

# SERVICIOS REST



# Fundamentos REST

- Los servicios REST (Representational State Transfer), exponen a través de Internet una serie de recursos a los que se puede acceder con peticiones HTTP sencillas (tipo GET, PUT, POST, DELETE, etc.)
- Los recursos consisten en datos que se ofrecen (pedidos, pedido de un determinado identificador) u operaciones sobre esos datos (modificar un pedido, añadir un pedido,...)
- Cada recurso se identifica por: URL, método HTTP, variables, tipo de respuesta, tipo consumido

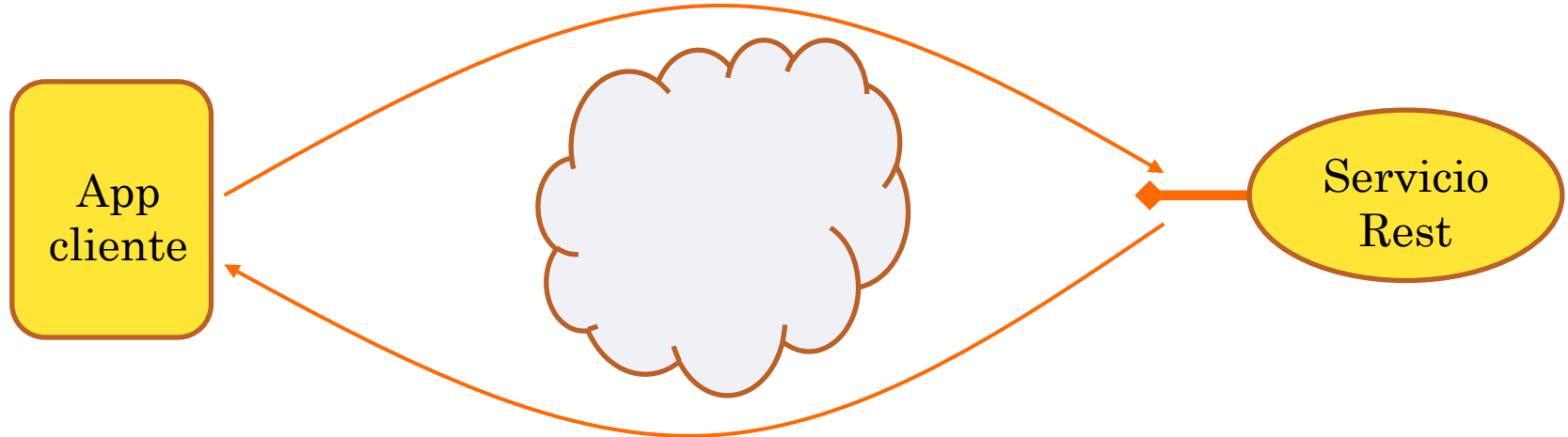


# Formato de intercambio de datos

Petición HTTP (GET,POST,PUT,DELETE)

URL

	<b>Text, XML, JSON,..</b>
--	---------------------------



Respuesta HTTP

	<b>Text, XML, JSON,..</b>
--	---------------------------



# Ventajas de REST

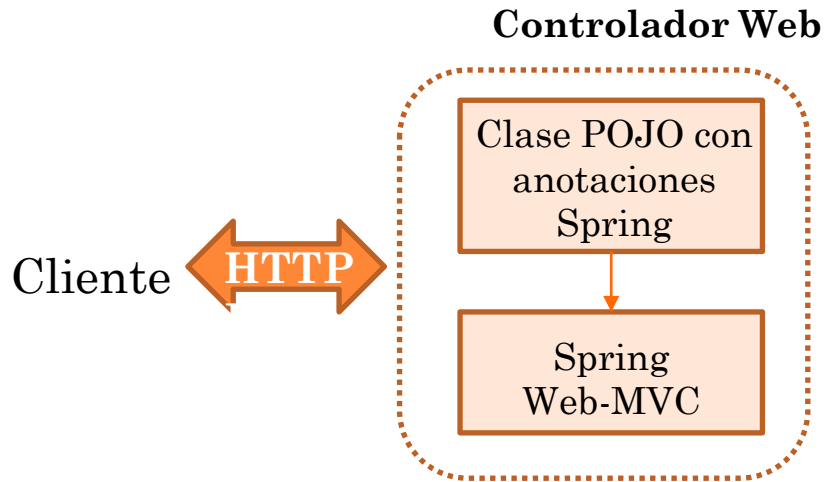
Entre las ventajas que nos proporciona la arquitectura de servicios REST frente a los servicios XML podemos destacar:

- **Acceso más simple.** Para acceder a un servicio REST **no necesitamos incluir ningún documento XML en el cuerpo de la petición**, basta con enviar una simple petición HTTP, que hará que se ejecute una operación dentro del servicio.
- **Flexibilidad.** Los servicios REST nos ofrecen una gran flexibilidad a la hora de intercambiar datos con la aplicación cliente, puesto que se admiten múltiples formatos de. Los servicios web REST no solo admiten datos en XML, sino también en **JSON**, binario, texto plano, etc. Además, la forma de estos datos puede ser cualquiera, no tiene que ajustarse a una determinada estructura. Y lo mismo sucede con los datos de la respuesta.
- **Eficiencia.** La construcción y manipulación de documentos XML resulta bastante tediosa y consume muchos recursos de la máquina. Al poder emplear otros formatos de datos mucho más eficientes se mejora el rendimiento de las aplicaciones.



# Servicios REST con Spring

- El módulo Web-MVC proporciona el soporte necesario para la creación de servicios REST con Spring
- Incluye anotaciones específicas de Spring



# Estructura controlador REST

## ➤ Clase POJO con anotaciones spring MVC

```
@RestController
public class ClaseServicio{
    @GetMapping(..)
    public String metodo(..){..}

    @PostMapping(..)
    public List<> metodo2(..){..}
    :
}
```



# Principales anotaciones Spring

- **@RestController**. Indica que la clase es un controlador REST.
- **@GetMapping, @PostMapping, @PutMapping, @DeleteMapping**. Asocian a los métodos del servicio un determinado método HTTP. A través de su atributo *value*, se indica también la url a la que se asociará el método.
- **@PathVariable**. Asocia una variable de la URL a un parámetro de método.
- **@RequestBody**. Asocia el contenido del cuerpo de la petición a un parámetro objeto, dentro del método de respuesta.

