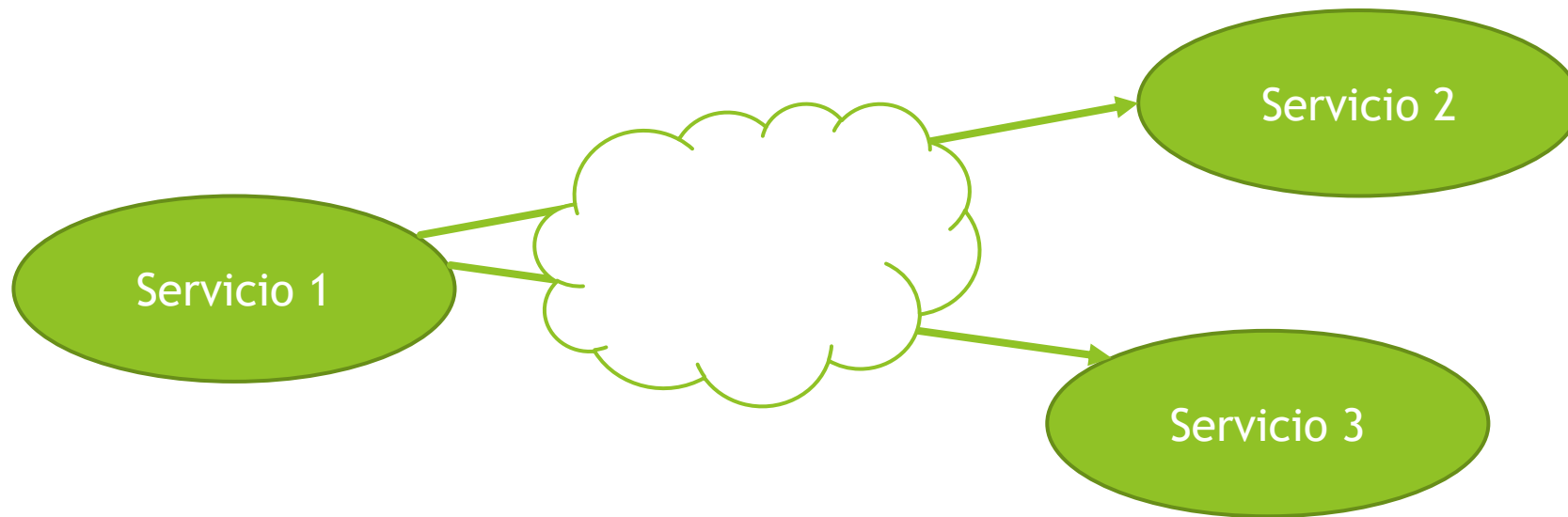


Interacción con microservicios



Interacción entre microservicios

- ▶ La clase `RestTemplate`
- ▶ Forma parte del módulo Web de Spring, por lo que hay que añadir el starter Web.
- ▶ Su creación se define en una clase de configuración

```
@Bean  
public RestTemplate getTemplate(){  
    return new RestTemplate();  
}
```

Interacción entre microservicios

```
@Bean  
public RestTemplate getTemplate(){  
    return new RestTemplate();  
}
```

- Para inyectarlo en una variable:

```
@Autowired  
RestTemplate template;  
}
```

Petición GET con RestTemplate

- ▶ Se realiza mediante el método `getForObject`
- ▶ `getForObject(String url, Class<T> responseType, Object...uriVariables)`

```
String url="http://localhost:8080/cursos";  
Curso[] cursos=template.getForObject(url,Curso[].class);  
Curso cur = template.getForObject(url+"/{name}",Curso.class,"PHP");
```

Petición POST con RestTemplate

- ▶ Se realiza mediante el método **postForObject** si se devuelve resultado

```
postForObject(String url, Object request, Class<T> responseType, Object...uriVariables)
```

- ▶ Se realiza mediante el método **postForLocation** si no se devuelve resultado:

```
postForLocation(String url, Object request, Object...uriVariables)
```

```
String url="http://localhost:8080/curso";  
Curso curso=new Curso("PHP", 30, "tardes";  
template.postForLocation(url, curso);
```

Petición PUT con RestTemplate

- ▶ Se asume que no devuelve resultado
- ▶ `void put(String url, Object request, Object...uriVariables)`

```
String url="http://localhost:8080/curso";  
Curso curso = template.getForObject(url+"/{name}",Curso.class,"PHP");  
Curso.setDuration(70);  
template.put(url,curso);
```

Petición DELETE con RestTemplate

- ▶ Se asume que no devuelve resultado
- ▶ `void delete(String url, Object request, Object...uriVariables)`

Método exchange

- ▶ Permite realizar cualquier tipo de petición
- ▶ Adecuado cuando los métodos anteriores no permiten hacer la petición deseada (por ejemplo, petición PUT con devolución de resultados):

`ResponseEntity<T> exchange(String url, HttpMethod método,
HttpEntity<?> entity, Class<T> responseType, Object...uriVariables)`

```
Curso curso = template.getForObject(url+"/{name}",Curso.class,"PHP");  
Curso.setDuration(70);  
ResponseEntity<Curso[]> rp =  
    tmp.exchange(url, HttpMethod.PUT, new HttpEntity<Curso>(),Curso[].class;  
Curso[] cursos= rp.getBody();
```


Petición POST con RestTemplate

- ▶ Se realiza mediante el método **postForObject** si se devuelve resultado

```
postForObject(String url, Object request, Class<T> responseType, Object...uriVariables)
```

- ▶ Se realiza mediante el método **postForLocation** si no se devuelve resultado:

```
postForLocation(String url, Object request, Object...uriVariables)
```

```
String url="http://localhost:8080/curso";  
Curso curso=new Curso("PHP", 30, "tardes";  
template.postForLocation(url, curso);
```

Ejemplo

