

Proyecto Prácticas Fidesol



Hecho por: FºJavier Vicaria Garcia
Curso: 1ºDAW A

ÍNDICE

ÍNDICE.....	1
- ¿Cómo funciona una API?.....	2
- ¿Qué principios presenta el estilo arquitectónico REST?.....	2
- ¿Qué beneficios aportan las API RESTful?.....	3
- ¿Cómo funciona una API RESTful?.....	3
- ¿Qué métodos de autenticación de las API RESTful hay?.....	4
- Intercambio de información a través de JSON.....	4
- Características del formato JSON.....	5
- Tipos de datos JSON.....	5
- Beneficios del formato JSON.....	5
- LISTA DE RECURSOS.....	5

Una *API* es una aplicación que sirve como herramienta para que otras 2 aplicaciones puedan intercambiar información o datos entre sí de manera segura, confiable y eficiente.

- ***¿Cómo funciona una API?***

Una aplicación cualquiera expone una *API* la cual solicita un tipo de datos. Cuando se recibe esta información, la procesa y devuelve los valores que se han pedido anteriormente. Es importante destacar que una *API*, en términos generales es una puerta de enlace entre los clientes y los recursos de las webs.

Una *API* puede regirse por las condiciones impuestas por la arquitectura de software *REST*, lo cual la convierte en una *API REST*. Por otro lado, los servicios web que también usan la arquitectura *REST* son llamados “*servicios web RESTful*”

- ***¿Qué principios presenta el estilo arquitectónico REST?***

La interfaz uniforme es un principio clave en el diseño de servicios web *RESTful*. Su objetivo es garantizar que la información intercambiada entre el servidor y el cliente siga un formato estándar, facilitando la interoperabilidad.

Esta interfaz establece cuatro restricciones arquitectónicas:

1. *Identificación de recursos*: Se realiza mediante un identificador uniforme (URI).
2. *Autonomía del cliente*: La representación del recurso debe contener metadatos suficientes para permitir modificaciones o eliminación.
3. *Mensajes autodescriptivos*: El servidor proporciona instrucciones sobre cómo procesar la información.
4. *Descubrimiento de recursos*: Se incluyen hipervínculos para que los clientes accedan dinámicamente a otros recursos necesarios.

En esencia, este enfoque mejora la escalabilidad y la eficiencia de los sistemas distribuidos.

Otros principios son por ejemplo:

- 1.- La tecnología sin estado: consiste en que el servidor complete todas las solicitudes del cliente de manera independiente de las otras solicitudes.
- 2.- Sistema por capas: se organizan los servidores en diferentes niveles(seguridad, lógica, almacenamiento...) invisibles para el cliente.
- 3.- Almacenamiento de caché: guarda respuestas temporales para reducir la carga del servidor.
- 4.- Código bajo demanda: hace posible que el servidor envíe fragmentos de código al cliente para mejorar su funcionamiento.

- ***¿Qué beneficios aportan las API RESTful?***

Los sistemas *API REST* admiten la escalabilidad sin provocar cuellos de botella que reduzcan el rendimiento además de una separación total entre el cliente y el servidor. Los componentes del servidor evolucionan de manera independiente de manera que los cambios de plataforma no afectan a la aplicación y por último, las *API REST* son completamente independientes de la tecnología que se utiliza, es decir, nada afecta al diseño de la *API*.

- ***¿Cómo funciona una API RESTful?***

Desde el cliente se le envía una solicitud al servidor, este autentica al cliente y se asegura que tiene el derecho de hacer la solicitud emitida. Después el servidor recibe la solicitud y la procesa para luego devolver una respuesta al cliente con información que le indica al cliente si la solicitud se ha procesado correctamente.

Dicha solicitud contiene los siguientes componentes:

- 1.- Identificador único de recursos: Un URL parecido a la dirección de un sitio web.
- 2.- Método: indica lo que el servidor debe de hacer con cada recurso

- 3.- Get: accede a los recursos que la URL especifica.
- 4.- Post: envían datos al servidor.
- 5.- Put: actualizan los recursos del servidor.
- 6.- Delete: elimina recursos.
- 7.- Encabezados HTTP: intercambian metadatos cliente-servidor.
- 8.- Datos: se incluyen en los métodos para que funcionen de manera correcta.
- 9.- Parámetros: proporcionan al servidor más detalles sobre las instrucciones.

- ***¿Qué métodos de autenticación de las API RESTful hay?***

Durante la autenticación se verifica una entidad para establecer una “*confianza*”.

Existen cuatro métodos de autenticación:

- 1.- HTTP: se puede utilizar directamente al implementar la API REST. Se divide en ***básica*** (nombre y contraseña) y ***del portador***(a través de un token cifrado con una cadena de caracteres).
- 2.- Claves API: el servidor asigna un valor único generado a un cliente por primera vez. Estas son menos seguras.
- 3.- OAuth: contraseña + token adicional para la autorización.

- ***Intercambio de información a través de JSON.***

Json se usa para especificar datos en aplicaciones y sitios web. Es ligero ya que los datos, por lo general, no ocupan mucho como en otros lenguajes. Sus características que han hecho que sea muy adecuado en el ámbito del desarrollo son algunas como su facilidad de escritura y lectura, lo poco que pesa, la facilidad con la que se analiza y genera y por último su compatibilidad con la mayoría de los lenguajes de programación.

JSON se adapta a cualquier stack de tecnologías por ser independiente del lenguaje de programación. También se utiliza para guardar la configuración de las

aplicaciones, además muchos sistemas de bases de datos están empezando a adoptar este formato como notación para almacenar datos.

- ***Características del formato JSON***

- Estructura de datos basada en pares clave: similar a los tipos de datos utilizados en otros lenguajes como los objetos y los arrays.
- Ligereza y legibilidad: no consume apenas recursos, lo cual hace que JSON sea muy bueno en dispositivos de bajos recursos.
- Multiplataforma: se puede usar en cualquier lenguaje lo cual hace que sea increíblemente flexible y adaptable.

- ***Tipos de datos JSON***

- Cadenas
- Números
- Booleanos
- Objetos
- Arrays
- Null

- ***Beneficios del formato JSON***

JSON es particularmente fácil de leer y escribir debido a su simplicidad, esto también ayuda a la eficiente transmisión de datos, lo que hace que la comunicación entre el cliente y el servidor sea ideal. Por último, JSON tiene compatibilidad con una grandísima gama de lenguajes de programación, por lo que lo convierte en una opción muy recomendable, es muy complicado encontrar un lenguaje que no admita este formato.

- **LISTA DE RECURSOS**

- Video explicativo: [click aquí](#)
- Páginas adicionales:
 - [¿Qué es una API de REST?](#)
 - [{json:api}](#)