### **¿Qué es una API?**

#### **Definición de API**

Una **API (Interfaz de Programación de Aplicaciones)** es un conjunto de definiciones y protocolos que permite que diferentes aplicaciones se comuniquen entre sí. Facilitan la integración de software y simplifican el desarrollo, permitiendo flexibilidad, eficiencia, y oportunidades de innovación.

#### **¿Cómo funcionan las API?**

* Las API actúan como contratos entre partes, definiendo cómo se envían y reciben datos.
* Permiten a los desarrolladores integrar aplicaciones nuevas con arquitecturas existentes.
* Apoyan la creación de aplicaciones mediante la conexión de microservicios en arquitecturas nativas de la nube.
* Proporcionan acceso a recursos internos o externos (por ejemplo, la API de Google Maps).

#### **Ejemplo práctico: Distribuidora de libros**

Una distribuidora podría ofrecer una API para consultar inventarios en lugar de desarrollar una aplicación específica. Esto permite:

1. A los clientes acceder a los datos según sus necesidades.
2. Cambios internos sin afectar a los usuarios.
3. Desarrollo de aplicaciones por terceros, ampliando las oportunidades comerciales.

#### **Tipos de políticas de API**

1. **Privadas**: Solo uso interno, mayor control.
2. **De partners**: Compartidas con socios específicos, balance entre calidad y nuevas oportunidades.
3. **Públicas**: Abiertas para terceros, fomentando innovación y colaboración externa.

#### **Historia y evolución**

* Las API comenzaron como bibliotecas locales para sistemas operativos.
* Evolucionaron hacia **API remotas**, diseñadas para comunicarse a través de redes (frecuentemente usando HTTP).
* Los formatos comunes de respuesta son **JSON** o **XML**.

#### **SOAP vs REST**

1. **SOAP**:
   * Protocolo formal.
   * Utiliza exclusivamente XML.
   * Compatible con diversas plataformas y lenguajes.
2. **REST**:
   * Estilo arquitectónico más flexible.
   * Usa formatos como JSON o XML.
   * Cumple con restricciones como cliente-servidor, sin estado y capacidad de caché.
   * Más común que SOAP por su simplicidad.

#### **Nuevas alternativas**

* **GraphQL**: Proporciona exactamente los datos solicitados por el cliente, optimizando consultas.
* **OpenAPI**: Estandarización de la documentación y diseño de API RESTful.

#### **API y arquitectura**

* **SOA (Arquitectura Orientada a Servicios)**: Modulariza aplicaciones monolíticas mediante servicios integrados.
* **Microservicios**: Independientes, más pequeños, usan API RESTful, y favorecen una implementación más rápida.

#### **Diferencia entre API y webhooks**

* Un **webhook** es una notificación basada en eventos que permite a un servidor enviar datos a un cliente sin necesidad de solicitudes constantes. Son más ligeros pero requieren una API para funcionar.

#### **Beneficios de las API**

* Facilitan la colaboración y la innovación.
* Abren nuevos canales de ingresos.
* Promueven la eficiencia operativa.
* Incrementan el alcance de la marca.

Este enfoque conecta datos, sistemas y servicios de manera eficiente, fomentando la transformación digital y adaptándose rápidamente a las demandas del mercado.