DigitalHouse >
Coding School

HTTP

“

HTTP es un **protocolo de transferencia** que permite de manera estandarizada la comunicación entre el **cliente** y el **servidor**.

DigitalHouse >
Coding School

REST

“

REST es un tipo de arquitectura de servicios que **proporciona estándares** entre sistemas informáticos para establecer **cómo** se van a comunicar entre sí.

REST:

CLIENTE - SERVIDOR

CLIENTE - SERVIDOR

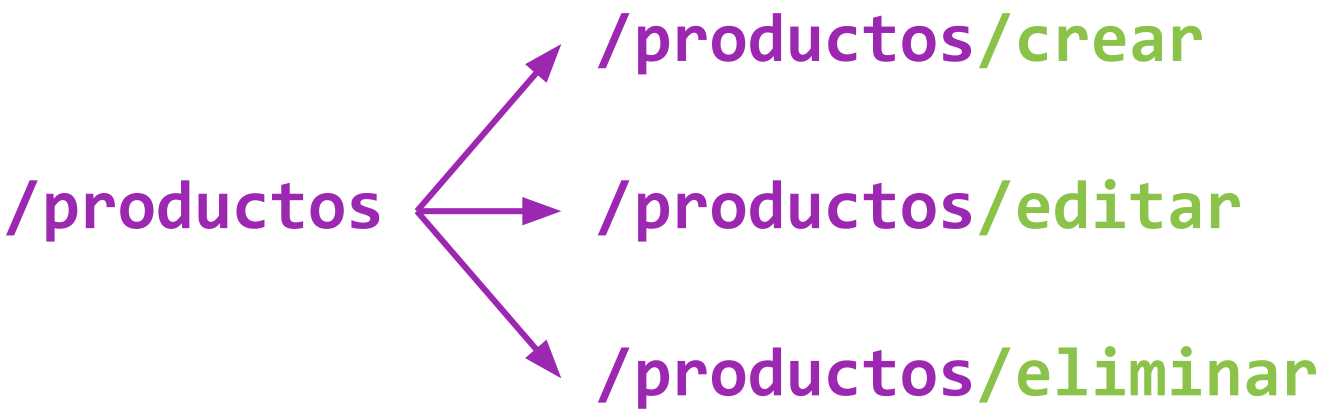
Uno de estos estándares es el de **cliente - servidor** el cual indica que la aplicación del cliente y la aplicación del servidor deben poder desarrollarse, extenderse, cambiarse, sin interferir una con otra.

El cliente nunca debería enterarse que se cambió algo del servidor y el servidor no debería tener ningún inconveniente con cambios que realice el cliente.

Lo único que el cliente debería conocer del servicio son los accesos a los recursos (endpoints o URIs)

RECURSOS UNIFORMES

Un recurso en el sistema debería tener solamente un identificador lógico, y éste proveer acceso a toda la información relacionada.



STATELESS

El paradigma REST propone que todas las interacciones entre Cliente y Servidor deben ser tratadas como nuevas y de forma absolutamente independiente.

Por lo tanto, si quisiéramos tener una aplicación que distinga de usuarios logueados o invitados, debemos mandar toda la información de autenticación necesaria en cada petición.

Esto permite desarrollar aplicaciones más **confiables**, **performantes** y **escalables**, ya que permite que puedan ser **dirigidas** y **actualizadas** incluso mientras estas se encuentra funcionando.

CACHEABLE

En el paradigma de REST el **cacheo de datos** es una herramienta muy importante, que se implementa del lado del cliente, para mejorar la performance y reducir la demanda al servidor.

El Caché debe aplicarse a los recursos tanto como sea posible.

“

El **cacheo de datos** se utiliza para guardar información que no suele cambiar usualmente, en el equipo del **cliente**. Esto mejora el rendimiento y reduce las peticiones al **servidor**.

DigitalHouse >
Coding School

FORMATOS DE ENVÍO DE DATOS

JSON

Es el formato más común para el envío de datos.

Cuando queramos enviar datos en formato JSON debemos agregar un encabezado en los headers que diga:

"Content-Type": "application/json"

RAW

Se utiliza para mandar datos con texto sin ningún formato en particular.

No suele ser tan utilizado.

TEXT

Se utiliza para enviar datos que no sean en formato JSON como archivos html, css.

URL-encoded

Indica que se nos van a enviar datos codificados en forma de URL. Por lo tanto, nos envía algo muy similar a un query string.

Un dato enviado mediante este método se vería de la siguiente manera:

email%3Dcosme%40fulanito.fox
%26password%3Dverysecret

DigitalHouse >
Coding School

MÉTODO HEAD

HEAD

Es un método HTTP que permite conocer la **fecha de la última modificación** de un recurso.

Se utiliza en conjunto con el método GET, de tal manera que si la fecha de modificación no coincide con la del recurso que el cliente tiene cacheada localmente se puedan pedir los datos para actualizarlos.

“

Existen más métodos para diseñar una arquitectura o api rest.

Podés investigarlos escaneando el código qr!