



UNIVERSIDAD AUTONOMA  
DE QUERÉTARO  
FACULTAD DE INFORMÁTICA



Jorge Rojas Cárdenas  
Investigación sobre API REST  
Mauricio Ibarra Corona  
247936  
13/09/19

Las API son es la colección de definiciones y protocolos que son utilizados para desarrollar e integrar software de aplicaciones, es decir es la forma en que los sitios webs intercambian datos, que normalmente los formatos son en JSON o XML, algunas de sus ventajas es que ofrece diferentes datos tanto para un usuario en un formato estándar hasta ofrecer datos a nuestra página web y a aplicaciones que se ejecutan en nuestro móvil o celular.

Mientras el termino REST, por su acrónimo en inglés (REpresentational State Transfer), fue introducido por el creador de las versiones 1.0 y 1.1 de HTML, Roy Fielding en el año del 2001. Con la finalidad de que este de la facilidad de ser usado por todos los lenguajes de programación y que permita testearse, de igual manera en REST el cliente y servidor sean independientes uno del otro, lo que lo hace que la comunicación sea más ligera y optima, ya que hace que la interacción cliente y servidor se mejoren al tener un número muy limitado de operaciones, debido a que cada instrucción o verbo tiene un significado específico, ejemplo:

- GET se usa para obtener recursos:
  - Antes del recurso va el verbo, es decir lo que queremos que haga, seguido por el recurso que queremos, en este caso obtener, ejemplo:
    - GET <http://www.recursoasolicitar.com/api/desarrollo/1>
- POST se usa para crear un recurso en el servidor
  - Al crear un recurso puede arrojar diferentes resultados, como los siguientes:
    - Error 403 Acceso prohibido
    - Error 400 petición incorrecta
    - Error 500 al crear un recurso del lado del servidor
    - 201 Recurso creado
- PUT se usa para cambiar el estado de un recurso o actualizarlo
  - Si actualizas es significado de cambiar todos los datos, aunque existe un método llamado PATCH el cual permite cambiar solo ciertos datos, sin embargo, muchos framework no lo soportan, puede dar diferentes resultados, uno ya antes mencionados en el método POST con la diferencia del siguiente:
    - 200 recurso modificado
- DELETE se usa para eliminar un recurso
  - Al ejecutar este verbo pude dar los siguientes resultados:
    - 200 Ok
    - 404 not found
    - 500 server error

API REST tiene diversas reglas:

- Interfaz uniforme
  - La interfaz se basa en recursos, ejemplo recurso estudiante (id, nombre, apellido, expediente)

- El servidor mandara los datos vía HTML, JSON, XML, protegiendo lo que envíe, ya que no se podrán ver por parte del cliente
  - Al llegar al cliente este puede borrarlo o cambiarlo
- Peticiones sin estado:
  - No necesitas poner http al requerir un recurso, bórralo etc.
  - El servidor no guarda los datos de la consulta previa
- Cacheable
  - En la web el cliente puede cachear las respuestas del server
  - En las respuestas se puede marcar como cacheables
  - En futuras peticiones el cliente puede reutilizarlas
- Separación de cliente y servidor
  - Como se mencionó anteriormente la comunicación es más rápida y por lo tanto el frontend y backend se hacen por separado
  - Se puede cambiar el cliente y servidor
- Sistema de capas
  - El cliente puede estar conectado mediante la interfaz al server o un intermediario
  - Al cliente solo le interesa que la API REST funcione correctamente
- Código bajo demanda
  - Los servers pueden aumentar o definir funcionalidades en el cliente dándole cierta lógica que puede ejecutar, como:
    - Componentes compilados como applets de java
    - JavaScript en el cliente