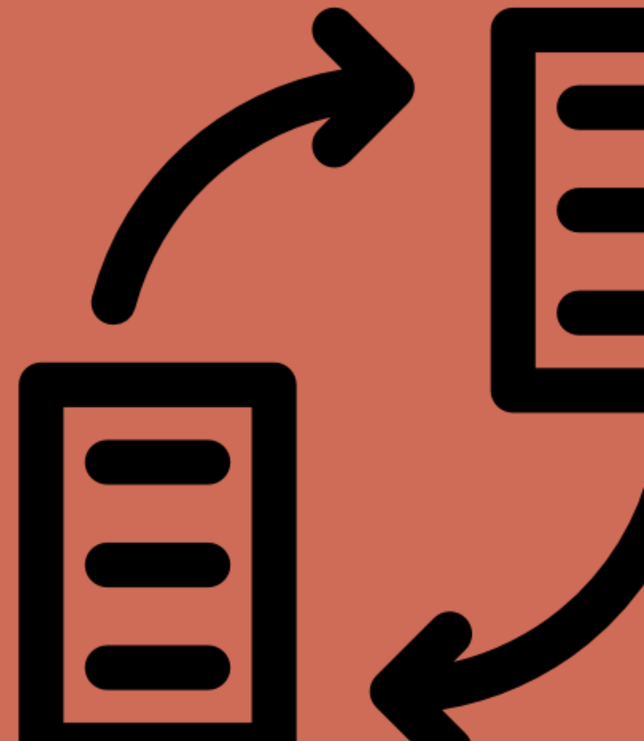
Las empresas punteras en innovación reposan gran parte de su músculo creativo en una o varias APIs. Google, Facebook, Twitter, Square, Stripe, Amazon... Empresas de gran volumen o outsiders dentro de su sector. Da igual que sean dentro del mercado publicitario, tecnológico, bancario... Las APIs son el elemento clave en cualquier proceso innovador.

{ api }

REQUEST I RESPONSE

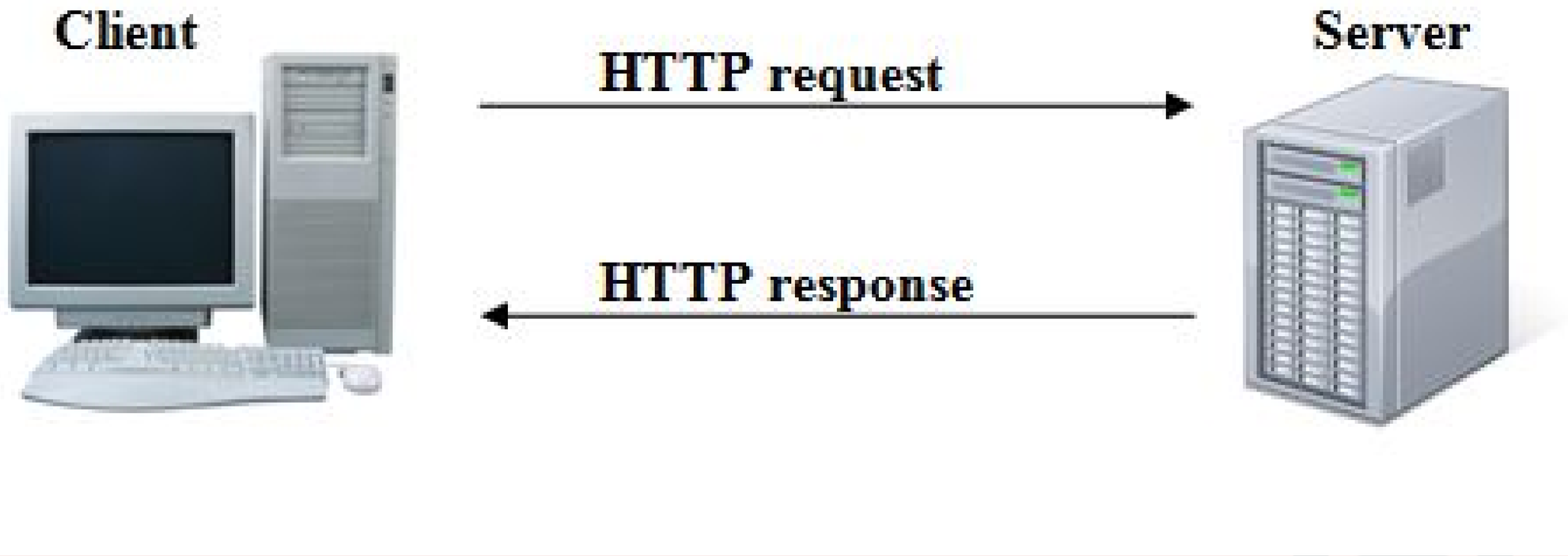
Como todo protocolo de comunicación, el protocolo HTTP define unas normas que debe seguir el emisor que desea transmitir información a un receptor.

HTTP utiliza el patrón de request / response (petición / respuesta), lo cual quiere decir, que un emisor (sistema X) realiza una solicitud (request) y un servidor (Sistema Y) responde (response) a dicha solicitud.



{ api }

REQUEST I RESPONSE



{ api }

REQUEST I RESPONSE

POST /v1/customers HTTP/1.1 <-- Linea de Solicitud
Host: api.midominio.com <-- Encabezados ó Headers
Accept: application/json
Content-Type: application/json
Content-Length: 25
{ <-- Cuerpo ó Body Request
"name": "Pepito S.A.C",
"customer_category": "Grandes Empresas"
}

{ api }

REQUEST I RESPONSE

La primera línea se llama Línea de Solicitud y contiene tres elementos;

1. El método HTTP (para este ejemplo es POST).
2. Una ruta de acceso (también denominada como endpoint en una API).
3. La versión HTTP (para este ejemplo es HTTP/1.1)

El contenido que le sigue a la «Línea de Solicitud» se denomina **Headers** (en español encabezados). Los headers son pares de key – value (clave – valor), donde la clave es la primera palabra, seguido de dos puntos y un espacio y, a continuación, el valor.

{ api }

MÉTODOS - API REST

Entender los métodos HTTP es fundamental para comprender la forma en que funciona la arquitectura REST, pues mediante los métodos le indicamos al servidor la forma en que debe de tratar una determinada petición, dicho esto, una misma URL puede ser tratada de forma diferente por el servidor.

REST API

{ api }

GET

Es utilizado únicamente para consultar información al servidor, muy parecidos a realizar un SELECT a la base de datos. No soporta el envío del payload.

REST API

{ api }

POST

Es utilizado para solicitar la creación de un nuevo registro, es decir, algo que no existía previamente, es decir, es equivalente a realizar un INSERT en la base de datos. Soporta el envío del payload.

REST API

{ api }

PUT

Se utiliza para actualizar por completo un registro existente, es decir, es parecido a realizar un UPDATE a la base de datos. Soporta el envío del payload.

{ api }

PATCH

Este método es similar al método PUT, pues permite actualizar un registro existente, sin embargo, este se utiliza cuando actualizar solo un fragmento del registro y no en su totalidad, es equivalente a realizar un UPDATE a la base de datos. Soporta el envío del payload

REST API

{ api }

DELETE

Este método se utiliza para eliminar un registro existente, es similar a DELETE a la base de datos. No soporta el envío del payload.

{ api }

HEAD

Este método se utilizar para obtener información sobre un determinado recurso sin retornar el registro. Este método se utiliza a menudo para probar la validez de los enlaces de hipertexto, la accesibilidad y las modificaciones recientes.

REST API

Consulta de todos los usuarios (@GET)

Crear un nuevo usuario (@POST)

Actualización de un usuario existente (@PUT)

Eliminar un usuario (@DELETE)

{ api }

REQUEST I RESPONSE

Una solicitud HTTP (HTTP Request) solo necesita tener una Línea de Solicitud y algunos headers para ser válido. Esto se aplica principalmente a las solicitudes que utilizan los métodos HTTP GET, DELETE, HEAD y OPTIONS.

Sin embargo, también podemos proporcionar un cuerpo de solicitud (request body), que requiere dos líneas nuevas después del último header antes del contenido del cuerpo. Los cuerpos se utilizan principalmente con los métodos HTTP POST, PUT y PATCH.

{ api }

ENDPOINTS

/companies
/companies/{company_id}
/employees
/employees/{employee_id}
/companies/{company_id}/employees
/companies/{company_id}/employee/{employee_id}

¿DUDAS?