



## Tema 7: Aplicaciones web interactivas y aplicaciones web híbridas

### ¿Qué aprenderás?

- Identificar aplicaciones web híbridas y saber clasificarlas.
- Conocer las ventajas de la reutilización de código y uso de recursos ya existentes.
- Utilizar librerías de código para incorporar funcionalidades a una aplicación web.

### ¿Sabías que...?

- RSS y Atom usan XML para la redifusión de contenido.
- En el año 2000, eBay lanzó su API. 2 años más tarde, Amazon hizo lo propio.
- La técnica Screen scraping para proveer de contenido de terceros a nuestra web no es recomendable y está en desuso.



## 1. Aplicaciones web híbridas. Mashups

### 1.1. ¿Qué es una aplicación web híbrida?

Imaginemos que se nos presenta un proyecto para desarrollar una aplicación web que se encargue de la venta o intercambio de productos o servicios por cercanía. Por ejemplo, busco profesor de repaso, o me ofrezco como tal, para dar clases en mi barrio. Podemos imaginar las funcionalidades de esta aplicación y cómo implementar la mayoría, pero en seguida nos viene a la cabeza un elemento clave: un mapa donde poder ubicar a clientes y proveedores. Obviamente, el desarrollo del mapa es algo que complicaría el desarrollo de la aplicación enormemente y dispararía los costes. ¿Qué alternativa tenemos? Existe Google Maps. ¿Podríamos incluir funcionalidades de Google Maps en nuestra aplicación?

Con el ejemplo anterior podemos definir ya las aplicaciones web híbridas. También conocidas como mashups, son aplicaciones que incluyen contenido o servicios de terceros.

La mayoría de veces, para crear un mashup se hace uso de una API, que ofrece una integración de recursos de terceros de forma fácil y rápida. Este tipo de servicios son ofrecidos y consumidos a través del protocolo de comunicación HTTP.

Aplicaciones web que, por ejemplo, incluyan un vídeo de Youtube, o un mapa sin procesar de GoogleMaps, no son mashups. Hace falta que la aplicación web acceda a la información de forma externa o procese los datos del sitio que nos ofrece el servicio.

### 1.2. Arquitectura de una aplicación web híbrida

La arquitectura de un mashup consta de tres partes:

- El proveedor de contenidos.
- El sitio mashup.
- El cliente.

A continuación, veremos más en detalle estos tres actores y la relación existente entre ellos.

#### 1.2.1. El proveedor de contenidos

Es el sitio de donde provienen los datos. Para cogerlos, solemos hacer uso de una API y protocolos web como RSS, Atom, Screen scraping (en desuso), REST o servicios web.

- RSS: Really Simple Syndication (“sindicación realmente simple”) es una tecnología que nos permite distribuir información por la web con código XML, de manera que puede ser visualizada desde otra página web. Suele usarse para difundir información actualizada y noticias sobre algún tema.



- Atom: esta tecnología es similar a RSS en cuanto distribuye contenido mediante un archivo XML. Pero incorpora mejoras respecto a RSS, ya que el documento XML de Atom es capaz de contener más información, más compleja y consistente. Además, esta tecnología no ofrece algunos problemas que sí tenía RSS debido a la incompatibilidad de las diferentes versiones existentes.
- Screen scraping: es una técnica mediante la cual se accede al contenido HTML de una web y se “raspa” en búsqueda de los fragmentos de código que nos interesa incorporar a nuestra aplicación híbrida. Es una técnica en desuso y no recomendada, ya que es compleja y propensa a provocar fallos inesperados. Además, muchas webs no autorizan acceder a sus recursos de esta manera.
- REST: es una forma de acceder a servicios web mediante una URL y obtener información en múltiples formatos, como XML, XHTML y JSON. Las aplicaciones web que ofrecen servicios usando REST se denominan Restfull web.
- API: Application Programming Interface (“interfaz de programación de aplicaciones”) ofrece un conjunto de funciones y métodos agrupados en una librería que nos permite utilizar software de terceros e incorporarlo en nuestra aplicación. El cliente realiza la llamada a las funciones de la API que se ejecutan en el servidor y generan una respuesta, la cual se utiliza para generar la aplicación web híbrida. La API es desarrollada por el proveedor de contenidos, y en ella se documenta el uso de las funciones incluidas en ella.

### 1.2.2. El sitio mashup

Es la nueva aplicación web generada usando información de la que el proveedor es dueño. Ofrece un nuevo servicio y utiliza algunos de los elementos de comunicación con el proveedor vistos en el apartado anterior.

### 1.2.3. El cliente

Es la interfaz de usuario del mashup que se visualiza a través del navegador web, el cual puede usar lenguajes como Javascript o AJAX para generar el contenido a mostrar.

## 1.3. Tipos de aplicaciones web híbridas

---

Los mashups se pueden clasificar atendiendo a diversos aspectos. Por ejemplo, están los mashups de consumidores, ejemplificados por las muchas aplicaciones que utilizan Google Maps, y los mashups de negocio, más enfocados a la agregación de datos y la creación de un sitio colaborativo orientado al negocio.

No obstante, se suele hacer una división de las aplicaciones web híbridas en función de los objetivos que éstas persiguen y el tipo de información que presentan. Existen 4 grupos:



- Mapas: en este grupo se encuentran los mashups que muestran información localizada sobre un mapa.
- Vídeo y fotos: son sitios que acceden a servidores web que almacenan grandes cantidades de fotos y vídeos. Estos servidores normalmente ofrecen APIs que permiten operaciones para compartir, filtrar, buscar, editar, etc.
- Búsqueda y compras: esta categoría engloba sitios web que permiten realizar búsquedas de las mejores ofertas comerciales. En ellos podemos ver productos de diversos sitios de venta, como Amazon, eBay, etc.
- Enciclopedias y noticias: son aplicaciones cuya información se nutre de sitios de noticias. Suelen usar RSS y Atom para traer los datos de fuentes confiables. Estos mashups pueden ofrecer funcionalidades para agrupar noticias según ciertos parámetros del usuario.

#### 1.4. Uso de la API de Google Maps

En esta sección veremos una de las API más usadas para crear mashups como es la de Google Maps. Veremos cómo incorporar un mapa a nuestra aplicación web de forma sencilla, y las posibilidades que ofrece la API. El lenguaje de programación que se usará es Javascript, por tanto, se ejecutará en el cliente.





## Test de autoevaluación

---

A las aplicaciones web híbridas se les conoce como:

- a) Servicio web
- b) API
- c) Mashup
- d) RSS

La arquitectura de una aplicación web híbrida está formada por:

- a) El proveedor de contenido, el sitio mashup y el cliente
- b) El proveedor de contenido y el mashup
- c) El mashup y el cliente
- d) El servidor, el mashup y el cliente

¿Qué protocolo de comunicación utiliza una aplicación web híbrida para consumir los servicios de terceros?

- a) XML
- b) SOAP
- c) HTML
- d) HTTP



## Recursos y enlaces

---

- [API de Google Maps en Javascript](#)

## Conceptos clave

---

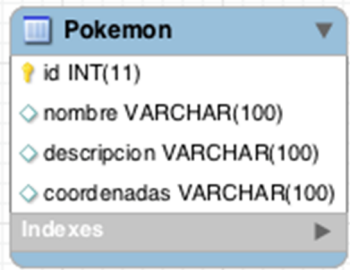
- **Mashup:** es una aplicación web que utiliza contenidos o servicios de terceros y los combina para crear una aplicación web nueva.
- **API:** Application Programming Interface (“interfaz de programación de aplicaciones”) ofrece un conjunto de funciones y métodos agrupados en una librería que nos permite utilizar software de terceros e incorporarlo en nuestra aplicación.



## Ponlo en práctica

### Actividad 1

Crea una aplicación web híbrida que muestre la ubicación de pokemons almacenados en una base de datos en un mapa de Google Maps. Los datos se muestran en la siguiente imagen:



id	nombre	descripcion	coordenadas
1	Pikachu	Pokemon amarillo y eléctrico	{lat: 41.385775, lng: 2.165547}
2	Charmander	Pokemon que se convierte en dragón	{lat: 41.386273, lng: 2.159388}
3	Squirtle	Pokemon que se parece a una tortuga	{lat: 41.386901, lng: 2.170525}

La aplicación deberá cargar un mapa en el que se visualizará un marcador por cada registro en la base de datos.

Al clicar sobre un marcador, se debe abrir un cuadro de texto con los campos “nombre” y “descripción” de la base de datos.



SOLUCIÓN