

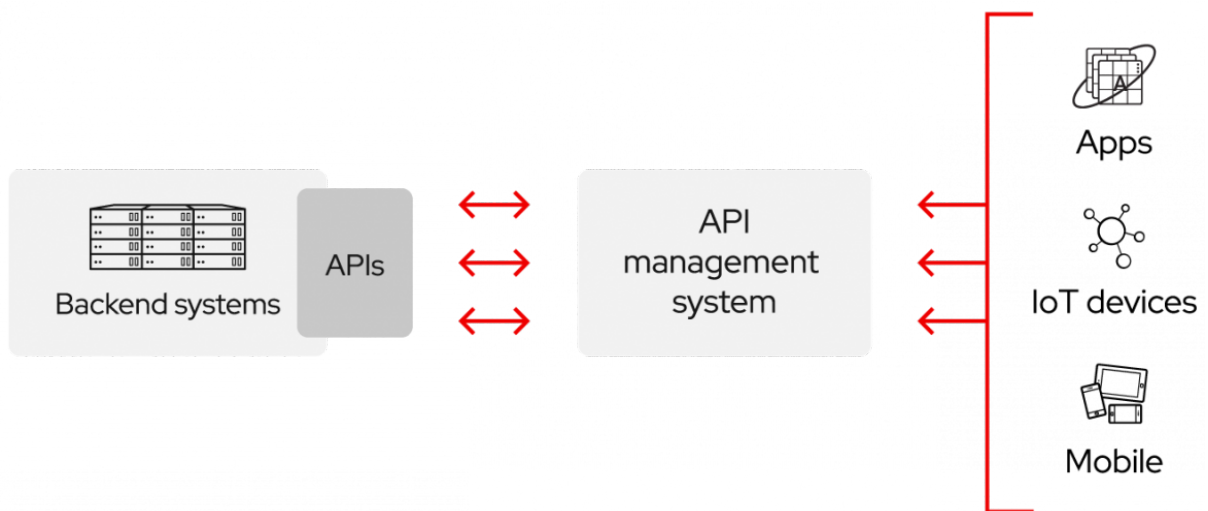
# ¿Qué es una API?

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

Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados, simplificando el desarrollo de las aplicaciones y permitiendo ahorrar tiempo y dinero, también ofrecen oportunidades de innovación.

Las API facilitan la colaboración entre los equipos comerciales y de TI al simplificar la integración de nuevos elementos en aplicaciones existentes. En entornos digitales cambiantes, es fundamental adaptarse rápidamente para mantenerse competitivo. Las API permiten el desarrollo ágil de servicios innovadores, especialmente al conectar microservicios a través de una arquitectura de aplicaciones nativas de la nube.

Las API públicas aportan un valor comercial único porque simplifican y amplían sus conexiones con los partners y, además, pueden rentabilizar sus datos (un ejemplo conocido es la API de Google Maps).



Las API **privadas** solo se pueden usar internamente, así que las empresas tienen un mayor control sobre ellas.

Las API **de partners** se comparten con partners empresariales específicos, lo cual puede ofrecer flujos de ingresos adicionales, sin comprometer la calidad.

Todos tienen acceso a las API **públicas**, así que otras empresas pueden desarrollar API que interactúen con las de usted y así convertirse en una fuente de innovaciones.

Las API brindan oportunidades significativas al permitir el acceso a socios y al público en general. Esto resulta en la generación de nuevas fuentes de ingresos, la expansión de canales y la mayor visibilidad de la marca teniendo así efectos positivos. Facilitando la innovación y colaboración externa.

Utilizar una API pública estimula la creación de un ecosistema de aplicaciones alrededor de su tecnología, lo que puede impulsar el crecimiento de su negocio.

**Red Hat Integration** ayuda a las empresas a optimizar el rendimiento de las aplicaciones y los resultados empresariales

Las API tienen una larga historia, surgiendo en los primeros días de la informática antes de la era de las computadoras personales. Inicialmente, se usaban como bibliotecas para sistemas operativos y operaban principalmente de forma local. Con el tiempo, se expandieron para permitir la interacción en redes de comunicación, principalmente a través de Internet. Las API remotas son aquellas diseñadas para interactuar con recursos ubicados fuera de la computadora que envía la solicitud. La mayoría de las API web utilizan HTTP para las solicitudes y proporcionan una estructura definida para las respuestas, comúnmente en formatos como XML o JSON para facilitar el manejo de datos por otras aplicaciones.

A medida que se ha difundido el uso de las API, se desarrolló una especificación de protocolo para permitir la estandarización del intercambio de información; se llama Protocolo de Acceso a Objetos Simples, más conocido como **SOAP**.

Las API diseñadas con SOAP usan XML para el formato de sus mensajes y reciben solicitudes a través de HTTP o SMTP. Con SOAP, es más fácil que las aplicaciones que funcionan en entornos distintos o están escritas en diferentes lenguajes compartan información.

Otra especificación es la Transferencia de Estado Representacional o **REST**. Las API web que funcionan con las limitaciones de arquitectura REST se llaman API de RESTful.

REST y SOAP son dos enfoques diferentes para construir y gestionar API. SOAP es un protocolo, mientras que REST es una arquitectura. Las API se consideran RESTful si cumplen con seis limitaciones principales:

1. Cliente-Servidor: Utiliza una arquitectura con roles separados para el cliente y el servidor, gestionando las solicitudes a través de HTTP.
2. Sin Estado: La información del cliente no se guarda en el servidor entre las solicitudes. El cliente retiene el estado de la sesión.
3. Capacidad de Caché: Permite almacenar en caché, reduciendo algunas interacciones cliente-servidor.

4. Sistema en Capas: Puede haber capas adicionales entre cliente y servidor, proporcionando funciones adicionales como equilibrio de carga, cachés compartidos o seguridad.
5. Código Bajo Demanda: Los servidores pueden enviar código ejecutable al cliente para ampliar sus capacidades.
6. Interfaz Uniforme: Define cuatro aspectos clave para el diseño de API RESTful:
  - Identificación de recursos en las solicitudes.
  - Administración de recursos a través de representaciones.
  - Mensajes autodescriptivos.
  - Hipermedios como motor del estado de la aplicación.

Aunque estas limitaciones pueden parecer complejas, son menos restrictivas que un protocolo predefinido como SOAP. Por lo tanto, las API RESTful están ganando popularidad en comparación con las basadas en SOAP.

En los últimos años, OpenAPI se ha convertido en un estándar para definir APIs de REST, permitiendo a los desarrolladores diseñarlas de manera clara y comprensible sin depender de un lenguaje específico. Otro estándar emergente es GraphQL, que se presenta como una alternativa a REST. GraphQL se enfoca en proporcionar a los clientes solo los datos que solicitan, permitiendo a los desarrolladores realizar consultas eficientes que obtienen información de múltiples fuentes en una sola llamada a la API.

En el artículo se describen dos enfoques arquitectónicos comunes para APIs remotas: **SOA y microservicios**.

SOA fue una mejora de aplicaciones monolíticas, permitiendo el uso de varias aplicaciones especializadas. Sin embargo, puede generar complicaciones si las interacciones no se entienden bien.

Por otro lado, los microservicios dividen arquitecturas en partes más pequeñas, utilizando APIs de RESTful para una comunicación eficiente y permitiendo una distribución rápida de funciones y actualizaciones. Cada servicio es independiente, lo que optimiza los recursos y permite adaptaciones dinámicas.

Se mencionan webhooks como funciones de devolución de llamadas que facilitan la comunicación entre APIs de manera ligera. A diferencia de las API convencionales, en los webhooks es el servidor quien inicia la comunicación. Se destaca que, aunque complementarios, los webhooks no son APIs por sí mismos y las aplicaciones deben tener una API para utilizar un webhook.