INSTITUTO TECNOLOGICO DE CIUDAD MADERO

MATERIA: Tec. de Servicios Web

REUMEN: API RESTful, REST

ALUMNO: Victor Alfonso Vera Meza

NUMERO DE CONTROL: 18070547

MAESTRO: Fernando Manzanares

CARRERA: Ingeniería en sistemas computacionales

¿Qué es API RESTful?

La API RESTful es una interfaz que dos sistemas de computación utilizan para intercambiar información de manera segura a través de Internet. La mayoría de las aplicaciones para empresas deben comunicarse con otras aplicaciones internas o de terceros para llevar a cabo varias tareas. Por ejemplo, para generar nóminas mensuales, su sistema interno de cuentas debe compartir datos con el sistema bancario de su cliente para automatizar la facturación y comunicarse con una aplicación interna de planillas de horarios. Las API RESTful admiten este intercambio de información porque siguen estándares de comunicación de *software* seguros, confiables y eficientes**.**

¿Qué es una API?

Una interfaz de programa de aplicación (API) define las reglas que se deben seguir para comunicarse con otros sistemas de software. Los desarrolladores exponen o crean API para que otras aplicaciones puedan comunicarse con sus aplicaciones mediante programación. Por ejemplo, la aplicación de planilla de horarios expone una API que solicita el nombre completo de un empleado y un rango de fechas. Cuando recibe esta información, procesa internamente la planilla de horarios del empleado y devuelve la cantidad de horas trabajadas en ese rango de fechas.

¿Qué es REST?

La transferencia de estado representacional (REST) es una arquitectura de software que impone condiciones sobre cómo debe funcionar una API. En un principio, REST se creó como una guía para administrar la comunicación en una red compleja como Internet. Es posible utilizar una arquitectura basada en REST para admitir comunicaciones confiables y de alto rendimiento a escala. Puede implementarla y modificarla fácilmente, lo que brinda visibilidad y portabilidad entre plataformas a cualquier sistema de API.

Principios de REST

- *Interfaz uniforme*: Esta basado en recursos y estos deben ser sustantivos en plural por ejemplo: libros, alumnos, peliculas, etc.
- *Stateless*: REST no tiene estados, esto quiere decir que una llamada al API es independiente de otra llamada y no depende de ella, sin embargo sí se puede usar caché para reducir el tiempo de espera en consultas GET.
- *Operaciones específicas*: Cada acción u operacion sobre un recurso está bien definido y tiene un claro propósito, no es "multifuncional" si un endpoint por ejemplo es para insertar nuevos clientes, no debería también insertar pedidos.
- *Sintaxis estandarizada*: Cada recurso es accesibe únicamente desde su URI.
- *Cliente-Servidor*: El servidor hace el procesamiento del API y expone los recursos a uno o muchos clientes, que pueden ser una aplicación de escritorio, una página web, etc. El cliente debe ser independiente del servidor y toda comunicación a él se debe dar mediante el API.

RESTful

Los servicios RESTful se basan en la manipulación de recursos. Los recursos pueden contener datos actualizados de forma estática o dinámica. Identificando los recursos en la aplicación, puede hacer que el servicio resulte más útil y más fácil de desarrollar.

Beneficios de las API REST

Las API REST son las API más utilizadas debido a los diversos beneficios que ofrecen: este es el motivo por el que los desarrolladores prefieren trabajar con las API REST:

- Sencillez y facilidad de uso:

Las API REST son relativamente sencillas de entender y utilizar, ya que siguen métodos HTTP estándar (GET, POST, PUT, DELETE) y utilizan convenciones estándar para la representación de recursos (normalmente JSON o XML).

- Escalabilidad:

Los servicios RESTful se pueden escalar fácilmente horizontalmente, ya que no tienen estado. Cada solicitud de un cliente contiene toda la información necesaria para cumplir con esa solicitud, lo que facilita la distribución y el equilibrio de carga.

- Flexibilidad:

REST permite una amplia gama de formatos de datos, pero JSON es el más utilizado debido a su simplicidad y facilidad de análisis. Esta flexibilidad hace que las API REST sean adecuadas para varios tipos de clientes y aplicaciones.

- Apatridia:

Cada solicitud de un cliente a una API REST es independiente y sin estado. El servidor no necesita almacenar ninguna información sobre el cliente entre solicitudes, lo que simplifica el diseño y la implementación tanto del cliente como del servidor.

- Interoperabilidad:

Las API REST son independientes de la plataforma y se pueden implementar en cualquier lenguaje de programación. Los clientes pueden consumirlos fácilmente en diferentes tecnologías, lo que conduce a una mayor interoperabilidad.

- Capacidad de almacenamiento en caché:

REST admite mecanismos de almacenamiento en caché, lo que permite a los clientes almacenar en caché las respuestas. Esto mejora el rendimiento y reduce la carga en el servidor, especialmente para los recursos que no cambian con frecuencia.

- Interfaz uniforme:

Es más fácil para los desarrolladores trabajar con API RESTful ya que tienen una interfaz uniforme y consistente. Esta uniformidad se puede atribuir a la estandarización de los URI de recursos, los métodos HTTP y los formatos de representación.

- Latencia reducida:

La naturaleza sin estado de REST elimina la necesidad de que el servidor almacene información sobre el cliente, lo que reduce la latencia general. Los clientes pueden incluir toda la información necesaria en cada solicitud y los servidores responden con los datos requeridos.

- Facilidad de integración:

El proceso de desarrollo con API RESTful es bastante sencillo ya que se pueden integrar fácilmente con diferentes sistemas.

- Seguridad:

Puede proteger fácilmente las API REST con protocolos HTTPS estándar y establecer un canal de comunicación seguro entre clientes y servidores. Además, también puede implementar mecanismos de autenticación y autorización para controlar el acceso a los recursos.