

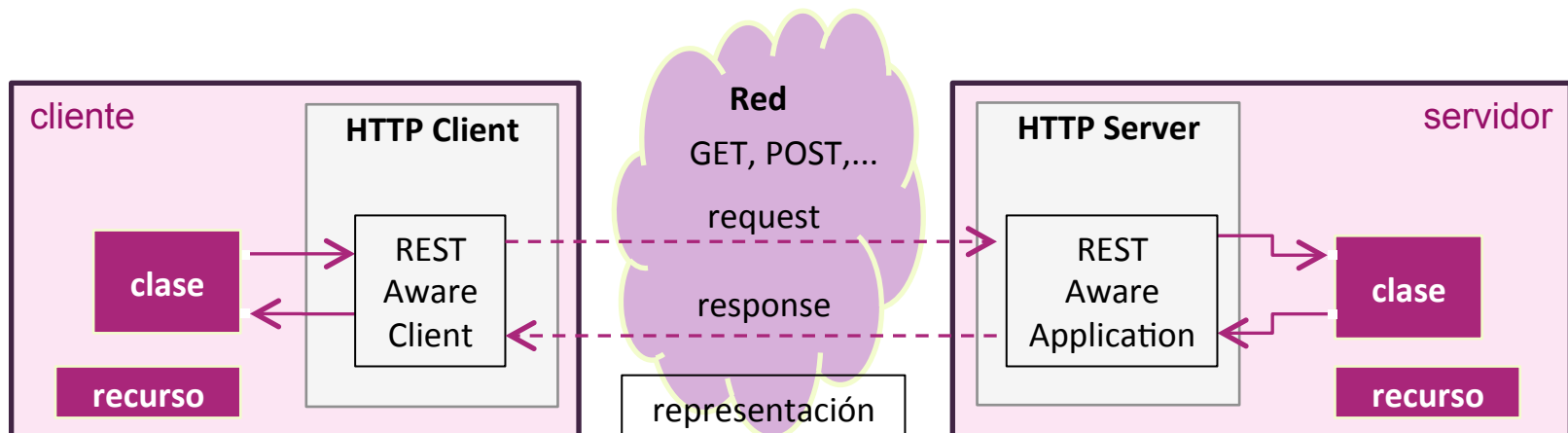
Servicios REST

desarrollo de servicios REST

arquitectura REST

REST (Representational State Transfer) es un estilo de arquitectura para sistemas distribuidos, desarrollada por la W3C, junto con el protocolo HTTP.

- Las arquitecturas REST tienen clientes y servidores.
- El cliente realiza un envío (request) al servidor, el cual lo procesa y retorna una respuesta al cliente.
- Las peticiones y respuestas son construidas alrededor de representaciones de recursos. Recurso es una entidad, y representación es cómo se formatea.



desarrollo de servicios REST

Web Services RESTful

RESTful:

- Una API del tipo **RESTful**, o RESTful Web Service, es una API web implementada con HTTP y los principios REST, con los siguientes aspectos:
 - Una URI base del servicio.
 - Un formato de mensajes, por ejemplo JSON o XML.
 - Un conjunto de operaciones, que utilizan los métodos HTTP (GET, PUT, POST o DELETE).
 - La API debe manejar hipertextos.
- A diferencia de los Web Services basados en SOAP, **no hay un estándar** comúnmente aceptado para los RESTful. Esto es porque REST es una arquitectura, mientras que SOAP es un protocolo.
- Esta desventaja se compensa con la simplicidad de su utilización y el bajo consumo de recursos durante el binding. Esto es especialmente útil en aplicaciones para dispositivos móviles.

desarrollo de servicios REST

utilización de métodos HTTP

Con REST, los **métodos HTTP** se asocian a **tipos de operaciones** sobre recursos. El uso comúnmente aceptado es el siguiente:

- **GET**: Para recuperar la representación de un recurso. Es idempotente, es decir, si se invoca múltiples veces, retorna el mismo resultado.
- **POST**: Para crear un recurso, o para actualizarlo. También, por las características del método, se utiliza para envíos grandes, o para evitar limitaciones de los otros métodos.
- **PUT**: Para actualizar un recurso, ya que POST no es idempotente.
- **DELETE**: Para eliminar un recurso.
- **OPTIONS**: Se puede utilizar para hacer un "ping" del servicio, es decir, verificar su disponibilidad.
- **HEAD**: Para buscar un recurso o consultar estado. Similar a GET, pero no contiene un body.

desarrollo de servicios REST

definición

WADL (Web Application Description Language):

- Es el descriptor de servicio de aplicaciones basadas en HTTP, y en particular REST
- Equivalente al WSDL utilizado en SOAP.
- **No es un estándar** y por el momento no hay planes para hacerlo.
- Estructura:
 - application
 - grammars: incluye los XSD.
 - resources: define la URI
 - resource: define la operación
 - method: puede ser GET, POST, ...
 - request: parámetros en el envío.
 - response: incluye representaciones.

desarrollo de servicios REST

comparación con SOAP

- Ventajas con respecto a SOAP:
 - El mensaje se construye directamente con HTTP, lo que da más simplicidad y rendimiento.
 - Los mensajes normalmente son más pequeños.
 - Permite definir interacciones con estado en forma nativa, a través del uso de la sesión HTTP.
 - Se puede invocar en forma sencilla desde JavaScript, lo que facilita su uso en páginas dinámicas con Ajax.
- Desventajas con respecto a SOAP:
 - SOAP es un protocolo estándar abierto.
 - SOAP facilita más la interoperabilidad entre sistemas, al ser estándar.
 - SOAP facilita más la depuración.

buenas prácticas

buenas prácticas en el uso de REST

El uso de REST tiene asociadas buenas prácticas, que es recomendable seguir para que su utilización sea una ventaja. Algunas de ellas son:

- Definir un servicio en forma simple.
- Implementar un cliente de prueba.
- Agregar validaciones en el servidor, y manejar los códigos de error HTTP.
- Utilizar cache cuando sea aplicable. Esto permite ahorrar invocaciones cuando se conoce la respuesta.