

1) ¿Qué es una Comunicación REST?, ¿Para qué se usan? Ejemplifique.

La comunicación REST (Representational State Transfer) es un estilo arquitectónico de comunicación entre sistemas que se basa en el protocolo HTTP (Hypertext Transfer Protocol). Se utiliza para permitir que diferentes sistemas se comuniquen entre sí de manera eficiente, a través de una interfaz web que utiliza un conjunto predefinido de operaciones (como GET, POST, PUT y DELETE) para manipular los recursos.

Los servicios REST se utilizan en una amplia variedad de aplicaciones y servicios web, incluyendo redes sociales, comercio electrónico, aplicaciones móviles, entre otros. Algunos de los usos más comunes incluyen:

- *Intercambio de datos*: REST permite que diferentes sistemas intercambien información y recursos de manera estandarizada y predecible, lo que facilita la integración de sistemas y la automatización de procesos.
- *Construcción de APIs*: Las APIs (Application Programming Interfaces) son interfaces de programación de aplicaciones que permiten que diferentes aplicaciones o servicios se comuniquen entre sí. REST es un estilo de arquitectura muy utilizado para construir APIs que permiten la interacción entre diferentes servicios.
- *Acceso a bases de datos*: REST se utiliza a menudo para permitir el acceso a bases de datos a través de la web. Los sistemas pueden utilizar una API RESTful para realizar consultas a la base de datos y obtener los resultados a través de una respuesta HTTP.

Un ejemplo común de uso de REST es en una aplicación de redes sociales. La aplicación puede utilizar una API RESTful para permitir que los usuarios compartan y vean publicaciones, fotos, videos y otros recursos. Los usuarios pueden interactuar con la aplicación a través de la interfaz web de la API, que utiliza operaciones HTTP como GET y POST para recuperar y actualizar recursos.

Criterios de API REST

Existen diversos criterios para identificar si una API es REST o no. Algunos de ellos son que:

Debe usar una arquitectura cliente-servidor.

Las ejecuciones de la API no deben considerar el estado del cliente, el estado de peticiones anteriores o algún indicador almacenado que haga variar su comportamiento. La comunicación debe ser sin estado (stateless).

Ha de estar orientada a recursos, usando las operaciones estándar de los verbos HTTP.

Hace uso de la URL como identificador único de los recursos.

Debe ser hipermedia: cuando se consulte un recurso, este debe contener links o hipervínculos de acciones o recursos que lo complementen.

Diferencia entre RESTful y RESTless

A menudo escuchamos estos términos al momento de diseñar, construir o interactuar con API. La diferencia es sencilla. Llamamos RESTful a todas aquellas API que cumplen completamente los criterios REST; mientras que llamamos RESTless a aquellas API que no cumplen del todo con los criterios REST.

Por ejemplo, una API que utiliza el verbo POST para todas sus operaciones no es una API RESTful, si no una API RESTless.

¿Qué es la API Specification?

La especificación de una API, o API Spec, es aquella documentación donde se describe el comportamiento de una API, también conocido como el contrato de la API. La finalidad de dicha documentación es guiar al desarrollador que va a integrar el uso de la API en su sistema. Es tal la importancia que ha tomado este rubro que existen diversas herramientas y estándares creados específicamente para describir una API REST como son RAML, Swagger y el estándar OpenAPI.

Los componentes primordiales que se describen en un API Spec son los siguientes:

Verbo HTTP

Son aquellos verbos propios del protocolo HTTP que fueron tomados para definir operaciones muy puntuales y específicas sobre los recursos de la API.

Los más utilizados son:

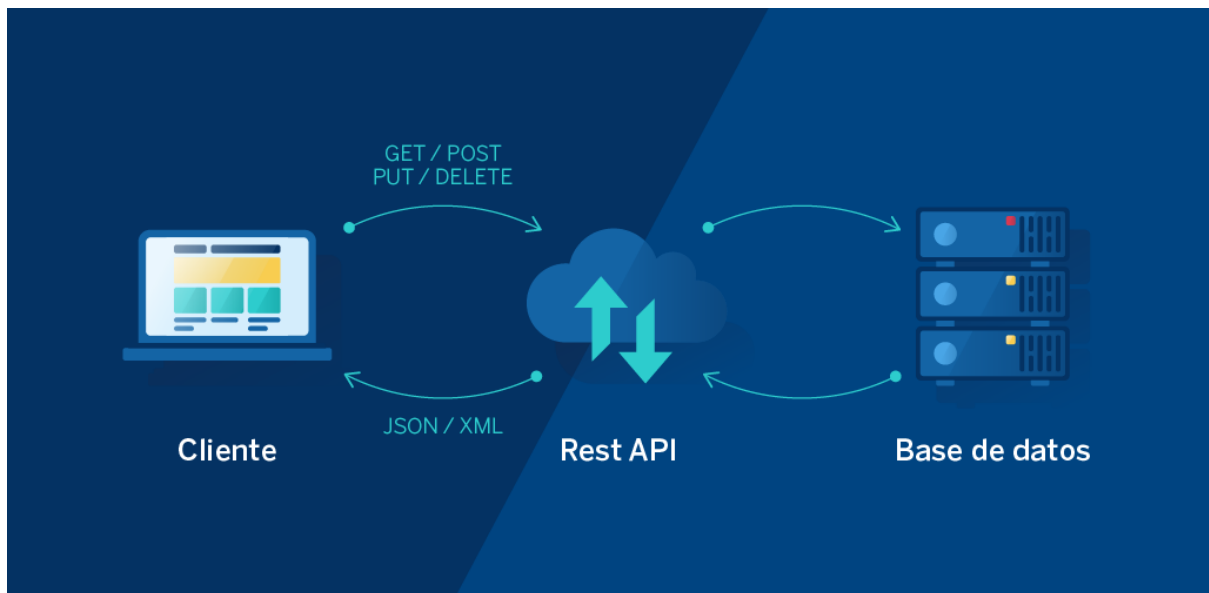
GET: listado de recursos. Detalle de un solo recurso.

POST: creación de un recurso.

PUT: modificación total de un recurso.

PATCH: modificación parcial de un recurso.

DELETE: eliminación de un recurso. En muchas ocasiones es un soft delete, es decir, no se elimina definitivamente un recurso sino que únicamente es marcado como eliminado o desactivado.



URL orientada a recursos: la definición de las URL de los endpoint de la API están orientadas a recursos, es decir, a entidades que tienen coherencia dentro del contexto de la API. Por ejemplo, en una API para un sistema que administra libros sería fácil encontrar entidades como libros, autores, editoriales, colecciones, etc. Estas entidades las veremos reflejadas en URL orientadas a recursos que las representen, por ejemplo:

/api-libros/v0/autores: identifica los recursos autores

/api-libros/v0/autores/{id-autor}: identifica un recurso autor

/api-libros/v0/autores/{id-autor}/libros: identifica los libros de un autor en específico

/api-libros/v0/libros: identifica los recursos libros

/api-libros/v0/editoriales: identifica los recursos editoriales

/api-libros/v0/editoriales/{id-editorial}/libros: identifica los libros de una editorial

HTTP Status

Son aquellos estatus de respuesta propios del protocolo HTTP que fueron tomados para informar sobre el resultado de la operación solicitada. Los más comunes en API REST son:

200 - OK

201 - Created

204 - No Content

400 - Bad Request

401 - Unauthorized

403 - Forbidden

404 - Not Found

500 - Internal Server Error

A continuación, verás una tabla de ejemplo para ilustrar estos conceptos:

A continuación, verás una tabla de ejemplo para ilustrar estos conceptos:

VERBO HTTP	URL DEL RECURSO	ACCIÓN	HTTP STATUS
GET	/libros	Obtener una lista de libros	200 - OK
			204 - Not Content: cuando no hay libros a listar
			200 - OK
GET	/libros/{id-libro}	Obtener detalle de un libro	404 - Not Found: cuando no existe el libro buscado
POST	/libros	Crear un recurso nuevo libro	201 - Created

			200 - OK
PUT	/libros/{id-libro}	Modificar un recurso libro completamente	400 - Bad request: cuando algún parámetro enviado no cumple con las reglas
			200 - OK
PATCH	/libros/{id-libro}	Modificar un recurso libro parcialmente	201 - Created: si el recurso a modificar no existe se crea en automático 400 - Bad request: cuando algún parámetro enviado no cumple con las reglas
DELETE	/libros/{id-libro}	Eliminar un recurso libro	201 - Not Content: es el status standard a regresar en un verbo DELETE

Las API REST han aportado mucho en la forma de comunicación entre sistemas. Al día de hoy, un gran porcentaje de las empresas que usan la tecnología para aportar valor a sus productos tienen API REST como forma estándar de comunicación.

Existe una cantidad enorme de API disponibles en Internet para agregar valor a cualquier desarrollo de software, ya sea para tercerizar algún módulo necesario para el sistema en cuestión (como timbrado de facturas, pagos online, etc.) o simplemente para mejorar la experiencia del usuario, como la geolocalización, el acceso con cuentas de redes sociales y demás.