



TECNICATURA SUPERIOR EN

Telecomunicaciones

Arquitectura y Conectividad

MÓDULO II: Familia de Protocolos de Comunicaciones II

Familia de Protocolos de Comunicación II

¿Qué es la Comunicación REST?

- La comunicación REST (Representational State Transfer) es un estilo arquitectónico para sistemas distribuidos que permite la comunicación entre diferentes aplicaciones web utilizando el protocolo HTTP.
- En la comunicación REST, las aplicaciones web interactúan a través de una API (Interfaz de Programación de Aplicaciones) que utiliza un conjunto de verbos HTTP (GET, POST, PUT, DELETE, entre otros) para enviar y recibir datos en formato JSON (JavaScript Object Notation) o XML (Extensible Markup Language).
- En este modelo de comunicación, cada recurso (como un objeto o una colección de objetos) se representa con una URL (Uniform Resource Locator) única y es posible acceder a él mediante el uso de los verbos HTTP mencionados anteriormente. Esto permite a las aplicaciones web solicitar y recibir información específica, actualizarla o eliminarla según sea necesario.
- La comunicación REST se utiliza ampliamente en el desarrollo de aplicaciones web y móviles, ya que permite una comunicación eficiente y escalable entre sistemas distribuidos, independientemente del lenguaje de programación utilizado para desarrollarlos.

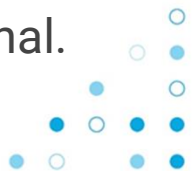


Comunicaciones REST

Solicitudes y comunicaciones

REST requiere que un cliente realice una solicitud al servidor para enviar o modificar datos. Una solicitud consiste en:

- Un verbo o método HTTP que define qué tipo de operación realizará el servidor.
- Un encabezado, con el encabezado de solicitud que pasa información sobre la solicitud.
- Un camino o ruta al servidor, como por ej: <https://app.aluracursos.com/course/http-base-internet>
- Información en el cuerpo de la solicitud, siendo esta información opcional.



Comunicaciones REST

Métodos HTTP

En la aplicación REST, los métodos más utilizados son:

- El método GET es el método más común, generalmente se usa para solicitar a un servidor que envíe un recurso;
- El método POST está diseñado para enviar datos de entrada al servidor. En la práctica, a menudo se usa para admitir formularios HTML;
- El método PUT edita y actualiza documentos en un servidor;
- El método DELETE que, como su nombre lo indica, elimina un determinado dato o colección del servidor.



Comunicaciones REST

Códigos de respuesta

A cada respuesta que devuelve la aplicación REST se le envía un código que define el estado de la solicitud. Por ejemplo:

- 200 (OK) solicitud cumplida con éxito.
- 201 (CREADO) objeto o recurso creado con éxito.
- 204 (SIN CONTENIDO) objeto o recurso eliminado con éxito.
- 400 (MALA SOLICITUD) ocurrió un error en la solicitud (puede haber numerosas causas).
- 404 (NO ENCONTRADO) ruta o colección no encontrada.
- 500 (ERROR INTERNO DEL SERVIDOR), se ha producido algún error del servidor.



Comunicaciones REST

Compatibilidad con el Navegador

	PC					Móvil					
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
CONNECT	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
DELETE	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
GET	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes



Comunicaciones REST

Compatibilidad con el Navegador

HEAD	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
OPTIONS	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
POST	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
PUT	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
TRACE	⊛ ?	⊛ ?	⊛ ?	⊛ ?	⊛ ?	⊛ ?	⊛ ?	⊛ ?	⊛ ?	⊛ ?	⊛ ?



Comunicaciones REST

Diferencias entre SOAP vs REST

Los servicios web son los responsables entre la comunicación entre máquinas, en muchos casos son los que proveen los datos las aplicaciones front end.

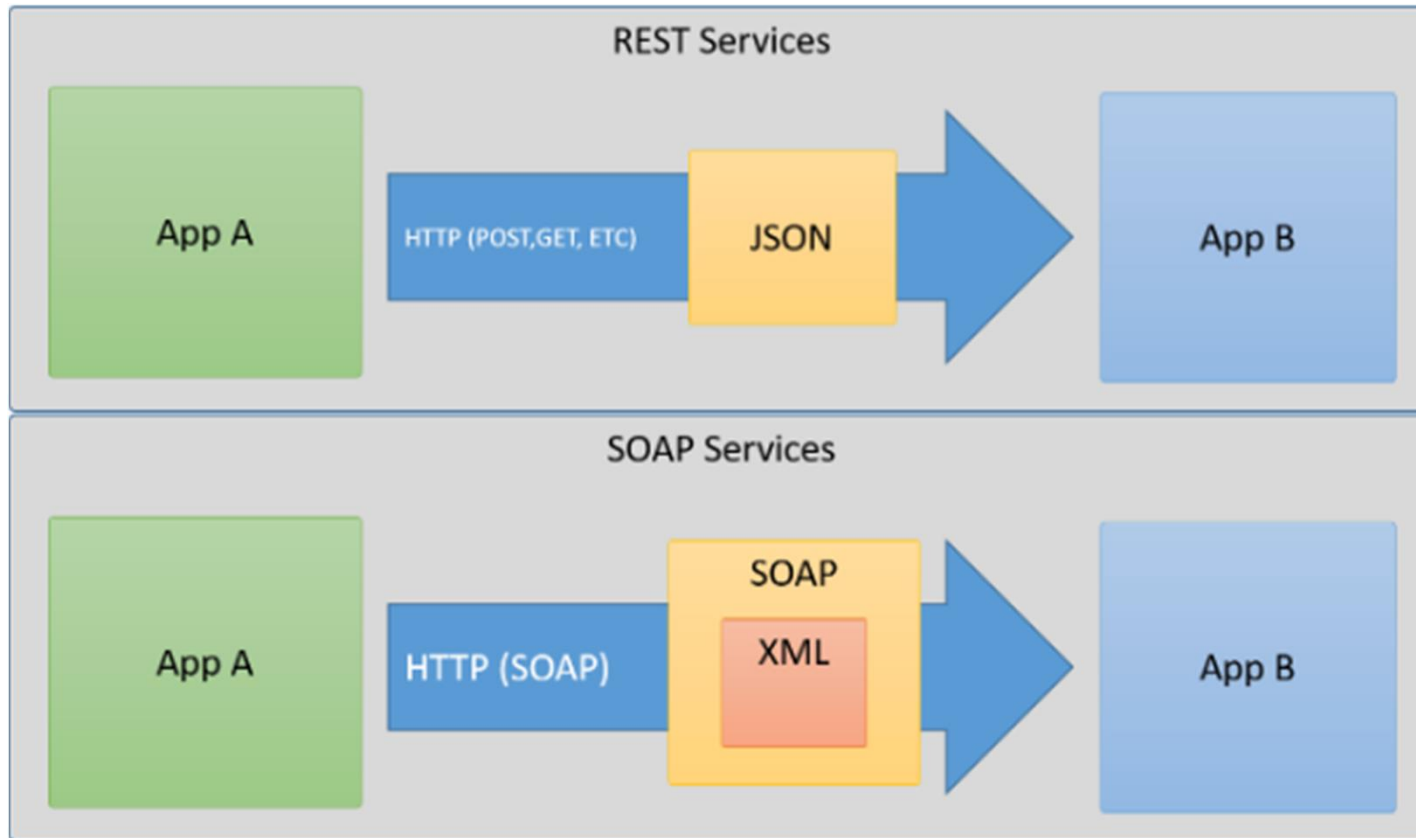
Los servicios webs disponen de una API que es consumida para la transferencia de la información, esta API puede tener diferentes arquitecturas.

Durante mucho tiempo se utilizó SOAP como el protocolo de comunicación, pero la necesidad de la aparición de desarrollar aplicaciones ligeras ha hecho que aparezca y tome fuerza REST ganando mucha popularidad en la actualidad.



Comunicaciones REST

Diferencias entre SOAP vs REST



Comunicaciones REST

	SOAP	REST
Significado	Simple Object Access Protocol	Representational State Transfer
Diseño	Protocolo estandarizado con reglas predefinidas a seguir.	Estilo arquitectónico con pautas sueltas y recomendaciones.
Enfoque	Basado en funciones (datos disponibles como servicios, por ejemplo: «getUser»)	Controlado por datos (datos disponibles como recursos, por ejemplo, «user»).
Control de estado	Sin estado de forma predeterminada, pero es posible hacer que una API SOAP sea controlada.	Sin estado (no hay sesiones del lado del servidor).



Comunicaciones REST

Almacenamiento en caché	Las llamadas a la API no se pueden almacenar en caché.	Las llamadas a la API se pueden almacenar en caché.
Seguridad	WS-Security con soporte SSL. Cumplimiento de ACID incorporado.	Soporta HTTPS y SSL.
Rendimiento	Requiere más ancho de banda y potencia de cálculo.	Requiere menos recursos.
Formato de mensaje	Sólo XML.	Texto sin formato, HTML, XML, JSON, YAML y otros.
Protocolo (s) de transferencia	HTTP, SMTP, UDP, y otros.	Solo HTTP



Comunicaciones REST

Recomendado para	Aplicaciones empresariales, aplicaciones de alta seguridad, entorno distribuido, servicios financieros, pasarelas de pago, servicios de telecomunicaciones.	APIs públicas para servicios web, servicios móviles, redes sociales.
Ventajas	Alta seguridad, estandarizada, extensibilidad.	Escalabilidad, mejor rendimiento, facilidad de navegación, flexibilidad.
Desventajas	Menor rendimiento, más complejidad, menos flexibilidad.	Menos seguridad, no apto para entornos distribuidos.



Comunicaciones REST

¿Qué es el Formato de Datos JSON?

- El formato de datos JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos que se utiliza para transmitir información estructurada entre aplicaciones. Fue diseñado originalmente para ser utilizado con JavaScript, pero hoy en día es un formato comúnmente utilizado en una amplia variedad de lenguajes de programación.
- JSON utiliza una sintaxis sencilla basada en pares de nombre-valor, que permite representar datos estructurados como objetos y matrices. Los datos se organizan en una estructura jerárquica de objetos y valores, que puede anidarse de manera recursiva.
- La sintaxis de JSON es muy similar a la de los objetos de JavaScript, lo que hace que sea fácil de leer y escribir para los desarrolladores familiarizados con este lenguaje. Además, JSON es un formato independiente del lenguaje, lo que significa que puede ser utilizado por cualquier lenguaje de programación que soporte la manipulación de cadenas de texto.
- En resumen, JSON es un formato de datos ligero y fácil de leer que se utiliza para transmitir información estructurada entre aplicaciones.



Formato de Datos JSON

Ejemplo de Formato de datos JSON

```
{
  "titulo": "El gran Gatsby",
  "autor": "F. Scott Fitzgerald",
  "año_publicación": 1925,
  "editorial": {
    "nombre": "Scribner",
    "ubicación": "Nueva York"
  },
  "personajes": [
    {
      "nombre": "Jay Gatsby",
      "edad": 32,
      "profesión": "magnate del petróleo"
    },
    {
      "nombre": "Daisy Buchanan",
      "edad": 23,
      "profesión": "heredera adinerada"
    },
    {
      "nombre": "Tom Buchanan",
      "edad": 30,
      "profesión": "hombre de negocios"
    }
  ]
}
```

Aquí hay un ejemplo de un objeto JSON que representa información sobre un libro:

Este objeto JSON representa un libro llamado "El gran Gatsby" escrito por F. Scott Fitzgerald y publicado por Scribner en 1925. El objeto incluye información sobre la editorial, así como una lista de personajes que aparecen en el libro.



Comunicaciones REST

¿Qué es una API RESTful?

Una API RESTful (Representational State Transfer) es una arquitectura de software para sistemas distribuidos que utiliza el protocolo HTTP para realizar operaciones CRUD (Create, Read, Update, Delete) en recursos (datos) que se encuentran en un servidor web.

En una API RESTful, los recursos son identificados por URLs (Uniform Resource Locators) y se acceden a través de los métodos HTTP estándar, como GET, POST, PUT y DELETE. Cada método tiene una función específica:

- GET: obtener información del servidor.
- POST: enviar información al servidor para crear un nuevo recurso.
- PUT: actualizar un recurso existente en el servidor.
- DELETE: eliminar un recurso del servidor.

La API RESTful utiliza un formato de intercambio de datos común como JSON o XML, lo que permite que diferentes sistemas puedan interactuar y compartir información de forma estandarizada y sin depender de plataformas específicas.

En resumen, una API RESTful es una forma de comunicación entre sistemas basada en el protocolo HTTP, que permite la manipulación de recursos a través de URLs y métodos HTTP estándar, y utiliza un formato de intercambio de datos común para permitir la interoperabilidad entre diferentes sistemas.



Comunicaciones REST

Librerías HTTP a REST

Existen muchas librerías en diferentes lenguajes de programación que se utilizan para realizar solicitudes HTTP y consumir servicios REST. A continuación, menciono algunas de las más populares:

- En Python, las librerías más utilizadas son **requests** y **httplib2**.
- En Java, una de las librerías más utilizadas es **OkHttp**, aunque también existen otras como **HttpClient** y **RestTemplate**.
- En Ruby, la librería más utilizada es **Faraday**.
- En JavaScript, se utiliza comúnmente la librería **Axios**.
- En PHP, las librerías más populares son **Guzzle** y **HTTP_Request2**.
- En C#, la librería más utilizada es **HttpClient**.
- En Go, la librería estándar **net/http** es muy utilizada para realizar solicitudes HTTP.

Estas son solo algunas de las muchas librerías disponibles en diferentes lenguajes de programación para realizar solicitudes HTTP y consumir servicios REST. La elección de la librería adecuada dependerá de las necesidades específicas de cada proyecto y del lenguaje de programación utilizado.



¡Muchas gracias!