

Diseño API RESTful

ARQUITECTURA



Para empezar

¿Qué es una API?

Es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones.

Reglas de arquitectura REST



Interfaz Uniforme

- La interfaz se basa en los recursos.
- El servidor mandará los datos (via html, json, xml...)



Peticiones sin estado

Http es un protocolo sin estado ---> mayor rendimiento.



Cacheable

En la web los clientes pueden cachear las respuestas del servidor.



Separación de cliente y servidor

El cliente y servidor están separados, su unión es mediante la interfaz uniforme.



Sistema de capas

El uso de capas o servidores intermedios puede servir para aumentar la escalabilidad.



Código bajo demanda

Los servidores pueden ser capaces de aumentar o definir cierta funcionalidad en el cliente transfiriéndole cierta lógica.

Buenas Prácticas RESTful

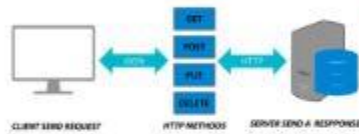
¡LO QUE DEBES SABER!



Debe ser agradable para el uso

PARA ESÓ:

- Debe usar estándares Web donde tenga sentido.
- Debe ser amigable con el desarrollador.
- Debe ser explorable con el navegador.
- Debe ser sencilla e intuitiva.
- El API debe ser consistente.



Identificar los recursos

Los Recursos abstraen y aíslan de detalles de implementación.



Identificar las acciones

Una vez identificados los recursos identificaremos las acciones que aplican a estos y cómo mapean al API.



Usar URLs y verbos RESTful

Los recursos se manipulan con peticiones HTTP con los verbos GET, POST, UPDATE y DELETE.

Metodos HTTP

ACCIONES:

Una de las ventajas de RESTful es que no es necesaria una convención de metodos para poder invocar el API y que la estructura de las URLs es clara y limpia.



<https://view.genial.ly/5eb9e6aae16e4d0d7cbc9be3/vertical-infographic-buenas-practicas-restful>



<https://view.genial.ly/5ebae577ccdd4d0db34d835a/vertical-infographic-documentacion-de-apis>