

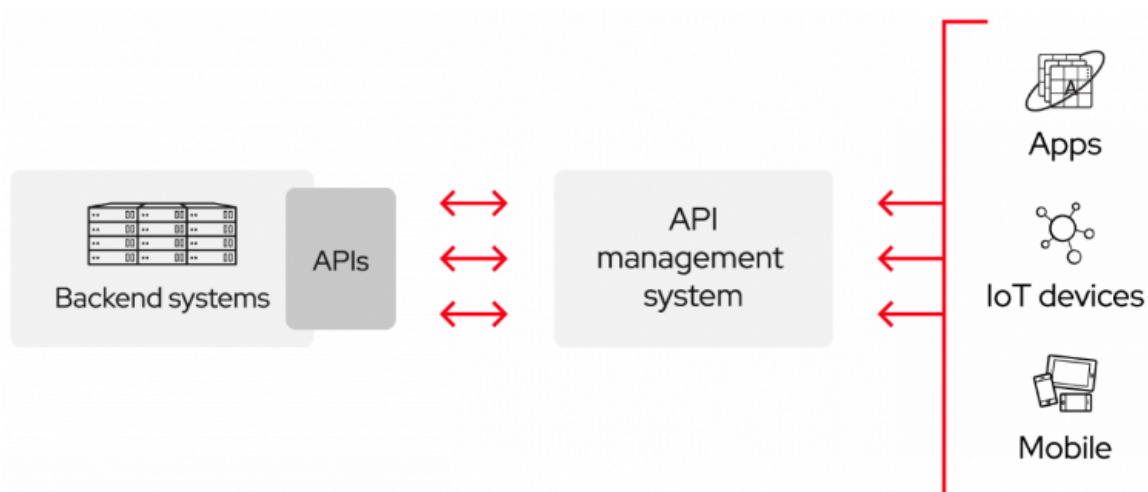
¿Qué es una API?

Una API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usan para diseñar e integrar el software de las aplicaciones.

¿Cómo funciona una API?

Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones; y ofrecen oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales).

Las API son un medio simplificado para conectar su propia infraestructura a través del desarrollo de aplicaciones nativas de la nube, pero también le permiten compartir sus datos con clientes y otros usuarios externos. 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).



En resumen, las API le permiten habilitar el acceso a sus recursos y, al mismo tiempo, mantener la seguridad y el control. Usted decide cómo habilitar el acceso y a quiénes se lo otorga. La seguridad de las API depende de su buena gestión, lo cual incluye el uso de una puerta de enlace de API. Para conectarse a las API y crear aplicaciones que utilicen los datos o las funciones que ofrecen, se puede utilizar una plataforma de integración distribuida que conecte todos los elementos, como los sistemas heredados y el Internet de las cosas (IoT).

Políticas de lanzamiento de las API

Privado

Las API solo se pueden usar internamente, así que las empresas tienen un mayor control sobre ellas. Esto les da a las empresas un mayor control sobre sus API.

De partners

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

Público

Todos tienen acceso a las API, así que otras empresas pueden desarrollar API que interactúen con las de usted y así convertirse en una fuente de innovaciones. Esto permite que terceros desarrollen aplicaciones que interactúan con su API, y puede ser un recurso para innovar.

Innovaciones con las API

El acceso de los partners o el público a sus API permite:

- Crear nuevos canales de ingresos o ampliar los existentes.
- Expandir el alcance de su marca.
- Facilitar la innovación abierta o lograr mayor eficiencia, gracias al desarrollo y la colaboración externa.

Las API remotas

Las API remotas están diseñadas para interactuar en una red de comunicaciones. La palabra *remota* indica que los recursos que administra la API se encuentran fuera de la computadora que envía la solicitud. Dado que la red de comunicaciones más usada es Internet, la mayoría de las API están diseñadas de acuerdo con los estándares web. No todas las API remotas son API web, pero se puede suponer que las API web son remotas.

Por lo general, las API web usan HTTP para los mensajes de solicitud y proporcionan una definición de la estructura de los mensajes de respuesta. Estos mensajes suelen tomar la forma de un archivo XML o JSON, que son los formatos preferidos porque presentan los datos en una manera que otras aplicaciones pueden manejar con facilidad.

Diferencias entre SOAP y REST

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 (REST). Las API web que funcionan con las limitaciones de arquitectura REST se llaman API de RESTful. La diferencia entre REST y SOAP es básica: SOAP es un protocolo, mientras que REST es un estilo de arquitectura. Esto significa que no hay ningún estándar oficial para las API web de RESTful. Tal como se define en la tesis de Roy Fielding, "Architectural Styles and the Design of Network-based Software Architectures", las API son RESTful siempre que cumplan con las 6 limitaciones principales de un sistema RESTful:

- **Arquitectura cliente-servidor:** la arquitectura REST está compuesta por clientes, servidores y recursos; y administra las solicitudes con HTTP.
- **Sistema sin estado:** el contenido de los clientes no se almacena en el servidor entre las solicitudes. En su lugar, la información sobre el estado de la sesión está en posesión del cliente.
- **Capacidad de almacenamiento en caché:** el almacenamiento en caché elimina la necesidad de algunas interacciones cliente-servidor.
- **Sistema en capas:** las interacciones cliente-servidor pueden estar mediadas por capas adicionales. Estas capas pueden ofrecer funcionalidades adicionales, como equilibrio de carga, cachés compartidos o seguridad.
- **Disponibilidad del código según se solicite (opcional):** los servidores pueden ampliar las funciones de un cliente transfiriendo código ejecutable.
- **Interfaz uniforme:** esta limitación es fundamental para el diseño de las API de RESTful e incluye cuatro aspectos:
 - **Identificación de los recursos en las solicitudes:** se identifican los recursos en las solicitudes y se separan de las representaciones que se envían al cliente.
 - **Administración de los recursos mediante representaciones:** los clientes reciben archivos que representan los recursos. Estas representaciones deben tener la información suficiente como para poder ser modificadas o eliminadas.
 - **Mensajes autodescriptivos:** cada mensaje que se envía al cliente contiene una descripción clara sobre la manera en que debe procesar la información.

- **Hipermedios como el motor del estado de la aplicación:** es necesario que después de que el cliente REST acceda a un recurso, pueda descubrir todas las otras acciones que están disponibles actualmente mediante hipervínculos.

Estas limitaciones pueden parecer demasiadas, pero son mucho más sencillas que un protocolo definido previamente. Por eso, las API de RESTful son cada vez más frecuentes que las de SOAP.

En los últimos años, la especificación de OpenAPI se ha convertido en un estándar común para definir las API de REST. OpenAPI establece una manera para que los desarrolladores diseñen las API de REST sin depender de un lenguaje, de forma tal que los usuarios puedan comprenderlas con el mínimo esfuerzo.

Otro estándar cada vez más popular de las API es GraphQL, un lenguaje de consulta y tiempo de ejecución del servidor que se presenta como una alternativa a REST. Para GraphQL, lo más importante es que los clientes reciban exactamente los datos que solicitan y nada más. Como alternativa a REST, GraphQL permite que los desarrolladores creen consultas para extraer datos de varias fuentes en una sola llamada a la API.

La arquitectura orientada a los servicios y la arquitectura de microservicios

Los dos enfoques de arquitectura que más se utilizan para las API remotas son la arquitectura orientada a los servicios (SOA) y la arquitectura de microservicios. La SOA es el más antiguo de los dos, y comenzó como una mejora de las aplicaciones monolíticas. En lugar de usar una sola aplicación que haga todo, se pueden usar varias aplicaciones que proporcionan diferentes funciones y que no tengan conexión directa, todo gracias a un patrón de integración, como un bus de servicios empresariales (ESB).

Aunque en muchos aspectos la SOA es más sencilla que una arquitectura monolítica, conlleva un riesgo de cambios en cascada en todo el entorno si las interacciones de los componentes no se comprenden claramente. Esta complejidad adicional vuelve a presentar algunos de los problemas que la SOA pretendía solucionar.

Las arquitecturas de microservicios se parecen a los patrones SOA en que los servicios son especializados y no tienen conexión directa. Pero además, descomponen las arquitecturas tradicionales en partes más pequeñas. Los servicios de la arquitectura de microservicios usan un marco de mensajería común, como las API de RESTful. Utilizan API de RESTful para comunicarse entre sí, sin necesidad de operaciones complejas de conversión de datos ni capas de integración adicionales. Usar las API de RESTful permite e incluso fomenta una distribución

más rápida de nuevas funciones y actualizaciones. Cada servicio es independiente. Un servicio se puede reemplazar, mejorar o abandonar, sin afectar los demás servicios de la arquitectura. Esta arquitectura liviana optimiza los recursos distribuidos o en la nube y admite la adaptación dinámica de los servicios individuales.