

Desarrollo de API REST

Docente: Ing. Gerardo Pineda Zapata



Actividad Exposicion#2 –
Restricciones REST

Alumno:

David De Jesus Perez Compean – 21100266

Fecha: 11 de febrero de 2025



INDICE

Contenido:

Introducción	3
Restricciones del Estilo Arquitectónico REST	3
Cliente-Servidor:	3
Interfaz Uniforme:.....	4
Sin Estado:	4
Cacheable:	4
Sistema en Capas:	5
Código bajo demanda (Opcional):.....	5
Importancia de las Restricciones REST	5

Introducción

En el mundo del desarrollo de software, las arquitecturas de servicios web han evolucionado significativamente, y una de las más utilizadas es REST (Representational State Transfer). REST es un estilo arquitectónico para el diseño de servicios web que se basa en un conjunto de restricciones que garantizan escalabilidad, eficiencia y facilidad de mantenimiento. Estas restricciones definen las reglas fundamentales para que un sistema RESTful mantenga su compatibilidad con los principios del protocolo HTTP. Este reporte explora en detalle estas restricciones, su importancia y sus implicaciones en la construcción de servicios web modernos.

Restricciones del Estilo Arquitectónico REST

Para que un servicio sea considerado RESTful, debe cumplir con un conjunto de restricciones definidas por Roy Fielding en su disertación doctoral en el año 2000. Estas restricciones son las siguientes:

Cliente-Servidor:

- La arquitectura REST separa a los clientes de los servidores, lo que permite que ambas partes evolucionen de manera independiente.
- Los clientes solicitan recursos y los servidores responden con la representación de dichos recursos.

Interfaz Uniforme:

- Define una interfaz estándar que simplifica la interacción entre los clientes y los servidores.
- Incluye cuatro principios clave:
 - Identificación de recursos mediante URIs.
 - Manipulación de recursos a través de representaciones.
 - Uso de mensajes autodescriptivos.
 - Hipermedios como motor del estado de la aplicación (HATEOAS).

Sin Estado:

- Cada solicitud HTTP del cliente al servidor debe contener toda la información necesaria para procesarla.
- No se mantiene información de estado en el servidor entre solicitudes, lo que mejora la escalabilidad y la confiabilidad.

Cacheable:

- Los clientes pueden almacenar respuestas y reutilizarlas para reducir la carga en el servidor y mejorar el rendimiento.
- Las respuestas HTTP deben incluir información sobre si pueden ser cacheadas y por cuánto tiempo.

Sistema en Capas:

- Una arquitectura REST puede estar compuesta por múltiples capas (balanceadores de carga, servidores proxy, cachés, etc.).
- Cada capa debe operar de manera independiente sin que los clientes sepan cuántas capas existen.

Código bajo demanda (Opcional):

- Permite que los servidores envíen código ejecutable al cliente, como scripts o applets.
- No es una restricción obligatoria, pero puede mejorar la flexibilidad de los clientes.

Importancia de las Restricciones REST

El cumplimiento de estas restricciones garantiza que los servicios web sean escalables, mantenibles y eficientes. REST ha sido ampliamente adoptado en la industria debido a su simplicidad y compatibilidad con el protocolo HTTP, lo que facilita la interoperabilidad entre sistemas. Además, permite la creación de Apis seguras y fáciles de consumir, lo que es clave en entornos distribuidos y aplicaciones modernas basadas en microservicios.