

# API REST

# Índice

<b>1. ¿Qué es una API?</b>	<b>3</b>
<b>2. Tipos de API</b>	<b>4</b>
<b>3. ¿Qué son las API 's REST?</b>	<b>4</b>
<b>4. ¿Qué beneficios tienen las API REST?</b>	<b>4</b>
Escalabilidad	4
Flexibilidad	5
Independencia	5
<b>5. ¿Qué son los métodos de autenticación de la API REST?</b>	<b>5</b>
Autenticación HTTP	5
Autenticación básica	5
Autenticación del portador	5
Claves de la API	6
OAuth	6
<b>6. Archivos JSON</b>	<b>6</b>
¿Qué es un JSON?	6
¿Para qué sirve el formato JSON?	6
Transferencia de datos en aplicaciones web y servicios web	6
Configuración y almacenamiento de datos en aplicaciones	7
Intercambio de datos entre diferentes lenguajes de programación	7
Características del formato JSON	7
Estructura de datos basada en pares clave / valor	7
Ligereza y legibilidad	7
Soporte multiplataforma	7
Ventajas y beneficios del formato JSON	7
Facilidad de uso y lectura por humanos	7
Eficiencia en la transmisión de datos	8
Compatibilidad con numerosos lenguajes de programación	8
Comparación con otros formatos de datos	8
JSON vs XML	8
JSON vs CSV	8

## 1. ¿Qué es una API?

Una API es un conjunto de reglas y protocolos que permiten a diferentes aplicaciones interactuar entre sí. En otras palabras, es una interfaz que permite que dos o más sistemas se comuniquen y compartan datos y servicios.

Las APIs son utilizadas en la mayoría de los sistemas de software modernos, desde aplicaciones móviles hasta servicios web, y conocer su terminología es fundamental para desarrollar y utilizar aplicaciones de manera eficiente y efectiva. Una comprensión adecuada de la terminología de las API permite a los desarrolladores integrar diferentes sistemas y servicios, lo que resulta en una mejor experiencia para el usuario final.

Las API son fundamentales para la comunicación y la integración de aplicaciones y sistemas, y son utilizadas por muchas empresas y organizaciones para conectar sus sistemas y aplicaciones a otros sistemas externos.

Las API funcionan permitiendo que las aplicaciones se comuniquen entre sí utilizando una serie de mensajes y llamadas de función. Un sistema expone una API a través de una interfaz, que puede ser utilizada por otras aplicaciones para interactuar con el sistema y realizar tareas específicas.

## 2. Tipos de API

- SOAP: Es un protocolo de comunicación que utiliza XML para intercambiar información entre diferentes sistemas.
- RPC: Permiten que una aplicación invoque una función o método en un sistema remoto a través de una red.
- WebSocket: Es un protocolo de comunicación bidireccional que permite que los datos se intercambien entre una aplicación web y un servidor en tiempo real.
- REST: Es una forma de permitir que diferentes aplicaciones se comuniquen entre sí a través de internet utilizando el protocolo HTTP.

## 3. ¿Qué son las API 's REST?

La API RESTful es una interfaz que dos sistemas de computación utilizan para intercambiar información de manera segura a través de Internet. La mayoría de las aplicaciones para empresas deben comunicarse con otras aplicaciones internas o de terceros para llevar a cabo varias tareas. Por ejemplo, para generar nóminas mensuales, su sistema interno de cuentas debe compartir datos con el sistema bancario de su cliente para automatizar la facturación y comunicarse con una aplicación interna de planillas de horarios. Las API RESTful admiten este intercambio de información porque siguen estándares de comunicación de software seguros, confiables y eficientes.

## 4. ¿Qué beneficios tienen las API REST?

### Escalabilidad

Los sistemas que implementan API REST pueden escalar de forma eficiente porque REST optimiza las interacciones entre el cliente y el servidor. La tecnología sin estado elimina la carga del servidor porque este no debe retener la información de solicitudes pasadas del cliente. El almacenamiento en caché bien administrado elimina de forma parcial o total algunas interacciones entre el cliente y el servidor.

Todas estas características admiten la escalabilidad, sin provocar cuellos de botella en la comunicación que reduzcan el rendimiento

## Flexibilidad

Los servicios web RESTful admiten una separación total entre el cliente y el servidor. Simplifican y desacoplan varios componentes del servidor, de manera que cada parte pueda evolucionar de manera independiente. Los cambios de la plataforma o la tecnología en la aplicación del servidor no afectan la aplicación del cliente. La capacidad de ordenar en capas las funciones de la aplicación aumenta la flexibilidad aún más. Por ejemplo, los desarrolladores pueden efectuar cambios en la capa de la base de datos sin tener que volver a escribir la lógica de la aplicación.

## Independencia

Las API REST son independientes de la tecnología que se utiliza. Puede escribir aplicaciones del lado del cliente y del servidor en diversos lenguajes de programación, sin afectar el diseño de la API. También puede cambiar la tecnología subyacente en cualquiera de los lados sin que se vea afectada la comunicación.

## 5. ¿Qué son los métodos de autenticación de la API REST?

Un servicio web RESTful debe autenticar las solicitudes antes de poder enviar una respuesta. La autenticación es el proceso de verificar una identidad. Por ejemplo, puede demostrar su identidad mostrando una tarjeta de identificación o una licencia de conducir. De forma similar, los clientes de los servicios RESTful deben demostrar su identidad al servidor para establecer confianza.

### Autenticación HTTP

HTTP define algunos esquemas de autenticación que se pueden utilizar directamente cuando se implementa la API REST. A continuación, se indican dos de estos esquemas:

#### Autenticación básica

En la autenticación básica, el cliente envía el nombre y la contraseña del usuario en el encabezado de la solicitud. Los codifica en base64, que es una técnica de codificación que convierte el par en un conjunto de 64 caracteres para su transmisión segura.

#### Autenticación del portador

El término autenticación del portador se refiere al proceso de brindar el control de acceso al portador del token. El token del portador suele ser una cadena de

caracteres cifrada que genera el servidor como respuesta a una solicitud de inicio de sesión. El cliente envía el token en los encabezados de la solicitud para acceder a los recursos.

## Claves de la API

Las claves de la API son otra opción para la autenticación de la API REST. En este enfoque, el servidor asigna un valor único generado a un cliente por primera vez. Cada vez que el cliente intenta acceder a los recursos, utiliza la clave de API única para su verificación. Las claves de API son menos seguras debido a que el cliente debe transmitir la clave, lo que la vuelve vulnerable al robo de red.

## OAuth

OAuth combina contraseñas y tokens para el acceso de inicio de sesión de alta seguridad a cualquier sistema. El servidor primero solicita una contraseña y luego solicita un token adicional para completar el proceso de autorización. Puede verificar el token en cualquier momento y, también, a lo largo del tiempo, con un alcance y duración específicos.

# 6. Archivos JSON

## ¿Qué es un JSON?

JSON es un formato para la especificación de datos usados de manera habitual en aplicaciones y sitios web. Su principal utilidad es el intercambio de datos entre sistemas informáticos. Es un archivo ligero ya que los datos de este no suelen ocupar mucho

## ¿Para qué sirve el formato JSON?

### Transferencia de datos en aplicaciones web y servicios web

JSON es comúnmente utilizado para transmitir datos en aplicaciones web entre el cliente y el servidor. Un uso común de JSON es obtener datos de un servidor web a través de una solicitud HTTP. Este intercambio de datos puede ser entre dos servidores completos o entre un servidor y un cliente.

Los servicios web, generalmente bajo arquitecturas REST, son uno de los marcos habituales de uso de JSON, desplazando a otros lenguajes más formales como XML.

## Configuración y almacenamiento de datos en aplicaciones

Los formatos JSON también se utilizan para almacenar configuraciones de las aplicaciones, ya sean introducidas en el código o como soporte al almacenamiento de las configuraciones introducidas por el usuario.

Además, muchos sistemas de bases de datos están adoptando este formato como notación para el almacenamiento de los datos. Comenzaron los Sistemas Gestores de Base de Datos conocidos como NoSQL, como MongoDB, pero actualmente la mayoría de los SGBD soporta también JSON como posibles tipos de datos para sus columnas.

## Intercambio de datos entre diferentes lenguajes de programación

Debido a su naturaleza independiente del lenguaje, JSON se puede usar para intercambiar datos entre diferentes lenguajes de programación. Por ejemplo, un servidor puede estar escrito en PHP y el cliente web que consume los datos del backend PHP puede estar escrito en Javascript, Sin embargo, ambos pueden enviar y recibir datos en formato JSON.

## Características del formato JSON

### Estructura de datos basada en pares clave / valor

La estructura de los datos JSON se basa en dos elementos: una colección de pares clave / valor y una lista ordenada de valores. Esto es similar a los tipos de datos utilizados en muchos lenguajes de programación, como los objetos y los arrays.

### Ligereza y legibilidad

JSON es un formato ligero, lo que significa que no consume muchos recursos ni en su construcción ni a la hora de transferirlo por Internet. Esto hace que JSON sea ideal para dispositivos con recursos limitados, como smartphones. Además, su sintaxis clara y simple facilita la lectura y el análisis por parte de los humanos.

### Soporte multiplataforma

JSON es independiente del lenguaje de programación, lo que significa que se puede usar en cualquier lenguaje que pueda analizar, generar y manipular datos en este formato. Esto hace que JSON sea muy flexible y adaptable a diferentes situaciones.

## Ventajas y beneficios del formato JSON

### Facilidad de uso y lectura por humanos

JSON es fácil de leer y escribir. Su estructura simple y clara facilita la comprensión de los datos a simple vista.

## Eficiencia en la transmisión de datos

Debido a su naturaleza ligera, JSON es eficiente en la transmisión de datos, lo que lo hace ideal para la comunicación entre el cliente y el servidor en aplicaciones web pero en general también en cualquier sistema distribuido.

## Compatibilidad con numerosos lenguajes de programación

JSON es compatible con una amplia gama de lenguajes de programación, lo que lo convierte en una opción versátil para el intercambio de datos. Realmente todos los lenguajes actuales soportan JSON, por lo que sería difícil encontrar uno que no admita este formato.

## Comparación con otros formatos de datos

### JSON vs XML

Aunque XML fue inicialmente utilizado para los servicios web y el intercambio de datos en general, con el tiempo JSON ha venido desplazando a XML debido a su simplicidad y eficiencia.

JSON tiene una sintaxis más simple y es más fácil de leer y escribir que XML. Además, JSON es más ligero, no solo para su transmisión sino también para el procesamiento de los datos. Por ello resulta más eficiente.

### JSON vs CSV

CSV es otro formato comúnmente utilizado para el intercambio de datos, especialmente útil para exportar información. Aunque CSV es simple y ligero, no es tan fácil de leer y asimilar qué función realiza cada dato. Además CSV no resulta nada flexible, porque las columnas siempre tienen una serie limitada y definida de valores. En cambio JSON soporta estructuras de datos complejas y variables entre sí.