

- 1. Introducción
- 2. Mashup
 - 2.1 ¿Qué opción elegir?
- 3. Tarea

1. Introducción

Los servicios web permiten a tus aplicaciones comunicarse con otras utilizando la web (el protocolo HTTP) como medio de transmisión.

En esta unidad habrá que **crear aplicaciones utilizando diversos servicios web**

Un servicio web implementado mediante REST puede:

- Utilizar una estructura de URIs para acceder a los recursos gestionables mediante el servicio web
 - `http://miweb.com/productos`
 - `http://miweb.com/productos/1`
- Usar los distintos **métodos HTTP** (POST, DELETE,...)
- Utilizar **JSON** o XML en sus comunicaciones (o incluso ambos)

2. Mashup

Una aplicación web híbrida, también conocida como **mashup** se caracteriza por combinar datos y/o funcionalidades procedentes de **diversos orígenes** para formar un nuevo tipo de aplicación o servicio.

Los tipos de fuentes de información más habituales que se utilizan en una aplicación web híbrida son:

- Información proveniente de **servicios web**
- Información generada y gestionada por el **propietario de la aplicación web híbrida**, como pueden ser datos internos de una empresa
- Datos utilizando protocolos de redifusión web (**sindicación web**) como **RSS** o **Atom**
- **Web scraping**: técnica utilizada para extraer datos de sitios web que no ofrezcan ningún servicio web (ojo con la legalidad del procedimiento)

¿Y cómo consumíamos estos servicios web REST?

Tenemos 2 opciones:

- **Desde el backend**: utilizando, por ejemplo, la librería **cURL** de PHP
- **Desde el frontend**: utilizando, por ejemplo, `fetch` en Javascript

Un ejemplo ya visto con cURL:

```
$url_servicio = "http://zoologico.com/api";
$curl = curl_init($url_servicio);
//establecemos el verbo http que queremos utilizar para la petición
curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "GET");
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
```

```
$respuesta_curl = curl_exec($curl);  
curl_close($curl);  
  
$respuesta_decodificada = json_decode($respuesta_curl);
```

2.1 ¿Qué opción elegir?

Ventajas de consumir APIs desde el **Frontend**:

- **Rendimiento: respuestas rápidas** para el usuario al evitar la latencia del servidor.
- Mayor **interactividad en tiempo real**: ideal para aplicaciones con **actualizaciones frecuentes**
- **Menos carga en el servidor**: al delegar las solicitudes al cliente
- **Desarrollo más rápido**: al evitar configuraciones adicionales en el servidor

Ventajas de consumir APIs desde el **Backend**:

- **Seguridad**: más seguro ya que las claves de las API y la lógica de negocio están en el servidor
- **Control completo**: mayor control sobre el flujo de datos
- **Escalabilidad**: permite escalar el servidor según las necesidades
- **Manejo de errores**: centralizamos los errores en el servidor y tendremos más información para gestionarlos

3. Tarea

Un ejemplo de aplicación web híbrida, es una aplicación que utilice la API de **Google Maps** y muestre información de ubicación geográfica de las franquicias de un bar almacenadas en una **base de datos**.

Al pulsar cada marcador del mapa se consultará otro servicio web que mostrará las **opiniones** de los clientes que han dejado en esa página.

Otro ejemplo: aplicación que lea distintos canales de **Telegram** de periódicos (Washington Post, Reuters, CNN, BBC, Yahoo News...) y se conecte a un **servicio web** para obtener de cada una de las noticias **ciudades y países** que se citen en ellas.

Luego se utilizaría la API de **Google Maps** para posicionar cada una de las ciudades o países de los que se hayan encontrado noticias.

En caso de que se pinche en cada marca del mapa se cargarían las noticias relacionadas con esta ciudad y utilizando **Flickr** se obtendrían 2 fotografías del lugar.