

Centro Técnico Laboral Kinal

Fecha: Abril 2025

Pofesor: Jose Braulio Echeverria Montufar

TECNOLOGIA III

Nombre: Harol Anibal Rodriguez Con.

Grado: Sexto Perito en Informatica.

Sección: IN6BV.

Carnet: 2023278

Índice

Contenido

Introducción:.....	3
API: Concepto y Funcionalidad:	3
REST: Definición y Principios.....	3
RESTful: Qué significa y su aplicación	4
Buenas prácticas en el desarrollo de APIs	4
Referencias	5

Introducción:

En el desarrollo de software moderno, la interacción entre aplicaciones, servicios y sistemas es fundamental. Para lograr esta comunicación, las Interfaces de Programación de Aplicaciones (APIs) desempeñan un papel crucial.

Especialmente, los enfoques basados en REST y RESTful han ganado gran popularidad debido a su eficiencia y estandarización. Esta investigación tiene como objetivo comprender los conceptos fundamentales de las APIs, diferenciar REST de RESTful, y resaltar las mejores prácticas para el desarrollo de APIs robustas y escalables.

API: Concepto y Funcionalidad:

Una API (Application Programming Interface) es un conjunto de definiciones y protocolos que permiten la comunicación entre aplicaciones de software. Según IBM (s.f.), "una API proporciona una manera estandarizada para que diferentes aplicaciones interactúen entre sí". Las APIs permiten a los desarrolladores acceder a ciertas funcionalidades de una aplicación sin necesidad de conocer su implementación interna. Por ejemplo, una API de un sistema de pagos permite integrar funciones de procesamiento de transacciones en diferentes plataformas.

Las APIs facilitan la interoperabilidad entre sistemas heterogéneos, promueven la reutilización de código y mejoran la escalabilidad de las aplicaciones. Están presentes en innumerables ámbitos, desde aplicaciones móviles hasta servicios web y dispositivos IoT. (IBM, 2024)

REST: Definición y Principios

REST (Representational State Transfer) es un estilo de arquitectura para diseñar redes y aplicaciones distribuidas. Fue propuesto por Roy Fielding en el año 2000 en su tesis doctoral. REST define un conjunto de restricciones que, si se siguen, permiten que los servicios web sean escalables, confiables y rápidos.

Principios fundamentales de REST incluyen:

Uso de HTTP como protocolo de comunicación.

Operaciones basadas en los métodos HTTP (GET, POST, PUT, DELETE).

Manipulación de recursos identificados a través de URIs (Uniform Resource Identifiers).

Mensajes autodescriptivos.

Hipermedia como motor del estado de la aplicación (HATEOAS).

Según Red Hat (s.f.), "una API REST usa solicitudes HTTP para realizar operaciones CRUD (crear, leer, actualizar, eliminar) sobre los recursos". (Red Hat, 2023)

RESTful: Qué significa y su aplicación

El término "RESTful" se refiere a las APIs que cumplen con los principios arquitectónicos de REST. Una API RESTful utiliza HTTP de manera adecuada, maneja recursos de forma consistente, y asegura que las respuestas del servidor contienen toda la información necesaria para que el cliente pueda entender y utilizar el servicio.

Las APIs RESTful son ampliamente utilizadas en el desarrollo de aplicaciones web y móviles por su simplicidad, escalabilidad y facilidad de integración. Su diseño permite que sistemas distribuidos trabajen de manera eficiente y que evolucionen sin afectar a los clientes existentes. Amazon Web Services (AWS, s.f.) señala que "las API RESTful son la opción preferida para el diseño de aplicaciones modernas basadas en la nube". (Amazon aws, 2024)

Buenas prácticas en el desarrollo de APIs

Desarrollar APIs efectivas implica seguir una serie de buenas prácticas, entre las cuales destacan:

Uso claro de los verbos HTTP: GET para obtener datos, POST para crearlos, PUT para actualizarlos y DELETE para eliminarlos.

Diseño consistente de URIs: Utilizar nombres de recursos en plural (ej. /usuarios, /productos).

Manejo adecuado de códigos de estado HTTP: Retornar respuestas significativas como 200 OK, 201 Created, 400 Bad Request, 404 Not Found, entre otros.

Versionado de la API: Incluir versiones en la ruta de la API (ej. /api/v1/usuarios) para facilitar cambios futuros sin afectar clientes existentes.

Documentación clara: Usar herramientas como Swagger para documentar la API y permitir que otros desarrolladores entiendan cómo interactuar con ella.

Seguridad: Implementar mecanismos de autenticación y autorización, como OAuth 2.0.

Paginar respuestas: Para recursos grandes, usar paginación para evitar sobrecargar el cliente o el servidor.

Akamai (s.f.) recomienda "utilizar autenticación robusta y cifrado TLS en todas las API REST para proteger la información confidencial". (Akami, 2025)

Referencias

Akami. (2025). *Akami*. Obtenido de Prácticas API REST:

<https://www.akamai.com/es/blog/security/rest-api-security-best-practices>

Amazon aws. (2024). *Amazon aws*. Obtenido de Que es un APIrestFull:

<https://aws.amazon.com/es/what-is/restful-api/>

IBM. (2024). *IBM*. Obtenido de IBM: <https://www.ibm.com/es-es/topics/rest-apis>

Red Hat. (31 de Julio de 2023). *Que es una API de REST?* Obtenido de Red Hat:

<https://www.redhat.com/es/topics/api/what-is-a-rest-api>