

## TEMA 9. DESARROLLO DE APLICACIONES WEB HÍBRIDAS.

---

### Objetivos

- Reconocer las ventajas de la reutilización de código y el aprovechamiento de la información existente.
- Identificar las librerías de código y las tecnologías para la creación de aplicaciones web híbridas.
- Crear aplicaciones web que recuperen y procesen repositorios de información ya existentes.
- Crear repositorios específicos a partir de información existente en internet y almacenes de información.
- Utilizar librerías de código para incorporar funcionalidades a una aplicación web.
- Programar servicios y aplicaciones web utilizando información y código generado por terceros.
- Probar, depurar y documentar las aplicaciones generadas.

### **9.1.- Reutilización de código e información.**

Una aplicación web híbrida es una página o aplicación Web que utiliza contenidos o servicios de terceros y los combina para crear una aplicación Web nueva. También se denomina mashup.

Las aplicaciones Web híbridas son accedidas a través de una interfaz pública o usando una API. El concepto de mashup implica una integración fácil y rápida. Por ejemplos: YouTube, Google, Flickr ofrecen mashups para integrar sus servicios con otras aplicaciones Web.

Incluir código embebido en una aplicación web no la convierte en una aplicación híbrida, ya que para ser considerada como híbrida debe acceder a la información externa o procesar los datos que nos ofrece el servicio dotando a la web de un valor añadido.

### **9.2.- Arquitectura de una aplicación web híbrida.**

Una aplicación web híbrida consta de tres partes:

- Proveedor de contenidos
- Servidor de la aplicación web
- Cliente web

RSS y Atom son tecnologías basadas en una estructura XML que permite compartir contenidos de una web determinada.

El Proveedor de contenidos es la fuente de datos y se accede a ellos a través de un interfaz pública o API que es un conjunto de funciones y procedimientos que ofrecen una biblioteca para utilizar otro software.

Las principales características son que pertenece al proveedor que se encarga de la robustez y el control de errores. Está documentado y su uso es sencillo.



El servidor del mashup es la aplicación Web que provee un nuevo servicio utilizando diferentes fuentes de información de las que no es propietario. El cliente Web es la interfaz de usuario del mashup.

Formas de acceder a una página Web híbrida o a un proveedor de contenidos:

- El usuario accede directamente al contenido que ofrece el proveedor. El contenido se mezcla en el cliente a través de Javascript.
- Acceso al proveedor desde el sitio Web del mashup. El contenido se mezcla en el servidor de aplicaciones y después se envía al navegador Web del cliente.
- El usuario accede al servidor de páginas híbridas desde el navegador.

Categorías de mashup:

- **Mashups de mapas:** muestran información sobre un mapa o imagen satélite. Se utilizan en aplicaciones Web en las que es necesario precisar una situación determinada.
- **Mashups de fotos y videos:** generan el contenido a partir de servidores Web que almacenan grandes cantidades de fotografías y vídeos. El proveedor de contenidos permite realizar tareas sobre ellas como: editar, ordenar, coleccionar, compartir, etc.
- **Mashups de compras:** permite realizar búsquedas de las mejores ofertas comerciales.
- **Mashups de noticias:** seleccionan fuentes de noticias a través de los protocolos RSS y Atom. Existe la posibilidad de agrupar noticias según ciertos parámetros de preferencias del usuario.

### Formato RSS

El lenguaje que utiliza RSS para formar sus documentos es XML. Las primeras etiquetas que conforman un documento RSS son la cabecera con la etiqueta `<?xml version=XXX encoding=XXX ?>`

Posteriormente la etiqueta `<rss></rss>` que engloba todos los elementos contenidos en el RSS. la etiqueta en el bloque principal `<channel> </channel>` y dentro todos los items `<item></item>`.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<rss version="2.0"
  xmlns:media="http://search.yahoo.com/mrss/">

  <channel>
    ...items...
  </channel>
</rss>

```

Dentro de cada item disponemos la información de cada elemento (título, enlace, descripción, autor y fecha):

```

<item>
  <title><![CDATA[Nieve para el fin de semana]]></title>
  <link><![CDATA[http://tiempo.com/0001.html]]></link>
  <description><![CDATA[Ya no nieva tanto]]></description>
  <author><![CDATA[Javier]]></author>
  <pubDate><![CDATA[30/01/2012]]></pubDate>
</item>

```

### Ejemplo: Leer Noticias RSS

```

<style>
  .noticia {
    border: 1px solid black;
    width: auto;
    margin: 15px;
    padding: 5px;
  }
</style>
<?php
function rssReader($url, $begin="", $end=""){
  $rss = simplexml_load_file($url);
  if($rss){
    $items = $rss->channel->item;
    $out="";
    foreach($items as $item){
      $out.=$begin.'<a href="'. $item->link.
        '" target="_blank" >'. $item->title.'</a>';
      if($item->children('media', true)->thumbnail->count()>0){
        $thumbAttr = $item->children('media', true)->
          thumbnail->attributes();
        $url=$thumbAttr['url'];
        $out.="<br><img src='". $url. "'>";
      }
      $out.=$end;
    }
  }
  return $out;
}
echo rssReader('http://estaticos.elmundo.es/el mundo/rss/portada.xml',
  '<div class="noticia">', '</div>');
?>

```

## Geolocalización

Podemos realizar tareas de geolocalización a partir de la IP de un usuario o una página.

Para obtener la ip una página web concreta utilizaremos la sentencia

```
gethostbyname('direccionWeb')
```

```
<?php
$i p = gethostbyname(' www. googl e. es' );
echo $i p;
?>
```

Podemos saber el país al que pertenece una dirección ip a través de la librería **geoiploc.php** que se descarga desde **<http://chir.ag/projects/geoiploc/>**

Podemos saber los países y sus códigos con los que trabaja la librería utilizando el array `$GLOBALS['geoipcountry']` cuya **key es el código del país y el valor el nombre del país**.

### Ejemplo:

```
<?php
i ncl ude("geoi pl oc. php");
echo "<pre>";
foreach ($GLOBALS['geoi pcountry' ] as $key => $val or){
    echo $key. ": ". $val or. "<br>";
}
echo "</pre>";
?>
```

```
AD: Andorra
AE: United Arab Emirates
AF: Afghanistan
AG: Antigua and Barbuda
AI: Anguilla
AL: Albania
:      :      :      :
:      :      :      :
```

La librería dispone de la función sobrecargada que nos devuelven el código del país o el nombre del país de una dirección IP.

```
//Devuel ve el código del país correspondiente a la IP
getCountryFromIP($i p);
//Devuel ve el nombre del país correspondiente a la IP
getCountryFromIP($i p, "Name");
```

El array del servidor \$\_SERVER, en su posición ['REMOTE\_ADDR'] almacena la dirección IP desde la cual está viendo la página actual el usuario.

**\$\_SERVER['REMOTE\_ADDR']**

**Ejemplo:**

```
<?php
include("geopl oc. php");
$ip = $_SERVER[' REMOTE_ADDR' ];
echo "Tu ip es: ".$ip."<br>";
$codpais = getCountryFromIP($ip);
echo "El codigo del pais desde el que te conectas es: ";
echo $codpais."<br>";
$nombrepais = getCountryFromIP($ip, "Name");
echo "El nombre del pais desde el que te conectas es: ";
echo $nombrepais."<br>";
?>
```

Tu ip es: X.X.X.X  
El codigo del pais desde el que te conectas es: ES  
El nombre del pais desde el que te conectas es: Spain