REST (Representational State Transfer, que significa Transferencia Representacional de Estado) es un modelo de arquitectura y no un lenguaje o tecnología de programación, que proporciona pautas para que los sistemas distribuidos se comuniquen directamente utilizando principios y protocolos web existentes sin necesidad de [SOAP](https://es.wikipedia.org/wiki/Simple_Object_Access_Protocol) u otro protocolo sofisticado.

**El origen del termino REST**

Los conceptos de **REST** se presentaron en las tesis doctoral de [Roy Fielding](https://en.wikipedia.org/wiki/Roy_Fielding) en la década de los 2000, dónde el principio fundamental es utilizar el [protocolo HTTP](https://app.aluracursos.com/course/http-base-internet) para la comunicación de datos.

Puedes encontrar en este artículo:

* El origen del termino REST
* Responsabilidades de REST
* Solicitudes y comunicaciones
* Métodos HTTP
* Códigos de respuestas
* Conclusión

La arquitectura REST es simple y brinda acceso a los recursos para que el cliente REST acceda y represente los recursos en el lado del cliente. El estilo REST, URI o ID globales ayudan a identificar cada recurso.

Esta arquitectura utiliza varias representaciones de recursos para representar su tipo como XML, JSON, texto, imágenes,etc.

Vale la pena señalar que REST no se limita a registrar solicitudes y respuestas. También es posible insertar un nuevo registro o eliminar uno ya existente.

**Responsabilidad REST**

Existe un principio en REST llamado STATELESSNESS (sin estado), dónde el servidor no necesita saber en qué estado se encuentra el cliente y viceversa. Pero, ¿qué es un servidor y un cliente?

**Cliente:** Es el componente solicitante de un servicio que envía solicitudes de varios tipos de servicios al servidor.

**Servidor:** Es el componente que es el provedor de servicios y propociona continuamente servicios al cliente según las solicitudes.

En esta arquitectura el modelo, cliente-servidor ayuda en la separación de responsabilidades entre la interfaz de usuario y el almacenamiento de datos. Es decir, cuando se realiza una solicitud REST, el servidor envía una representación de los estados que se solicitaron.

No existe un límite tope en la cantidad de clientes que un solo servidor puede atender. Tampoco es obligatorio que el cliente y el servidor se encuentren en sistemas separados.

La comunicación entre el cliente y el servidor se lleva a cabo mediante el intercambio de mensajes utilizando un patrón de solicitud-respuesta. El cliente básicamente envía una solicitud de servicio y el servidor devuelve una respuesta.

**Solicitudes y comunicaciones**

REST requiere que un cliente realice una solicitud al servidor para enviar o modificar datos. Una solicitud consiste en:

* Un verbo o método HTTP que define qué tipo de operación realizará el servidor.
* Un encabezado, con el encabezado de solicitud que pasa información sobre la solicitud.
* Un camino o ruta al servidor, como por ej: <https://app.aluracursos.com/course/http-base-internet>
* Información en el cuerpo de la solicitud, siendo esta información opcional.

**Métodos HTTP**

En la aplicación REST, los métodos más utilizados son:

* El método GET es el método más común, generalmente se usa para solicitar a un servidor que envíe un recurso;
* El método POST está diseñado para enviar datos de entrada al servidor. En la práctica, a menudo se usa para admitir formularios HTML;
* El método PUT edita y actualiza documentos en un servidor;
* El método DELETE que, como su nombre lo indica, elimina un determinado dato o colección del servidor.

En [este enlace](https://developer.mozilla.org/es/docs/Web/HTTP/Methods) encontrarás la lista completa de todos los métodos.

**Códigos de respuesta**

A cada respuesta que devuelve la aplicación REST se le envía un código que define el estado de la solicitud. Por ejemplo:

* 200 (OK) solicitud cumplida con éxito.
* 201 (CREADO) objeto o recurso creado con éxito.
* 204 (SIN CONTENIDO) objeto o recurso eliminado con éxito.
* 400 (MALA SOLICITUD) ocurrió un error en la solicitud (puede haber numerosas causas).
* 404 (NO ENCONTRADO) ruta o colección no encontrada.
* 500 (ERROR INTERNO DEL SERVIDOR), se ha producido algún error del servidor.

Estos son los principales, no obstante en [este enlace](https://developer.mozilla.org/es/docs/Web/HTTP/Status) encontrarás la lista completa del código de cada solicitud.

**Conclusión**

En este artículo presentamos el origen, conceptos y fundamentos principales de REST. [Los cursos de programación de Alura](https://app.aluracursos.com/formacoes) pueden ayudarte a iniciarte, aprender y desarrollarte en tecnología y además, luego puedes profundizar y [utilizar esta arquitectura en la gestión de estados de React](https://app.aluracursos.com/formacion-react) por ejemplo.