



FUNDACIÓN
KINAL

Desarrollo de API's

Josué Javier yax Ixcoy

2023589

IN6CV

Índice

¿Qué es una API?	4
¿Qué es una API Rest?	5
¿Qué es una API Restful?	5
Buenas practicas de desarrollo de API's.....	6
Conclusión	7
Recomendaciones	8
Referencias	9

Hoy en día, cuando hablamos de desarrollo de software moderno, es casi imposible no mencionar las **API**, especialmente las que siguen el estilo **REST** o son conocidas como **RESTful**. Estas herramientas son fundamentales para que las aplicaciones puedan "hablar" entre sí, compartiendo datos y funciones de manera ordenada, sin necesidad de que todo esté en un solo lugar. Por ejemplo, cuando usamos una app del clima que nos muestra datos actualizados, lo más probable es que esté conectándose a una API que le da esa información. Pero no basta con saber qué es una API: también es importante entender qué la hace **REST** o **RESTful**, y por qué seguir **buenas prácticas de desarrollo** puede hacer que nuestra API sea más fácil de mantener, más rápida y más útil para otros desarrolladores. Aprender estos conceptos no solo ayuda a escribir mejor código, sino que también nos prepara para trabajar en proyectos reales donde la conexión entre sistemas es clave. En esta investigación vamos a conocer qué es una API, qué significa que sea REST o RESTful, y cómo aplicar buenas prácticas para que nuestras APIs funcionen bien y sean un ejemplo de orden y eficiencia.

¿Qué es una API?

Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos, facilitando el intercambio de datos y funcionalidades de manera estandarizada y eficiente. Estas interfaces actúan como intermediarios, abstrayendo la complejidad interna de cada componente y exponiendo solo las operaciones y la información necesarias para su interacción. (Goodwin, 2024). Las API's tienen características principales como las siguientes:

- **Interfaz:** Una API actúa como una interfaz entre diferentes componentes de software, sistemas o servicios. Define los métodos y los formatos de datos que se pueden utilizar para comunicarse entre ellos.
- **Abstracción:** Las APIs abstraen la complejidad de la implementación subyacente. Los desarrolladores que utilizan una API no necesitan conocer los detalles internos de cómo funciona un sistema; solo necesitan saber cómo interactuar con la interfaz definida.
- **Gestión de errores:** Es fundamental gestionar los posibles errores que pueden surgir al comunicarnos con una API. Esto puede incluir errores de red, errores de autenticación, límites de tasa alcanzados o respuestas inesperadas del servidor. Debemos implementar mecanismos para manejar estos errores de forma eficiente y proporcionar retroalimentación adecuada al usuario si es necesario.
- **Reusabilidad:** Las API's permiten que la funcionalidad de un sistema sea reutilizada por diferentes aplicaciones o servicios sin necesidad de duplicar el código.
- **Evolución:** Las APIs pueden evolucionar con el tiempo, añadiendo nuevas funcionalidades sin romper la compatibilidad con los clientes existentes. (González, 2024)

¿Qué es una API Rest?

Una API REST (Interfaz de Programación de Aplicaciones - Transferencia de Estado Representacional) es una interfaz que permite a las aplicaciones interactuar entre sí utilizando el protocolo HTTP, siguiendo los principios de diseño de la arquitectura REST. Se basa en la idea de que la comunicación entre un cliente y un servidor se realiza a través de recursos identificados por URLs, y que estos recursos pueden ser manipulados usando métodos HTTP estándar como GET, POST, PUT y DELETE. (RedHat, 2023). Las API Rest tienen características principales como las siguientes:

- **Arquitectura Cliente-Servidor:** Existe una clara separación de responsabilidades. El cliente inicia las peticiones y el servidor las procesa y devuelve las respuestas. Esto permite que el cliente y el servidor evolucionen de forma independiente.
- **Sin Estado (Stateless):** Cada petición del cliente al servidor debe contener toda la información necesaria para que el servidor la entienda. El servidor no guarda ninguna información sobre la sesión del cliente entre peticiones. Si se necesita información sobre la sesión, el cliente debe enviarla en cada solicitud. Esto simplifica el diseño del servidor y mejora la escalabilidad. (AWS, 2022)

¿Qué es una API Restful?

Una API RESTful es una interfaz de programación de aplicaciones (API) que se adhiere a los principios del estilo arquitectónico REST (Representational State Transfer). Utiliza los protocolos estándar de la web, principalmente HTTP, para permitir la comunicación entre diferentes sistemas de software de una manera simple, escalable y fácil de entender. (AWS, 2022). Las API Restful tienen características principales como las siguientes:

- **Arquitectura Cliente-Servidor:** Existe una separación clara entre el cliente que realiza las peticiones y el servidor que gestiona los recursos y las responde. Esto permite que cada uno evolucione de forma independiente.

- **Interfaz Uniforme:** La interacción entre el cliente y el servidor se realiza a través de un conjunto consistente de interfaces. (Bigelow, 2022)

Buenas practicas de desarrollo de API's

Las buenas prácticas de desarrollo de APIs son un conjunto de instrucciones y recomendaciones que buscan crear APIs robustas, fáciles de usar, seguras, mantenibles y con un buen rendimiento. (jaffery, 2023). Algunas prácticas más importantes son:

- **Diseño Centrado en los Recursos:** Define claramente los recursos que tu API expondrá (sustantivos) y utiliza los métodos HTTP (verbos) de forma semántica para realizar operaciones sobre ellos (GET para obtener, POST para crear, PUT para actualizar, DELETE para eliminar).
- **Utilizar Nombres de Recursos Claros y Consistentes:** Emplea nombres en plural y utiliza un lenguaje consistente, por ejemplo: users, products, orders.
- **Filtrado, Ordenamiento y Búsqueda:** Permite a los clientes refinar sus consultas mediante parámetros para filtrar, ordenar y buscar datos dentro de las colecciones. (jaffery, 2023)

Seguridad para buenas prácticas de desarrollo de API's

- **Autenticación Robusta:** Implementación de mecanismos de autenticación seguros para verificar la identidad de los clientes, por ejemplo: OAuth 2.0, JWT.
- **Autorización Adecuada:** Asegurarse que los clientes autenticados solo puedan acceder a los recursos y realizar las acciones para las que tienen permisos.
- **Validación de Entradas:** Validar todas las entradas del cliente en el servidor para prevenir ataques de inyección y errores inesperados. (jaffery, 2023)

Conclusión

En conclusión, en esta investigación ha permitido comprender la distinción fundamental entre el concepto de API, el estilo arquitectónico REST y la implementación específica de API's RESTful. También se ha destacado cómo la arquitectura REST, la interfaz uniforme y uso de los métodos HTTP, se ha convertido en un estándar para la construcción de API's en el desarrollo de software moderno. Asimismo, se ha evidenciado la importancia crítica de adherirse a las buenas prácticas de desarrollo, que no solo mejoran la calidad y la seguridad de las API's, sino que también facilitan su integración y mantenimiento. En última instancia, el dominio de estos conceptos y la aplicación consciente de las buenas prácticas son habilidades esenciales para cualquier profesional involucrado en la creación de software en el panorama tecnológico de la actualidad.

Recomendaciones

A partir de esta investigación, se recomienda a los futuros desarrolladores dedicar tiempo a la práctica y experimentación con la creación de API's RESTful, utilizando diferentes lenguajes y frameworks. Es fundamental priorizar la seguridad en todas las etapas del desarrollo y familiarizarse con las herramientas de documentación estándar. Para las organizaciones, se sugiere establecer lineamientos claros para el diseño y desarrollo de API's, promover la colaboración y la revisión entre los miembros del equipo, e invertir en la monitorización continua de las API's implementadas para garantizar su correcto funcionamiento y seguridad.

Referencias

AWS. (31 de 7 de 2022). *aws*. Obtenido de <https://aws.amazon.com/es/what-is/restful-api/#:~:text=Las%20API%20REST%20son%20independientes%20de%20la,sin%20que%20se%20vea%20afectada%20la%20comunicaci%C3%B3n>.

Bigelow, S. J. (19 de 5 de 2022). *TechTarget*. Obtenido de https://www.techtarget-com.translate.goog/searchapparchitecture/definition/RESTful-API?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=sge

González, D. (23 de 6 de 2024). *davizgonzalez*. Obtenido de <https://davizgonzalez.com/blog/api-que-es/#:~:text=Caracter%C3%ADsticas%20de%20las%20API&text=Las%20API%20est%C3%A1n%20dise%C3%B1adas%20para%20facilitar%20la,o%20el%20lenguaje%20de%20programaci%C3%B3n%20que%20usen>.

Goodwin, M. (9 de 4 de 2024). *IBM*. Obtenido de <https://www.ibm.com/mx-es/think/topics/api#:~:text=Una%20API%2C%20o%20interfaz%20de,intercambiar%20datos%2C%20caracter%C3%ADsticas%20y%20funcionalidades>.

jaffery, A. (23 de 11 de 2023). *Astera*. Obtenido de <https://www.astera.com/es/type/blog/api-development/>

RedHat. (31 de 7 de 2023). *redhat*. Obtenido de <https://www.redhat.com/es/topics/api/what-is-a-rest-api>