Universidad

Universidad Autónoma de Sinaloa

Carrera

Lic. en Informática

Materia

Desarrollo web del lado del servidor

Actividad

Investigar que es una API Rest

Grupo

2-3

Fecha

08/06/2025 Culiacán, Sinaloa

Maestro

José Manuel Cazarez Alderete

Alumna

Núñez Sarabia Jessica Anahí

ÍNDICE

API

Una API (Interfaz de Programación de Aplicaciones) es un conjunto de reglas y protocolos que permiten la comunicación entre diferentes sistemas de software. Funciona como un puente entre aplicaciones, facilitando el intercambio de datos y funcionalidades sin que los usuarios necesiten conocer los detalles internos de cada sistema.

Características

- Interoperabilidad: Permiten que diferentes aplicaciones se comuniquen entre sí, independientemente del lenguaje de programación o la plataforma en la que estén desarrolladas.
- Modularidad: Facilitan la integración de nuevas funcionalidades sin modificar el código base de una aplicación.
- Seguridad: Controlan el acceso a los datos y servicios mediante autenticación y autorización.
- Estandarización: Siguen protocolos bien definidos para garantizar una comunicación eficiente.

Ejemplo

GET /weather?city=Mexico

```
{
  "temperature": "30°C",
  "humidity": "60%",
  "condition": "Soleado"
}
```

Casos de uso

- Aplicaciones móviles: Comunicación entre la app y el servidor.
- Sistemas de e-commerce: Gestión de productos, pedidos y usuarios.
- Servicios de streaming: Acceso a contenido multimedia.
- Redes sociales: Interacción entre usuarios y publicaciones.
- Integración de sistemas: Conexión entre diferentes plataformas.

API REST

Una API de REST es una interfaz de programación de aplicaciones (API) que sigue los principios de diseño del estilo de la arquitectura REST. REST significa transferencia de estado representacional y consiste en un conjunto de reglas y recomendaciones para diseñar una API web.

Características

- Interfaz uniforme: Las solicitudes deben seguir una estructura consistente y estandarizada.
- Desacoplamiento cliente-servidor: El cliente y el servidor deben ser independientes, solo conectados mediante la API.
- Sin estado: Cada solicitud debe contener toda la información necesaria para procesarla, sin depender de sesiones previas.
- Capacidad de almacenamiento en caché: Se puede almacenar información para mejorar el rendimiento y reducir la carga en el servidor.
- Arquitectura en capas: Una API REST puede involucrar múltiples capas de intermediarios sin afectar la comunicación.
- Código bajo demanda (opcional): En algunos casos, la API puede entregar código ejecutable para ciertos propósitos.

Funcionamiento

- Las API REST utilizan el protocolo HTTP para gestionar operaciones estándar de bases de datos, siguiendo el esquema CRUD (Crear, Leer, Actualizar y Eliminar):
- GET: Recupera información de un recurso.
- POST: Crea un nuevo recurso.
- PUT: Actualiza un recurso existente.
- DELETE: Elimina un recurso.
- La comunicación entre cliente y servidor generalmente se realiza utilizando JSON como formato de intercambio de datos, por su legibilidad y compatibilidad con múltiples lenguajes.

RESTFUL

REST (Representational State Transfer) es un estilo de arquitectura de software que define cómo los sistemas pueden comunicarse de manera eficiente a través de HTTP. Fue introducido por Roy Fielding en el año 2000 y ha revolucionado el desarrollo de aplicaciones web y servicios digitales. REST es una alternativa más sencilla y flexible a protocolos como SOAP, permitiendo el intercambio de datos en formatos como JSON y XML.

Ventajas

- Separación entre cliente y servidor: La interfaz de usuario y el almacenamiento de datos están desacoplados, lo que mejora la escalabilidad y portabilidad.
- Visibilidad y fiabilidad: Facilita la migración a otros servidores y la evolución independiente de los componentes.
- Independencia de plataformas y lenguajes: Compatible con servidores en PHP, Java, Python, Node.js, entre otros.
- Uso de HTTP: Aprovecha los métodos estándar como GET, POST, PUT, DELETE para gestionar recursos.

Principios

- Interfaz uniforme: Los recursos deben tener nombres claros y estructurados.
- Stateless: Cada solicitud es independiente y no depende de sesiones previas.
- Operaciones específicas: Cada endpoint tiene un propósito bien definido.
- Cliente-servidor: El servidor expone los recursos y el cliente los consume sin depender de su implementación interna.

REFERENCIAS

1. https://www.redhat.com/es/topics/api/what-is-a-rest-api
2. https://aws.amazon.com/es/what-is/api/
3. https://www.ibm.com/mx-es/think/topics/rest-apis
4. https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/
5. https://bravedeveloper.com/2021/09/01/que-es-rest-restful-api-restful-y-json/