

# Introducción

El Consorcio World Wide Web (W3C) es una comunidad internacional donde las organizaciones Miembro desarrollan estándares Webs, y han definido los Web Services como sistemas de software creados para soportar una interacción interoperable equipo a equipo sobre una red.

Los Web Services suelen ser APIs Web que pueden ser accedidos dentro de una red (principalmente Internet) y son ejecutados en el sistema donde residen. El estilo de arquitectura que hemos utilizado es el REST (Representational State Transfer), de tal manera que los servicios web deben cumplir con los siguientes principios:

- Un **protocolo cliente/servidor sin estado**: cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Por lo que ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URLs, no son permitidas por REST).
- Un conjunto de **operaciones bien definidas** que se aplican a todos los recursos de información: HTTP en sí define un conjunto pequeño de operaciones, las más importantes son **POST, GET, PUT y DELETE**. Con frecuencia estas operaciones se equiparan a las operaciones **CRUD** en bases de datos (ABMC: Alta, Baja, Modificación y Consulta) que se requieren para la persistencia de datos, aunque POST no encaja exactamente en este esquema.
- Una **sintaxis universal** para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su URI.
- El **uso de hipermedios**, tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST **son típicamente HTML o XML**. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

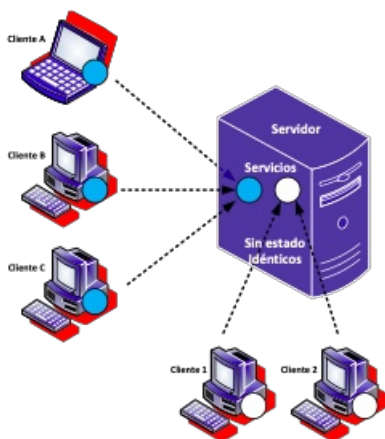
## Cliente/Servidor

Como servicios web, son cliente servidor y definen una interfaz de comunicación entre ellos, separando completamente las responsabilidades entre ambas partes.



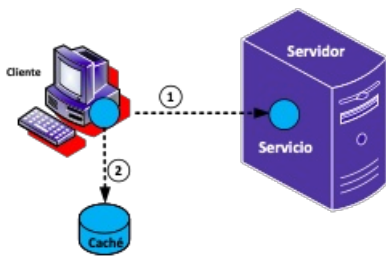
## Sin estado

Son servicios web que no mantienen estado asociado al cliente. Cada petición que se realiza a ellos es completamente independiente de la siguiente. Todas las llamadas al mismo servicio serán idénticas.



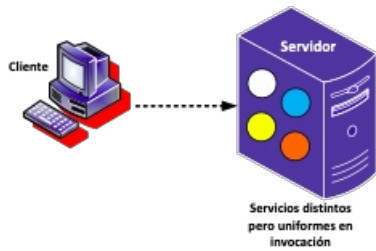
## Caché

El contenido de los servicios web REST se puede cachear de tal forma que, una vez realizada la primera petición al servicio, el resto puedan apoyarse en el caché si fuera necesario.



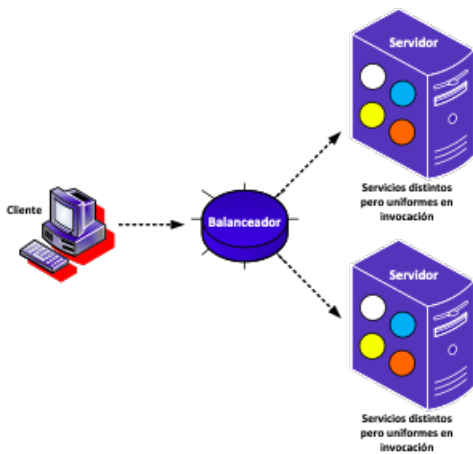
## Servicios Uniformes

Todos los servicios REST compartirán una forma de invocación y métodos uniforme utilizando los métodos GET, POST, PUT y DELETE.



## Arquitectura en capas

Todos los servicios REST están orientados hacia la escalabilidad y un cliente REST no será capaz de distinguir entre si está realizando una petición directamente al servidor, o se lo está devolviendo un sistema de cachés intermedio; o, por ejemplo, existe un balanceador que se encarga de redirigirlo a otro servidor.



🕒 Revision #2

★ Created Wed, Mar 20, 2019 2:35 PM by [Tralix México](#)

✎ Updated Wed, Mar 20, 2019 3:38 PM by [Tralix México](#)