

# Aplicaciones web híbridas.



## Caso práctico

Después de semanas de trabajo, **Carlos** da por finalizado el desarrollo de la aplicación en la que ha estado trabajando. Ha tenido que reescribir el código en varias ocasiones, cambiar la apariencia de algunas pantallas del interfaz, y retocar el esquema de la base de datos que habían diseñado en un principio, pero finalmente parece que el trabajo ha dado sus frutos. Habla con **Juan** y entre los dos revisan el resultado. **Juan** está muy contento con lo que ve, y le propone hablar con **Ada**, para mostrarle la aplicación web. Aunque la directora ha seguido el progreso de la misma, en gran parte ha sido en las conversaciones que ha mantenido con **Juan**, y hace ya tiempo que no le informan directamente sobre los últimos avances.



# 1.- Reutilización de código e información.



## Caso práctico

**Ada** revisa la aplicación web y les propone retocar un par de detalles, sobre todo relativos a la apariencia. Con los años de experiencia que tiene en el desarrollo de aplicaciones, les ofrece también algunos consejos sobre la usabilidad de los interfaces, y le indica que es muy importante seguir algunas normas básicas que garanticen la accesibilidad de la aplicación.

Por último, tienen que hablar con su amigo **Esteban** para concretar los detalles de la implantación de la aplicación en las dependencias del cliente. Parece que están a punto de finalizar el proyecto.



La unidad 6 de este módulo tenía por título "Servicios Web". En ella aprendiste a crear y utilizar servicios web, empleando una arquitectura orientada a servicios (SOA). Los principales temas que practicaste en esa unidad son:

- ✓ A utilizar el protocolo SOAP para comunicarte con un servicio web. Con ayuda de la clase SoapClient, intercambiabas peticiones y respuestas en formato SOAP con un servicio web existente.
- ✓ A crear tu propio servicio web. Mediante la clase SoapServer podías publicar tus propias funciones para que fueran accesibles como servicio web mediante SOAP.
- ✓ A procesar los documentos WSDL de descripción de los servicios web, y a crear los documentos WSDL de descripción de tus propios servicios.

Los servicios web permiten a tus aplicaciones comunicarse con otras utilizando la web (el protocolo HTTP) como medio de transmisión. Sin embargo, los mecanismos que has utilizado hasta ahora no son la única forma de implementar y utilizar servicios web. Desde hace un tiempo han ido apareciendo servicios web que utilizan arquitecturas basadas en REST.

REST hace referencia a un conjunto de normas que se pueden utilizar para establecer mecanismos de comunicación con servicios web. Concretamente, un servicio web implementado mediante REST debería al menos:

- ✓ Utilizar una estructura de URLs para acceder a los recursos gestionables mediante el servicio web:  
/articulo/KSTD TG332GBR  
/tienda/CENTRAL
- ✓ Usar los distintos  métodos HTTP para las peticiones. Por ejemplo, se podría utilizar el método HTTP GET para obtener información de un artículo:  
GET /articulo/KSTD TG332GBR  
Y el método HTTP DELETE para borrarlo:  
DELETE /articulo/KSTD TG332GBR
- ✓ No almacenar información sobre el estado: todas las peticiones se tratarán de forma independiente y deben incluir toda la información necesaria para poder atenderla.
- ✓ Utilizar XML o JSON en sus comunicaciones (o incluso ambos).



## Para saber más

REST no es un estándar, pero muchos servicios web actuales se implementan utilizando arquitecturas de tipo REST. Existen en Internet varios documentos sobre REST y ejemplos de su utilización que conviene revisar.

 [REST y Servicios Web.](#) (0.26 MB)

[Serie de artículos sobre REST.](#)

En la presente unidad crearás aplicaciones que utilicen diversos servicios web.

## 2.- Características de las aplicaciones web híbridas.



### Caso práctico

En poco tiempo, **Carlos** modifica los detalles que había apreciado **Ada**. Está contento con el resultado obtenido, pero con la experiencia que ha adquirido en programación web durante su desarrollo, sabe que habría algunos detalles que se podrían añadir a la aplicación y que influirían positivamente en la experiencia del usuario.

**Ada** le llama y le comenta que la próxima semana **Esteban** vendrá a BK Programación a conocer la aplicación. **Carlos** guarda una copia de la aplicación en su estado actual, y decide tomarse esa semana para intentar mejorarla en algunos aspectos. Hace una lista de los detalles que le podría añadir, y al revisarla se da cuenta de que otras aplicaciones que conoce ya los implementan; pero en muchos casos, no son desarrollos propios sino que se basan en servicios ofrecidos por terceros: mapas de Google, imágenes de Flickr,... ¿Podrá aprovechar algún servicio existente e integrarlo en su propia aplicación?



Una aplicación web híbrida, también conocida por su nombre en inglés (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, disponible mediante diversos protocolos y estructurada utilizando formatos de intercambio como JSON o XML.  
En ocasiones el proveedor del servicio ofrece también un interface de programación ([API](#)) para facilitar el acceso a los datos. Es el caso de las API de compañías como Google, Yahoo!, Flickr, Microsoft o Amazon.

En otras ocasiones los datos se ofrecen de forma pública utilizando protocolos de redifusión web (también conocido como sindicación web) como [RSS](#) o Atom, y puede ser necesario procesarlos para extraer la información necesaria.

#### Redifusión web.

- ✓ Información generada y gestionada por el propietario de la aplicación web híbrida, como pueden ser datos internos de una empresa.

De forma menos habitual, podemos encontrarnos aplicaciones web que utilicen técnicas de ingeniería inversa para extraer los datos que se muestran en algunos sitios web, como puede ser el caso de los precios de los productos en las tiendas web. Estas técnicas se conocen por su nombre en inglés: web scraping.

Por ejemplo, podrías montar una aplicación web híbrida que utilice la API de Google Maps, e información de ubicación geográfica de las franquicias de una empresa para mostrar la localización de las tiendas en un mapa.

En esta unidad, vas a programar una aplicación web híbrida para la tienda web con la que has venido trabajando. En este caso se trata de facilitar la gestión de los envíos de las compras.

## Autoevaluación

**Las siglas REST hacen referencia a un estándar que se utiliza en la implementación de servicios web.**

- Verdadero.
- Falso.

### 3.- Utilización de repositorios de información.



#### Caso práctico

**Carlos** se informa sobre los servicios web que pueden serle útiles, y decide añadir a la aplicación una funcionalidad que no tiene: un servicio de gestión de envíos para los productos que se vendan. En realidad nadie ha solicitado esa funcionalidad, pero cree que si es capaz de programarla la empresa de **Esteban** la llegará a utilizar. En ocasiones los clientes no piden una característica simplemente porque desconocen que es posible realizarla. Y cuando comenzó este proyecto en particular, no había nadie en BK Programación con la experiencia suficiente como para orientar correctamente al cliente.



Se pone manos a la obra. Si en la semana que tiene le da tiempo a finalizarla, se la mostrará a **Esteban**. Y si no le da tiempo, echará mano de la versión anterior. De cualquier modo, no es tiempo perdido. Este proyecto le está sirviendo para adquirir experiencia que a buen seguro aprovechará en el futuro inmediato.

Cuando utilices servicios de terceros para desarrollar una aplicación web híbrida, deberás tener en cuenta que en ocasiones existen condiciones y límites al uso que puedes hacer del mismo.

La mayoría de las grandes compañías que proveen servicios web al usuario, como Google o Yahoo!, requieren un registro previo y ofrecen unas condiciones para su uso gratuito que dependen del servicio al que necesites acceder. Algunos de estos servicios ofrecen una versión adicional de pago con mejores condiciones.

Por ejemplo, actualmente Google ofrece los siguientes límites de acceso gratuito para los siguientes servicios:

- ✓ Google Tasks: 5.000 consultas diarias.
- ✓ Google Maps: 25.000 consultas diarias.
- ✓ Google Custom Search: 100 consultas diarias.

En muchas ocasiones, el proveedor del servicio (aunque también puede ser un tercero) ofrece además librerías que facilitan la utilización del servicio desde un lenguaje de programación determinado y ejemplos de utilización del mismo. Por ejemplo, si quieres utilizar la API de Google Tasks existen librerías de programación para los lenguajes Java, Python, PHP y para la plataforma Microsoft.

Para que se pueda verificar la utilización que hace cada usuario de un servicio determinado, es necesario incluir dentro del código que interactúa con el mismo una clave personal que le identifica en el sistema. Por ejemplo, si quieras utilizar la API de Google Books, necesitarás indicar tu código de desarrollador al hacer una consulta de forma similar a:

```
$client->setDeveloperKey('la clave de desarrollador va aquí');
```

De la misma forma, si quieras acceder al servicio de PlaceFinder de Yahoo!, tendrás que indicar en la consulta un identificador de aplicación. Aunque desde 2013 este servicio lo han hecho de pago.

Para obtener las claves que te permitan acceder a los servicios web de Google, deberás darte de alta previamente en la "[Google Console Developers](#)".

Desde ahí debes crear un nuevo proyecto y habilitar APIs que puedes buscar desde la Biblioteca en una página como ésta:

The screenshot shows the Google API Library interface. On the left, there's a sidebar with navigation links: 'Panel de control', 'Biblioteca' (which is selected and highlighted in blue), and 'Credenciales'. The main content area is titled 'Biblioteca' and contains several sections of popular APIs:

- APIs de Google**: Includes Compute Engine API, BigQuery API, Cloud Storage Service, Cloud Dataset API, Cloud Deployment Manager API, Cloud DNS API, and more.
- Aprendizaje automático de Google Cloud**: Includes Vision API, Natural Language API, Speech API, Translation API, and Machine Learning Engine API.
- APIs de Google Maps**: Includes Google Maps Android API, Google Maps SDK for iOS, Google Places API for Android, Google Places API for iOS, and Google Maps Roads API.
- APIs de G Suite**: Includes Drive API, Calendar API, Gmail API, Sheets API, Google Apps Marketplace SDK, and Admin SDK.
- APIs para móviles**: Includes Google Cloud Messaging, Google Play Game Services, Google Play Developer API, and Google Places API for Android.
- APIs de redes sociales**: Includes Google+ API, Blogger API, Google+ Pages API, and Google+ Domains API.
- APIs de YouTube**: Includes YouTube Data API, YouTube Analytics API, and YouTube Reporting API.
- APIs de publicidad**: Includes AdSense Management API, DCM/DFA Reporting And Trafficking API, Ad Exchange Seller API, Ad Exchange Buyer API, DoubleClick Search API, and DoubleClick Bid Manager API.
- Otras APIs populares**: Includes Analytics API, Custom Search API, URL Shortener API, PageSpeed Insights API, Fusion Tables API, and Web Fonts Developer API.

Desde el Panel de Control se pueden ver las APIs habilitadas para nuestro proyecto:

Panel de control - Repartos-Unidad7-DWES - Mozilla Firefox

https://console.developers.google.com/apis/dashboard?project=repartos-unidad7-DWES-P

¿Te gustan nuestras API? Descubre nuestra infraestructura. Regístrate y consigue un crédito de 300 USD para explorar Google Cloud Platform durante 12 meses. [Más información](#)

[IGNORAR](#) [REGISTRATE PARA PROBAR LA VERSIÓN GRATUITA](#)

Google APIs Repartos-Unidad7-DWES

API Administrador de ... Panel de control [+ HABILITAR API](#)

Panel de control Biblioteca Credenciales

API habilitadas Algunas API se habilitan automáticamente

Actividad durante la última hora

1 hora 6 horas 12 horas 1 día 2 días 4 días 7 días 14 días 30 días

Tráfico Solicitudes/segundo No hay tráfico en este periodo.

Errores Porcentaje de solicitudes No hay errores en este periodo.

Mediana de latencia Milisegundos No hay datos de latencia.

API	Solicitudes	Errores	Proporción de errores	Latancia, mediana	Latancia (98%)	Inhabilitar
Google Maps Directions API	-	-	-	-	-	<a href="#">Inhabilitar</a>
Google Maps Geocoding API	-	-	-	-	-	<a href="#">Inhabilitar</a>
Google Maps JavaScript API	-	-	-	-	-	<a href="#">Inhabilitar</a>
Tasks API	-	-	-	-	-	<a href="#">Inhabilitar</a>



Elaboración propia

Existen ciertos servicios web que nos permiten acceder a información privada que almacenan de sus usuarios. Por ejemplo, la API de Google Calendar posibilita gestionar la información que cada usuario mantiene en sus calendarios personales. Si vas a usar un servicio de este tipo (como Google Tasks), necesitarás que tu aplicación solicite permiso a los propietarios de la información antes de poder acceder a la misma. Para ello muchos proveedores de servicios web utilizan un protocolo llamado OAuth (la versión actual es la 2.0).



## Para saber más

Para conocer más detalles acerca del coste de usar los servicios de Google Maps, puedes visitar su web de [precios y planes](#).

Para el servicio web a fecha de mayo de 2017 éstas son las condiciones:

	ESTÁNDAR	PREMIUM
<b>Google Maps JavaScript API</b>	Gratis hasta 25 000 cargas de mapa por día. <sup>3</sup>	Precio basado en el volumen requerido.
<b>Google Static Maps API</b>	USD 0,50 cada 1000 cargas de mapa adicionales, hasta 100 000 diarias, si está habilitada la facturación.	Funciones premium mejoradas:
<b>Google Street View Image API</b>	Uso gratuito ilimitado.	<ul style="list-style-type: none"><li>• Sin publicidad, garantizado.</li><li>• Tamaño de imagen de hasta 2048 x 2048 píxeles.</li></ul>
<b>Google Maps Embed API</b>	----	Para obtener más información, consulta <a href="#">Premium Plan Límites e índices de utilización</a> .

## 3.1.- OAuth2.

El protocolo estándar de autorización OAuth2, permite a una aplicación externa obtener acceso a información de carácter privado a través de un servicio web. Para ello establece un acuerdo de acceso a la misma entre la aplicación externa, el servicio web y el propietario de los datos a los que se solicita el acceso.

Por ejemplo, si una aplicación "X" solicita a Google acceso a los calendarios del usuario "dwes", Google pedirá a "dwes" permiso indicándole qué aplicación es la que solicita el acceso y a qué información. Si el usuario "dwes" otorga permiso, la aplicación "X" podrá acceder a los datos que solicitó a través del servicio de Google.

OAuth2 funciona de la siguiente manera, en nuestro caso supondremos que el solicitante será siempre una aplicación web. Veamos por ejemplo qué sucede cuando nuestra aplicación necesita acceder a información personal del usuario a través del servicio de Google Tasks. En este caso, los pasos que se seguirán son los siguientes:



- ✓ La aplicación web se comunica con el servidor de autorización OAuth2, indicando la información a que quiere acceder y el tipo de acceso a la misma.
- ✓ El servidor de autorización OAuth2 requiere al usuario de la aplicación web a que inicie sesión con su cuenta de Google (si aún no lo ha hecho), y le redirige a una página en la que le pide su consentimiento para otorgar acceso a su información privada.



- ✓ Si el usuario da su consentimiento, el servidor de autorización OAuth2 devuelve a la aplicación web un código de autorización.
- ✓ Antes de poder acceder a la información privada del usuario, la aplicación web debe intercambiar ese código de autorización por otro código de acceso.
- ✓ Utilizando el código de acceso, la aplicación puede utilizar el servicio de Google Tasks para gestionar la información privada del usuario, dentro de los límites de acceso que se han otorgado.
- ✓ Los códigos de acceso tienen un tiempo de vida limitado. Cuando caducan, la aplicación ha de comunicarse de nuevo con el servidor de autorización OAuth2 para obtener un código de refresco.



A continuación veremos un vídeo sobre cómo usar OAuth2 con Google:





## Debes conocer

La documentación oficial de la API Google para usar OAuth2 con PHP se encuentra disponible [aquí](#)



## Para saber más

Existe una extensión PHP para programar las partes cliente y servidor del protocolo OAuth.

[Extensión OAuth.](#)

## Autoevaluación

**Al utilizar OAuth2, cuando tu aplicación solicite información privada de un usuario, deberá acreditar su autorización utilizando:**

- Un código de autorización.
- Un código de acceso.

## 3.2.- JSON y XML.

Muchas de las operaciones que se llevan a cabo cuando utilizas un servicio web implican la obtención de cierta información por parte del mismo. La información obtenida puede ser bastante sencilla o tener cierto grado de complejidad. Por ejemplo, cuando utilizas un servicio de geocodificación para averiguar las coordenadas concretas de una dirección, obtienes básicamente esas coordenadas. Pero cuando utilizas un servicio como Google Directions para averiguar la ruta a seguir entre dos puntos, la respuesta que obtienes es una ruta compuesta por una serie de indicaciones a seguir para llegar al destino.

Los formatos más utilizados por los servicios web para dar formato a esas respuestas son dos: JSON y XML. En algunos casos tendrás que adaptar tu código al tipo de respuesta que ofrece el servicio. Otros servicios permiten que escojas el tipo de respuesta que prefieras. Veamos cómo se pueden procesar de forma sencilla desde PHP mensajes en ambos formatos.

A partir de la versión 5.2 de PHP, se incluye por defecto la extensión JSON. Su funcionamiento es muy sencillo. Incorpora dos funciones para tratar con cadenas de texto en notación JSON:



### Extensión JSON.

- ✓ `json_decode`. Decodifica una cadena de texto JSON y la transforma en un objeto PHP (opcionalmente también se podría convertir en un array).

```
String json = '{"color": "azul", "tamaño": 2, "verde": 3, "rojo": 4}';  
Object obj = json.decode();  
print obj;
```

- ✓ `json_encode`. Función inversa a la anterior. Devuelve una cadena de texto en notación JSON a partir del contenido de una variable PHP.

```
Object obj = {"color": "azul", "tamaño": 2, "verde": 3, "rojo": 4};  
String json = obj.encode();  
print json;
```

Así como las opciones para trabajar con JSON desde PHP están bien definidas, para utilizar XML hay más variedad de herramientas para escoger. En PHP4 podías utilizar dos formas para procesar documentos en formato XML: [DOM](#) y [SAX](#). La extensión SimpleXML, habilitada por defecto a partir de PHP 5.1.2, facilita la tarea de extraer información de un documento XML. Vamos a ver su funcionamiento.

### Extensión SimpleXML.

La extensión convierte un documento XML en un objeto de la clase SimpleXMLElement. Puedes cargar el documento:

- ✓ desde una cadena de texto, utilizando la función simple\_load\_string.

```
SimpleXML $xml = simplexml_load_string($xmlString);
```

- ✓ desde un fichero, utilizando la función simplexml\_load\_file. Puedes utilizar una dirección URI como origen, por ejemplo:

```
SimpleXML $xml = simplexml_load_file('http://www.google.com/Xml/GetDir?path=/etc');
```

Los nodos y atributos del documento XML se convierten en atributos. Los nodos pasan a ser arrays que contienen a su vez como elementos los atributos y subnodos del documento XML. Por ejemplo, para acceder al resumen (summary) de una ruta en formato XML obtenida a partir del servicio Google Directions, podrías utilizar:

```
$ruta = simplexml_load_file('http://maps.google.com/service?');
```

```
$ruta->route[0]->summary
```



## Debes conocer

Es importante que conozcas cómo procesar documentos XML para extraer información de los mismos. En el manual de PHP puedes encontrar información sobre la utilización de la extensión SimpleXML.

[Extensión SimpleXML.](#)

## 4.- Creación de aplicaciones web híbridas.



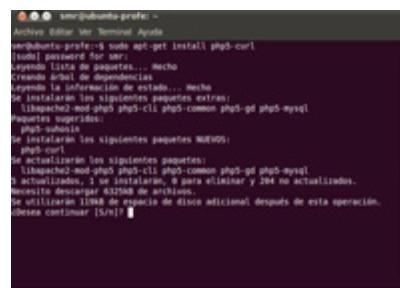
### Caso práctico

Pasa la semana y **Carlos** consigue tener a punto la nueva versión de la aplicación. Para la presentación, pone las dos versiones en funcionamiento, y no le comenta a nadie los nuevos cambios. Si hay algún problema con las últimas modificaciones, echará mano de la versión anterior.

Cuando en BK Programación muestran a **Esteban** la aplicación, éste queda asombrado por el resultado. El nuevo servicio de gestión de envíos sorprende a todos positivamente, especialmente a **Ada** y a **Juan**, que empiezan a darse cuenta de las nuevas capacidades que ha adquirido **Carlos** en las últimas semanas. **Esteban** comenta que muy posiblemente les sea útil, especialmente al poder consultar la información de los envíos desde un dispositivo móvil. Pasarán un par de meses probando la aplicación, y a continuación plantea la posibilidad de tener una nueva reunión para hablar de posibilidades de ampliación y de otros proyectos. Ha quedado muy contento con el trabajo de **Carlos** y quiere seguir contando con él próximamente.



## Manual de PHP.



curl	
cURL Support	enabled
cURL Information	7.19.7
Age	3
Features	
AsynchDNS	no
Debug	no
GSS-Negotiate	no
IDN	no
IPv6	no
Largefile	no
NTLM	no
SPNEGO	no
SSL	no
SSPI	no
krb4	no
libz	no
CharConv	no
Protocols	http, ftp, telnet, dict, imap, imaps, http, file, https, ftps
Host	4848-pc-linux-gnu
cURL Version	OpenSSL 3.0.8
ZLib Version	1.2.3.3

## Instalación de cURL en Windows.



## Debes conocer

Aunque la instalación y utilización de cURL en sistemas Linux/Ubuntu es inmediata, no sucede lo mismo con los sistemas Windows. En Windows la librería cURL se incluye en la instalación estándar de PHP, y no incluye una lista de autoridades de certificación.

Esta lista de autoridades de certificación es imprescindible para establecer conexiones seguras utilizando el protocolo HTTPS. Los certificados de los sitios web seguros deben estar firmados por una autoridad de certificación confiable, y su firma se debe poder comprobar. Si no tenemos una lista de autoridades de certificación confiables, no se podrá realizar esta comprobación y obtendremos un mensaje como:



Para solventar este problema, desde la web de cURL puedes descargar la lista adaptada de autoridades de certificación que incluye Mozilla en su navegador Firefox. Al utilizar la librería cURL desde tu código, tendrás que indicarle que utilice esa lista de autoridades de certificación, añadiendo una línea como la siguiente:



### [Lista de autoridades de certificación.](#)

Si utilizas Windows para los ejemplos de aplicaciones web híbridas de este tema, tendrás que modificar el código de la API de Google para no tener problemas de certificados con cURL.

# Autoevaluación

**La librería cURL debe utilizar una lista válida de autoridades de certificación:**

- Para poder acceder a servidores web utilizando HTTPS.
- Para poder acceder a servidores web utilizando HTTP.

## 4.1.- Google Geocoding.

El servicio de Google Geocoding, que forma parte de los servicios web de Google Maps, se basa en peticiones REST, indicando a continuación el tipo de respuesta que queramos obtener:

### [Google Geocoding](#)

Como parte de la petición, puedes utilizar entre otros los siguientes parámetros:



### Algunos parámetros del servicio Google Geocoding.

Parámetro	Significado
address	Dirección de la que quieras obtener las coordenadas. Es obligatorio, salvo para peticiones inversas.
latlng	Coordenadas a partir de las cuales quieras obtener una dirección. Es obligatorio solo para peticiones inversas.
language	Idioma en el que se devuelven los resultados. Nosotros utilizaremos "es". Es opcional.
sensor	Indica si la solicitud proviene de un dispositivo con sensor de localización. Sus posibles valores son true y false. Es obligatorio.

Para usar el servicio Google Geocoding no necesitas indicar tu ID de desarrollador registrado en Google. Tienes más información sobre el mismo en su página de documentación.

### [Documentación en español de Google Geocoding](#)

Así, por ejemplo, tendríamos las solicitudes siguientes devolviendo información en formato JSON en lugar de XML:

`http://maps.google.com/maps/api/geocode/json?  
address=plaza+de+la+peregrina,pontevedra,spain&language=es&sensor=false`

`http://maps.google.com/maps/api/geocode/json?  
latlng=42.430283,-8.643625&language=es&sensor=false`

La respuesta obtenida a la primera petición es:

1. `location`

2. `address_components`

3. `formated_address` "Plaza de la Encarnación, 16001, Madrid, Spain"

4. `geometry`

5. `location`

6. `lat` 40.416735

7. `lon` -3.702508

8. `place_id` ChIJFjRqDfCtj4gRzXWQYUkVJLc

9. `types` [POI]

10. `name` Plaza de la Encarnación

11. `long_name` Plaza de la Encarnación, Madrid, Madrid, Spain

12. `short_name` Plaza de la Encarnación

En este caso, la información que nos interesa es la que contiene el elemento location. En la petición inversa, para obtener la dirección podremos usar los elementos formated\_address o address\_components, que nos ofrecen la misma información de forma compacta o desglosada respectivamente.

De nuevo puedes recurrir a la documentación del servicio para obtener información sobre cada uno de los elementos que forman la respuesta.

## Autoevaluación

**El servicio Google Geocoding devuelve la información en un formato u otro:**

- Dependiendo de la URL que se utilice en la petición.
- En función de un parámetro GET que se debe utilizar en la petición.

## 4.2.- Aplicación web híbrida de geocodificación.

Vas a crear una aplicación sencilla que utilice el servicio web que acabas de ver de Google. Se trata simplemente de ver cómo se puede utilizar desde PHP. Para ello crearemos un interfaz como el siguiente:

The form is titled "Servicios de geocodificación". It has two main sections. The top section is for coordinates and contains fields for "Latitud" and "Longitud", followed by a button "Ver coordenadas con Google". The bottom section is for address details and contains fields for "Calle", "Ciudad", "País", and "CP". A magnifying glass icon is located at the bottom left of the form area.

El formulario está dividido en dos zonas. En la superior, el usuario podrá introducir unas coordenadas y en la inferior los datos de una dirección. Se trata de utilizar los dos botones del formulario para realizar consultas al servicio de geocodificación, de tal forma que:

- ✓ Al pulsar sobre el botón "Ver coordenadas con Google" se lanza una petición a Google Geocoding y se cubren las coordenadas de la parte superior del formulario con las que se reciben.

Para realizar las llamadas mediante Ajax utilizarás la librería Xajax, vista en la unidad anterior. La función que se encarga de realizar la llamada a Google Geocoding para obtener las coordenadas de una dirección (en formato XML) será:

```
Request = new XajaxRequest();
Request["Search"] = "http://maps.google.com/maps/api/geocode/xml?address=" . $direccion . "&geocode=true&sensor=false";
Request["Send"] = true;
Request["method"] = "xml";
Request["language"] = "es";
Request["xml"] = true;
Request["xml_type"] = "array";
Request["xml_index"] = 0;
```

The code shows a series of function calls to the XajaxRequest object. It starts with 'Request = new XajaxRequest();', then defines a parameter 'Search' with a value that includes a URL to the Google Geocoding API. It then sets 'Request["Send"]' to true, specifies the method as 'xml', sets the language to 'es', and indicates that the response should be an array in XML format ('xml\_type: array'). Finally, it specifies the index of the XML array to process ('xml\_index: 0').

Examina el código completo de la aplicación que se incluye en el siguiente fichero. Asegúrate de ajustar la ruta a la librería Xajax.

 [Código de la aplicación.](#) (2.71 KB)

HTML5 incluye entre sus nuevas características una API de geolocalización, que permite a las aplicaciones web utilizar código JavaScript para obtener las coordenadas en que se encuentra el usuario. La aplicación del código anterior utiliza esa funcionalidad. Para conocer más detalles sobre el funcionamiento de la geolocalización en HTML5 y ver cómo se puede integrar con Google Maps, puedes consultar la siguiente página web.

[Geolocalización con HTML5.](#)

## 4.3.- Google Directions.

Google Directions es un servicio web de Google, que al igual que Google Geocoding también forma parte de Google Maps, y cuya principal utilidad es el cálculo de rutas para llegar desde una ubicación origen a otra ubicación destino. Las rutas pueden incluir además puntos intermedios (hitos), y tanto ellos como el origen o el destino pueden especificarse mediante direcciones reconocibles por el usuario o mediante coordenadas (latitud y longitud).

[Documentación de Google Directions.](#)



[Google Directions respuestas en formato JSON \(recomendado por Google\).](#)

[Google Directions para respuestas en formato XML.](#)

Los únicos parámetros que deben figurar obligatoriamente en una petición son:

- ✓ El punto origen (origin).
- ✓ El punto destino (destination).
- ✓ La indicación de si se utiliza o no un dispositivo de localización (sensor).

Entre los parámetros opcionales están:

- ✓ El idioma en que se devuelven los resultados (language).
- ✓ El medio de transporte para el cálculo de las rutas (mode).
- ✓ Los puntos intermedios o hitos (waypoints). Se deben separar unos de otros utilizando una barra vertical "|". Opcionalmente se puede incluir "optimize:true" como primer argumento, lo que provocará que se reordenen los puntos intermedios para optimizar la ruta. En este caso, en el elemento waypoint\_order de la respuesta se mostrará el orden resultante de optimizar la ruta.

Por ejemplo, una consulta a Google Directions podría tener la siguiente forma:

`http://maps.google.com/maps/api/directions/json?  
origin=42.430283,-8.643625&destination=42.402497,-8.812001&language=es&sensor=false`

La respuesta obtenida en formato JSON mostrará las indicaciones necesarias para llegar desde el origen al destino. El código necesario en PHP para utilizar el servicio será:

// Con una URL como la anterior almacenada en la variable \$url:

```
$json = file_get_contents($url);  
  
$respuesta = json_decode($json);
```

Una vez decodificado el formato JSON obtenido, el acceso a la información se realiza utilizando los elementos del objeto \$respuesta. Revisa la documentación del servicio para ver cómo se estructuran las respuestas. Por ejemplo:

```
$resumen = $respuesta->routes[0]->summary;
```

Se utiliza la función `file_get_contents` de PHP para almacenar en una variable la respuesta obtenida del servicio Google Directions.

#### [Función `file\_get\_contents`.](#)

Cuando crees una aplicación que utilice servicios proporcionados por terceros, deberás tener en cuenta siempre sus **licencias de uso**. Por ejemplo, los resultados obtenidos al usar tanto Google Directions como Google Geocoding han de acompañarse obligatoriamente de los resultados de visualización de un mapa de Google. Además, en algunos casos las respuestas incluyen advertencias e información sobre derechos de autor que deberán mostrarse a los usuarios.

## Autoevaluación

**En una petición al servicio Google Directions, deben figurar obligatoriamente los parámetros:**

- `origin`, `destination` y `sensor`.
- `origin`, `destination`, `mode` y `sensor`.



## Recomendación

Es recomendable consultar los [ejemplos resueltos de la documentación de Google Directions](#).

Y éste [ejemplo para los waypoints](#).



## Para saber más

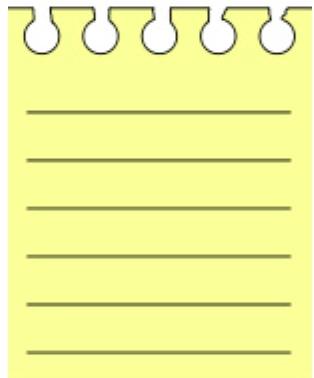
[Gmaps.js](#) es una librería que permite facilitar el uso de las APIs de Google Maps.



## 4.4.- Google Tasks.

Los servicios que has utilizado hasta ahora recibían peticiones por parte del usuario (en nuestro caso una aplicación web) y generaban, y devolvían, respuestas a las mismas en formatos JSON o XML. El servicio Google Tasks necesita, además, una autorización para devolver información privada de un usuario. Para obtener esa autorización, nuestra aplicación deberá usar el protocolo OAuth2.

El servicio de Google Tasks nos permite gestionar las tareas personales del usuario.



### Servicio de Google Tasks.

Básicamente existen dos tipos de recursos:

- ✓ **Listas de tareas.** Una de ellas es la lista de tareas por defecto; existe siempre y no se puede eliminar.
- ✓ **Tareas.** Es cada uno de los elementos que contiene una lista de tareas. Puede contener información como el título de la tarea, notas, o fechas.

Hay dos formas de utilizar el servicio:

- ✓ Mediante llamadas REST directamente, al igual que hicimos cuando utilizamos los servicios anteriores.
- ✓ Mediante una librería cliente, disponible para múltiples lenguajes (entre ellos PHP).

En la documentación del servicio tienes información y ejemplos sobre la utilización del servicio desde cualquiera de estos dos métodos. Para poder utilizar la librería cliente deberás, en primer lugar, descargarla en su versión para PHP.

### Librerías cliente para Google Tasks.

Una vez descomprimida y ubicada en una ruta accesible al servidor web, deberás incluirla en las aplicaciones que la utilicen. Existen un fichero común, y otro específico según el tipo de servicio al que necesites acceder. Por ejemplo, para poder utilizar el servicio Google Tasks tendrás que añadir las siguientes líneas en tu código PHP:

```
require_once 'ruta/a/google/tasks/client/libraries/Google_Client.php';
require_once 'ruta/a/google/tasks/client/libraries/contrib/Google_TasksService.php';
```

Asegúrate de ajustar correctamente la ruta a la librería.

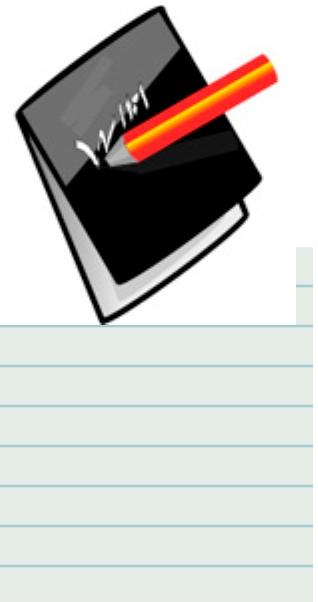
Será necesario crear dos objetos; uno de tipo Google\_Client, que utilizará OAuth2 para gestionar las autorizaciones de acceso a los servicios.



## 4.4.1- Google Tasks (II).

---

El primer paso para acceder al servicio es autenticarse utilizando el método authenticate de la clase Google\_Client. La clave de acceso obtenida se debe almacenar utilizando la función setAccessToken. Puede almacenarse en una variable de sesión para futuras llamadas al servicio.



```
1 // Autentificación
2
3 $client = new Google_Client();
4 $client->setApplicationName('Google Tasks API PHP Client');
5 $client->setAccessType('offline');
6 $client->setScopes('https://www.googleapis.com/auth/tasks.readonly');
7
8 $SESSION['key'] = $client->getAccessToken();
```

Una vez autenticado, puedes emplear el objeto de la clase Google\_TaskService para gestionar las listas de tareas y las tareas del usuario.

- ✓ Para crear una nueva lista de tareas:

```
    $nuevaTarea = new Google_Task();
    $nuevaTarea->listName("Google Tasks");
    $nuevaTarea->listOrder(1);
    $nuevaTarea->insert($nuevaLista);
```

- ✓ Para crear una nueva tarea y agregarla a una lista de tareas:

```
    $nuevaTarea = new Google_Task();
    $nuevaTarea->listName("Google Tasks");
    $nuevaTarea->listOrder(1);
    $nuevaTarea->content("Hacer la tarea");
    $nuevaTarea->insert($nuevaLista);
```

- ✓ Para eliminar una lista de tareas:

```
    $eliminarLista = $nuevaLista->delete();
```

- ✓ Para eliminar una tarea de una lista:

```
    $eliminarTarea = $nuevaTarea->delete();
```

- ✓ Para recorrer las listas de tareas y sus tareas:

```
    // Para recorrer las listas de tareas
    $tareas = $nuevaLista->getTasks();
    foreach ($tareas as $tarea) {
        echo $tarea->content();
    }
    // Recorrer las tareas dentro de la lista
    $tareas = $nuevaTarea->getTasks();
    foreach ($tareas as $tarea) {
        echo $tarea->content();
    }
```

Para que la librería funcione correctamente en sistemas Windows, deberás solucionar el problema de CURL con las autoridades de certificación. Una vez descargado el fichero de autoridades tal y como se mencionó anteriormente, deberás indicar que se utilice. Para ello, en el fichero src/io/apiCurlIO.php, tendrás que modificar la función makeRequest añadiendo la siguiente línea después del conjunto de llamadas a curl\_setopt:

```
curl_setopt($ch, CURLOPT_CAINFO, 'ruta a la lista');
```

# Autoevaluación

**Para utilizar el servicio Google Tasks desde PHP, puedes emplear:**

- La API que ofrece Google.
- La API que ofrece Google o llamadas REST.

## **4.5.- Aplicación web híbrida de gestión de repartos.**

---

Vamos a ver cómo puedes crear una aplicación web híbrida que utilice los servicios que acabas de ver, utilizando como punto de partida la aplicación web con la que has estado trabajando en unidades anteriores. El supuesto del que se parte es el siguiente:

Como la zona de influencia de la tienda aún es pequeña, y pensando también en mantener la relación con los clientes, se ha decidido hacer reparto directo de los productos que se compran en la tienda online. Para ello se ha pensado en crear una aplicación web híbrida con las siguientes características:

- ✓ Se utilizará la API del servicio de tareas de Google (Google Tasks) para almacenar como listas de tareas la información de los repartos. De esta forma la información podrá ser consultada desde cualquier lugar utilizando un dispositivo con conexión a Internet. Cada lista de tareas se corresponde en la aplicación con una lista de reparto, y cada una de sus tareas con un envío. Para diferenciar una lista de otra, se le pone como parte del título la fecha del día en que se hará el reparto.
- ✓ Para cada producto que se reparte se creará una tarea en la lista correspondiente. Esa tarea almacenará la dirección de envío y sus coordenadas. Para obtenerlas, y para mostrar su ubicación en un mapa, en el momento en que se introduzca la dirección se utilizará el servicio de geocodificación de Google (Google Geocoding).
- ✓ Para optimizar la ruta que se ha de recorrer, se utilizará Google Directions. La idea es reorganizar de forma automática el orden de los productos que se van a repartir cada día de forma que se minimice la distancia recorrida.

La apariencia de la aplicación será:



Cuando se cree una nueva tarea (un nuevo envío), se pedirá la dirección y se mostrará una pantalla como la siguiente para que el usuario complete los datos necesarios.

A screenshot of a modal dialog titled "Datos del nuevo envío". It contains fields for "Dirección:", "Latitud:", "Longitud:", and "Título:". Below these fields are two buttons: "Crear nuevo Envío" and "Ver en Google Maps". A magnifying glass icon is located at the bottom left of the dialog.

Se utiliza también Google Maps para mostrar en una nueva ventana el mapa correspondiente a las coordenadas de envío de los productos.

## 4.5.1.- Aplicación web híbrida de gestión de repartos (II).

Para gestionar (crear y eliminar) las listas de tareas y sus tareas, puedes utilizar parámetros GET y recargar la misma página. Si está presente un parámetro 'accion', se realizará un procesamiento determinado. Por ejemplo:

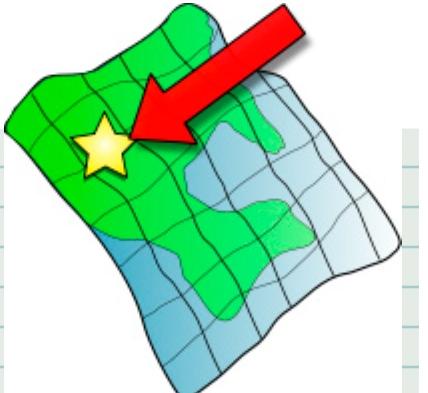
```
switch (q['accion']) {
    case 'nuevo':
        // Crear una nueva lista de reparto
        break;
    case 'listado':
        // Listado de las listas de reparto existentes
        break;
    case 'actualizar':
        // Actualizar una lista de reparto existente
        break;
    default:
        // Crear una nueva lista de reparto
}
```

Para abrir una ventana que muestre ciertas coordenadas en un mapa, puedes utilizar una función Javascript como la siguiente:

```
function abrirMapa(lat, lon) {
    var win = window.open('http://www.google.com/maps?ll=' + lat + ',' + lon + '&zoom=12&zoomtype=fit');
    win.moveTo(100, 100);
}

// Cogemos las coordenadas del diálogo
var latitud = document.getElementById('latitud').value;
var longitud = document.getElementById('longitud').value;

// Creamos la ventana
abrirMapa(latitud, longitud);
```



Para obtener las coordenadas de una dirección concreta, una solución es utilizar Xajax para llamar a Google Geocoding, de forma similar a como ya hiciste en la aplicación web de geocodificación anterior.

## **4.5.2.- Aplicación web híbrida de gestión de repartos (III).**



- ✓ Función ordenarPorPrioridad():
- ✓ Comparación: tienen prioridades más altas que las demás.
- ✓ de las prioridades que debemos ordenar.

ver código en SCSS y Response (funcionamiento):

- return this.ruta;

2. Ordenar por ID:

- ✓ Método de ordenamiento de documentos por ID.
- ✓ Función: ordenarPorID(): ordena los documentos por su ID de mayor a menor.
- ✓ Comparación:

  - ✓ Cognitivo (el ID es base del documento actual).
  - ✓ var id = document.querySelector('#id').value;

- ✓ Anotar el código de la lista de regalos.
- ✓ Ver los documentos ordenados por ID de regalos.
- ✓ Las nuevas posiciones que deben ocupar los envíos.
- ✓ Deben ser reordenados en función de su posición.

●	función ordenarRutas (función de punto intermedia)
●	- ordena los puntos de paso (punto de origen y destino)
●	- Se pide la ruta óptima con los puntos en orden óptimo
●	- Se obtiene el resultado final
●	- La ruta óptima es devuelta al punto de llamada original
●	- Se ordenan los puntos de paso que aparecen
●	- Se ordenan los puntos de paso
●	- Se ordenan los puntos de paso
●	- La ruta óptima es devuelta al punto de llamada original
●	- Se ordenan los puntos de paso
●	- Se ordenan los puntos de paso
●	- Se ordenan los puntos de paso
●	- Y devolvemos los arrays
●	- Se ordenan los puntos de paso
●	- Se ordenan los puntos de paso
●	- Se ordenan los puntos de paso
●	- Se ordenan los puntos de paso

El array obtenido se envía en parámetros GET a la misma página, que lo debe procesar para ordenar las tareas según indica.

## Autoevaluación

**Para obtener una ruta optimizada utilizando el servicio Google Directions, debes utilizar optimize:true en el parámetro waypoints, y al procesar la respuesta recibida:**

- Revisar el orden de los elementos step recibidos.
- Revisar el orden que contiene el elemento waypoint\_order.

## **4.5.3.- Aplicación web híbrida de gestión de repartos (IV).**

 [Aplicación web](#) (518356)

 [Documento explicativo configuración y funcionamiento aplicación](#) (566827)



## 4.6.- OpenStreetMap.

Hemos visto cómo trabajar con mapas de Google que hasta cierto límite de uso son gratuitos, pero para consultas masivas son de pago. En el Anexo también se puede encontrar cómo usar el servicio de Yahoo! PlaceFinder que fue gratuito durante un tiempo hasta que decidieron hacerlo de pago.

Si se necesita un servicio libre, potenciado por la comunidad y de código abierto, también es posible usar mapas en otros formatos, muy valiosos y populares en la actualidad, como son los mapas de OpenStreetMap.

[OpenStreetMap](#) también conocido por sus siglas **OSM**, es uno de los más populares [servicios de mapas online](#) de internet y un proyecto encaminado a la creación, publicación y visualización de mapas para su uso en diferentes plataformas y dispositivos.

Accediendo al sitio, podrás cargar mapas de cualquiera región o zona geográfica.

Es un proyecto de colaboración de código abierto y libre, en el que puede participar y aportar cualquier persona.

La principal característica y lo que distingue a OSM, es que los mapas son creados y editados por los propios miembros, aportando cada uno su conocimiento.

Cualquier persona puede colaborar, siempre que cuente con una conexión a Internet con la que poder subir los mapas editados.

De hecho en la medida que mas personas aporten, mas se enriquecerá el contenido de dichos mapas.



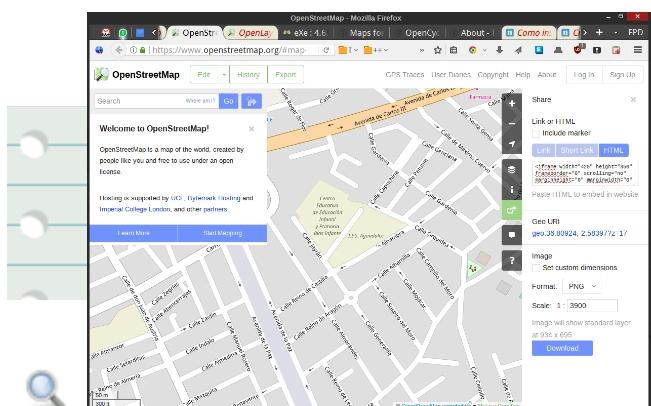
### Reflexiona

¿Por qué usar los mapas de OpenStreetMap? ¿Es posible añadir contenido a los mapas de Google?

[Mostrar retroalimentación](#)

Aunque el proyecto de OpenStreetMap no ofrece directamente una API en JavaScript (sí a través de terceros como veremos más adelante) para facilitar que los mapas sean mostrados y/o editados con cierto control en las páginas de sitios web, existe soporte oficial para una API basada en REST, [más info en su Wiki](#) y existe un [proyecto para trabajar de una forma más intuitiva con la API](#).

También se puede compartir un mapa de OSM para ser visualizado en otra web a través de un **iframe**. Aquí se puede ver una captura de cómo compartir un mapa desde OpenStreetMap.org

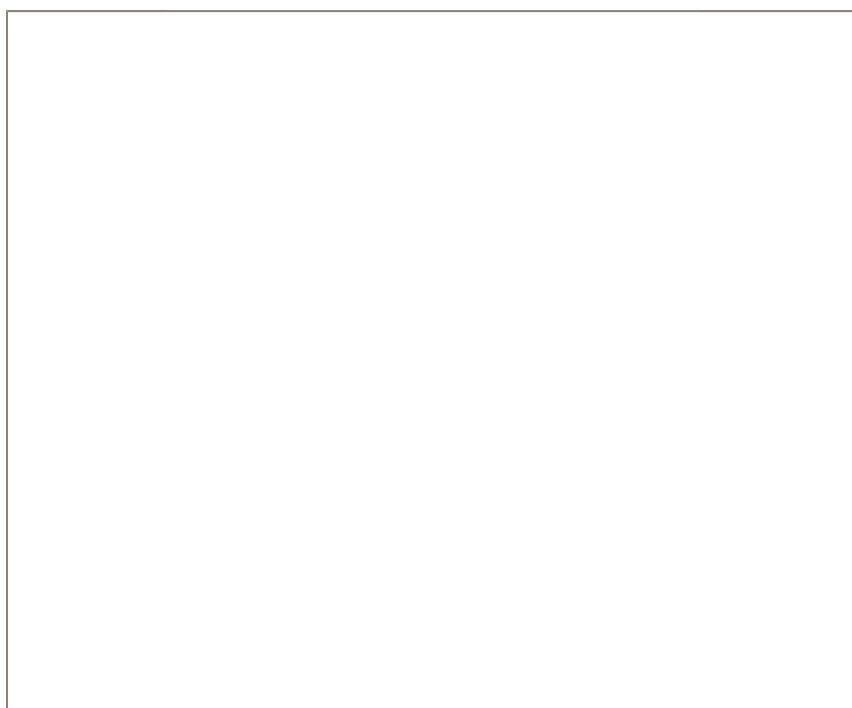


Elaboración propia

El código HTML para el iframe que muestre ese mapa sería:

```
<iframe width="425" height="350" frameborder="0" scrolling="no" src="https://www.openstreetmap.org/embed.html?lat=39.9924&lon=-2.3939&z=17">
```

Con este resultado al incluir ése código en una web (Prueba a interaccionar con el ratón arrastrando y haciendo zoom con la rueda):



[View Larger Map](#)

Sin embargo, ofrecer APIs de JavaScript para trabajar con mapas OSM es asumido por varios servicios independientes, la gran mayoría libres.

Vamos a detallar como usar los principales de estos servicios, para mostrar cualquier mapa en las páginas de un sitio web, de forma sencilla. Por supuesto obviamos los servicios de pago.

No es necesario saber de JavaScript, solo copiar y pegar y tener las nociones elementales de lo que es latitud, longitud y el nivel de zoom.

Los principales servicios son los siguientes:

- ✓ OpenLayers.
- ✓ Leaflet.
- ✓ Google Map. Otra de las opciones es usar la API 3 de Google Map, insertando los mosaicos de OSM como una capa. La ventaja de este método es aprovecharse de los recursos de Google, aunque los mapas insertados no poseen las opciones que obtenemos usando los recursos anteriores.

Existen otros servicios, pero solo relacionamos los mas empleados en internet y factibles. Todos los mapas generados usando los servicios anteriores, son adaptables y se adaptan al tamaño de la pantalla de cualquier dispositivo.



## Para saber más

En [ésta página](#) de la Wiki de OSM se detallan otros servicios y formas de insertar los mapas en webs y sistemas de gestión de contenidos como Wordpress, Joomla, Drupal, etc.

Los mapas de OSM pueden ser descargados libremente por zonas para su uso offline. [Más info aquí](#). Estos mapas son usados por programas de navegación offline libres para móviles como [OsmAnd](#).

**Level0** ([level0.osmz.ru](http://level0.osmz.ru)) es un editor de OpenStreetMap basado en web, diseñado para ser de bajo nivel y fácil de usar. No requiere soporte de JavaScript y, por tanto, puede utilizarse en dispositivos móviles y en navegadores basados en texto. [Más info aquí](#)

## **4.6.1.- Mostrar mapas de OpenStreetMap con OpenLayers.**

---

**OpenLayers** es una **librería JavaScript gratis y de código abierto**, para posibilitar integrar los mapas interactivos de OpenStreetMaps en los sitios web.

La librería es muy fácil de usar, con varias opciones.

Desde el sitio del proyecto se cargan los archivos necesarios, aunque los que poseen un servidor pueden alojarlos localmente para mejorar el rendimiento.

Es la librería que se recomienda a los aficionados por su facilidad de uso y por la gran cantidad de ejemplos prácticos en su documentación.

Para generar un mapa simple de OpenStreetMap de cualquier región del mundo o de una ciudad, como el que se muestra a continuación, solo basta con insertar una porción de código JavaScript en un archivo HTML.

Puede ser una página de un servidor de internet o un archivo para usarlo de forma local en la computadora, cargándolo con el navegador web.

Descarga usando [éste enlace](#) un archivo HTML con el código anterior, listo para usar offline.

En el código de ejemplo anterior, solo es necesario sustituir las coordenadas por otras en formato decimal.

Primero especifica la longitud y después la latitud, separadas por una coma.

En OpenLayers de forma predeterminada se usa la proyección **EPSG:3857**, que usa como unidad de medida **metros**.

En este caso se ha especificado la proyección **EPSG:4326**, para poder usar los **grados decimales**.

No obstante los que necesiten obtener una localización usando las coordenadas EPSG:3857, pueden usar la siguiente herramienta: [Convertir grados coordenadas decimales a Google Mercator](#)

Los que necesiten convertir de grados geográficos (grados, minutos, segundos) al sistema decimal, pueden hacerlo con otra herramienta: [Convertir grados geográficos, minutos y segundos a decimales](#)

## Agregarle elementos al mapa de OSM

La librería de OpenLayers ofrece gran cantidad de opciones, las que se pueden emplear para agregar elementos como controles, marcadores, imágenes, globos con información, escalas, etc.

Con la librería de OpenLayers también se pueden mostrar e insertar mapas de los servicios Bing, MapBox, Here Maps, ESRI, WMS y otros.

Además mapas temáticos como de zonas horarias, terremotos, nivel del mar, etc.

Los que estén interesados en crear mapas más profesionales e interactivos, pueden consultar la documentación incluida en la librería, que se puede descargar en un archivo ZIP, desde <https://openlayers.org/>

## ¿Cómo usar el código en un mapa offline?

Copia y pega el código anterior en un archivo de texto y dale cualquier nombre, pero con la extensión HTML.

Cambia el valor del "zoom" y especifica en "center" las coordenadas donde se centrará el mapa.

Guarda los cambios y da dos clics en el archivo HTML para abrirlo con el navegador predeterminado y cargar los elementos del mapa.

## Mejorar el rendimiento del mapa en un sitio web

El código usa la versión 4, la más reciente del proyecto y carga el archivo de estilo CSS y el JS desde el sitio de OpenLayer.

Para mejorar la carga de ellos y prevenir el bloqueo de la página ante cualquier irrupción del servicio, se pueden descargar los dos archivos anteriores incluidos en la librería y subirlos a nuestro servidor.

Debemos de aclarar que con este método las miniaturas (tiles) del mapa se cargan directamente de los servidores de OpenStreetMap.

Este servicio no ofrece garantía sobre la disponibilidad y el rendimiento no es el mejor.

Si se necesita generar mapas con mayor rendimiento, se pueden evaluar otras opciones para cargar las miniaturas desde otros servicios o generarlas dinámicamente.

**El servicio de OpenLayers es libre y de código abierto.**

**Para usar la librería no es necesario registrarse, ni usar ninguna API o clave.**

## 4.6.2.- Mostrar mapas de OpenStreetMap con Leaflet.

Existen otras alternativas, pero Leaflet es el preferido y más empleado por los desarrolladores en la mayoría de los sitios, debido a la gran cantidad de opciones que facilita.

Leaflet es una librería de JavaScript, gratis y de código abierto.

Es ligera, fácil de usar y los mapas generados son adaptables.

### Como generar un mapa de OSM usando Leaflet

Con Leaflet podemos generar y mostrar en una página uno o varios mapas de cualquier área geográfica.

Solo basta con insertar una pequeña porción de código Javascript para conseguir un mapa básico incluyendo un marcador (pop-up) con la entrada del IES Aguadulce, además de los controles del zoom (aunque se puede usar la rueda del mouse para ampliar y reducir) y los créditos de Leaflet y OSM. El mapa es adaptable, o sea se ajusta automáticamente al tamaño de la pantalla de los dispositivos táctiles también.



### Código para crear un mapa de OSM con Leaflet

Para insertar un mapa de cualquier área geográfica usando el servicio de OpenStreetMap y la librería de Leaflet, haz lo siguiente:

"leaflet.css".

2- Carga el archivo JS "leaflet.js".

3- Crea un contenedor DIV en el lugar exacto de la página donde se desea mostrar el mapa.

4- Inserta el script que carga las miniaturas del mapa, establece donde centrarlo, el nivel de zoom y los elementos que se desean agregarle.

A continuación se muestra el código para crear un archivo HTML, que incluso se puede ejecutar desde la computadora, para cargar un mapa básico, como el que se muestra más arriba.

Otra opción es descargar un archivo HTML con el código anterior, usando [éste enlace](#).

Para convertir a este formato del sistema tradicional de grados usado en mapas analógicos, se puede emplear la siguiente herramienta: [Convertir grados geográficos, minutos y segundos a decimales](#)

Los que se animen a usar este método, pueden descargar toda la documentación disponible en [Github](#)

## Mejorar la carga del mapa alojando Leaflet

El código que se muestra, así como el archivo HTML para descargar, cargan los archivos de estilo CSS y JavaScript desde el CDN de Leaflet.

Los archivos pueden alojarse en su propio servicio de hosting, para asegurarse que siempre estén disponibles y así evitar que pueda producirse un bloqueo en la página, en caso de cualquier interrupción.

La última versión de la librería de Leaflet se puede descargar en un archivo ZIP desde: <http://leafletjs.com/>

El archivo contiene el CSS (leaflet.css) JS (leaflet.js) y una carpeta con imágenes de los elementos que se representan en el mapa.

**El servicio de Leaflet es libre y de código abierto.**

**Para usar la librería no es necesario registrarse, ni usar ninguna API o clave.**

## Autoevaluación

OpenStreetMap.org ofrece APIs en JavaScript para trabajar directamente con sus mapas desde páginas webs.

- Verdadero  Falso



## Para saber más

Una muestra de la potencia de las librerías de Leaflet es el servicio [facilmap.org](http://facilmap.org) desarrollado con estas librerías y que permite la edición colaborativa de marcadores y rutas además de un montón de características interesantes.

En [su Github](#) se pueden ver todas las características avanzadas que ofrece, además de código libre.

## 4.6.3.- Mostrar mapas de OpenStreetMap con Google Maps

---

Uno de los métodos más sencillos de insertar y generar en una página web un mapa de OpenStreetMap, es usando la API de Google Map y agregando como un layer (capa) las miniaturas (tiles) de OSM.

Usando la API 3 del servicio de Google, no es necesario ninguna clave, solo insertar el código necesario en JavaScript.

Este es un método empleado en muchos sitios web, porque el mapa carga bastante rápido.

Un mapa completamente funcional que usa este método se puede probar en la siguiente página de este sitio: [Mapa mixto de Google Maps y OpenStreetMaps](#)

A continuación compartimos el código necesario para crear un mapa que usa este método y ofrecemos un enlace para descargar un archivo HTML de ejemplo.

Mediante [éste enlace](#) se puede descargar un archivo HTML de ejemplo con el código anterior, que puede ser ejecutado para cargar el mapa.



Elaboración propia



## Para saber más

- ✓ [Como editar los mapas de OpenStreetMap y agregar datos e información](#)
- ✓ [Códigos para detectar y mostrar la localización geográfica en las páginas web](#)
- ✓ [Códigos para crear y mostrar mapas de Google Maps](#)
- ✓ [Los mejores servicios de mapas de internet](#)
- ✓ [Convertir grados geográficos, minutos y segundos a decimales](#)
- ✓ [Marble, un atlas o globo del mundo con mapas libres](#)
- ✓ [Como guardar y restaurar la caché de Google Earth](#)

## Anexo.1.- Yahoo! PlaceFinder.

---

Desde 2013, los servicios de Yahoo, que antes eran gratuitos, pasan a ser de pago, por lo que estos Anexos referentes a Yahoo! PlaceFinder son únicamente a nivel informativo (por si alguien quiere probarlos por su cuenta). Pero **estos anexos no entran dentro de la materia que va a ser evaluada.**

Existen muchas fuentes de datos disponibles en Internet que puedes utilizar para construir una aplicación web híbrida. En esta unidad vamos a centrarnos en la información accesible a través de servicios web, concretamente en los que ofrecen las empresas Google y Yahoo!.



Comenzaremos creando una aplicación web híbrida sencilla que haga uso de los servicios de geocodificación que ofrecen ambos. El de Yahoo! se llama PlaceFinder. Puedes encontrar toda la información necesaria sobre su utilización en la web de desarrollo de Yahoo!.

### [Yahoo! PlaceFinder.](#)

El servicio **Yahoo! PlaceFinder** se basa en REST y los datos relativos a la petición se envían mediante parámetros GET. Es necesario indicar como mínimo un parámetro de localización.

La forma más sencilla de indicar una localización es utilizando el parámetro location (o su equivalente q).

Existe otro tipo de parámetros, los de control, que permiten indicar otra información no directamente relacionada con la localización. Es obligatorio el parámetro appid para indicar el ID de tu aplicación web. Otros parámetros de control son:

## Algunos parámetros de control del servicio de geocodificación Yahoo! PlaceFinder.

Parámetro de control	Significado
appid	ID de la aplicación que utiliza el servicio. Es obligatorio.
locale	Código del lenguaje y país. Por defecto "en_US". En nuestro caso deberíamos utilizar "es_ES".
count	Número máximo de respuestas que se devolverán. Por defecto 10.
flags	Cadena de caracteres que especifica qué datos se obtendrán. Por ejemplo: J – Indica que la información se devuelva en formato JSON (por defecto se utiliza XML). P – Indica que la información se devuelva en formato PHP serializado. G – Devuelve información global, no específica de los Estados Unidos.
gflags	Cadena de caracteres que especifica cómo se realizará la búsqueda. Por ejemplo: L – Buscar sólo en el país que se indica. R – Realizar una búsqueda inversa: obtener la dirección a partir de una latitud y longitud.

En la documentación del servicio tienes una lista completa de los parámetros de localización y de control que puedes emplear.

### [Parámetros de localización.](#)

Por ejemplo, una petición simple podría tener la siguiente forma:

`http://where.yahooapis.com/geocode?  
location=Plaza+de+la+Peregrina,Pontevedra,Spain&locale=es_ES&flags=G&count=1&appid=tuID`

Recuerda incluir en las peticiones a los servicios de Google y Yahoo!, tus propios identificadores (los que te han asignado al registrarte) allí donde sea necesario.

## Anexo 1.1- Yahoo! PlaceFinder (II).

Las respuestas obtenidas al utilizar el servicio, contienen los siguientes elementos (indiferentemente de si se indica XML, JSON o PHP serializado):

### Elementos principales de respuesta del servicio de geocodificación Yahoo! PlaceFinder.



Elemento	Significado
ResultSet	Elemento de nivel superior que contiene al resto.
Error	Código de error (0 si no se ha producido ningún error). En la documentación del servicio se incluye una tabla con los posibles códigos de error y su significado.
ErrorMessage	Mensaje de error. "No error" o "Sin errores" si no se ha producido ninguno.
Locale	Código del lenguaje utilizado en la respuesta.
Quality	Estimación de la calidad de la respuesta obtenida.
Found	Número de resultado obtenidos.
Result	Cada uno de los resultados obtenidos.

En la documentación del servicio tienes toda la información sobre los elementos que conforman los resultados obtenidos.

[Formato de las respuestas de PlaceFinder.](#)

Por ejemplo, como respuesta a la petición anterior podíamos obtener el siguiente documento en formato XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<Result>
    <Location>
        <Lat>42.430283</Lat>
        <Lon>-8.643625</Lon>
        <Label>Avda. Alfonso XIII, 100, Madrid</Label>
        <Address>Avda. Alfonso XIII, 100, Madrid, 28001, Madrid, Spain</Address>
        <Quality>HighQuality</Quality>
        <Bounds>42.429380, -8.644525, 42.431184, -8.642725</Bounds>
    </Location>
</Result>
</Response>
```

Si quisieras efectuar una consulta inversa (obtener una dirección a partir de unas coordenadas de latitud y longitud), podrías realizar una consulta utilizando gflags=R, como:

```
http://where.yahooapis.com/geocode?
location=42.430283,-8.643625&flags=G&locale=es_ES&gflags=R&appid=tulD
```

Y obtendrías:

```
<?xml version="1.0" encoding="UTF-8"?>
<Result>
    <Location>
        <Label>Avda. Alfonso XIII, 100, Madrid</Label>
        <Address>Avda. Alfonso XIII, 100, Madrid, 28001, Madrid, Spain</Address>
        <Quality>HighQuality</Quality>
        <Bounds>42.429380, -8.644525, 42.431184, -8.642725</Bounds>
    </Location>
    <LatLong>
        <lat>42.430283</lat>
        <lon>-8.643625</lon>
    </LatLong>
    <Time>
        <Timezone>Europe/Madrid</Timezone>
        <TimezoneOffset>36000</TimezoneOffset>
        <TimezoneName>Poniente Madrid</TimezoneName>
        <TimezoneOffsetName>CEST</TimezoneOffsetName>
    </Time>
</Result>
</Response>
```

# Autoevaluación

**Relaciona las palabras con su significado relativo al servicio Yahoo! PlaceFinder:**

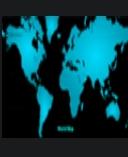
## Ejercicio de relacionar.

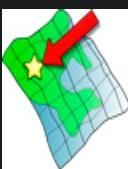
Palabra	Relación	Significado
appid	0	1. Número de resultados que incluye la respuesta.
Found	0	2. Parámetro que se utiliza en una petición para indicar el número máximo de respuestas que se devolverán.
count	0	3. Parámetro que se utiliza en una petición para indicar el identificador de la aplicación.
Quality	0	4. Estimación de la calidad de la respuesta obtenida.

**Enviar**

# Anexo.- Licencias de recursos.

## Licencias de recursos utilizados en la Unidad de Trabajo.

Recurso (1)	Datos del recurso (1)	Recurso (2)	Datos del recurso (2)
	Autoría: Stockbyte. Licencia: Uso educativo no comercial para plataformas públicas de Formación Profesional a distancia. Procedencia: CD-DVD Num. V43.		Autoría: Stockbyte. Licencia: Uso educativo no comercial para plataformas públicas de Formación Profesional a distancia. Procedencia: CD-DVD Num. V43.
	Autoría: Stockbyte. Licencia: Uso educativo no comercial para plataformas públicas de Formación Profesional a distancia. Procedencia: CD-DVD Num. V43.		Autoría: v1loes. Licencia: Creative Commons BY. Procedencia: YouTube.
	Autoría: Elaboración propia. Licencia: Uso educativo no comercial. Procedencia: Captura de una página web generada por Google.		Autoría: Elaboración propia. Licencia: Uso educativo no comercial. Procedencia: Captura de una página web generada por Google.
	Autoría: v1loes. Licencia: Creative Commons BY. Procedencia: YouTube.		Autoría: David Schoen / lyte. Licencia: Dominio público. Procedencia: <a href="http://openclipart.org/detail/1000/user-by-lyte">http://openclipart.org/detail/1000/user-by-lyte</a> .
	Autoría: Stockbyte. Licencia: Uso educativo no comercial para plataformas públicas de Formación Profesional a distancia. Procedencia: CD-DVD Num. V43.		Autoría: Ubuntu. Licencia: GPL <a href="http://www.gnu.org/licenses/gpl.html">http://www.gnu.org/licenses/gpl.html</a> Procedencia: Captura de pantalla Ubuntu.
	Autoría: PHP. Licencia: PHP License 3.01 <a href="http://www.php.net/license/3_01.txt">http://www.php.net/license/3_01.txt</a> . Procedencia: Captura de una página web generada por PHP.		Autoría: Gustavo Rezende / Licencia: Dominio público. Procedencia: <a href="http://openclipart.org/detail/1000/map-2-by-gustavorezende">http://openclipart.org/detail/1000/map-2-by-gustavorezende</a> .

	<p>Autoría: Anónimo.      Licencia: Dominio público.      Procedencia:  <a href="http://openclipart.org/detail/24268/by--24268">http://openclipart.org/detail/24268/by--24268</a>.</p>		<p>Autoría: jeronimo / jeronimo.      Licencia: Dominio público.      Procedencia:  <a href="http://openclipart.org/detail/by-jeronimo">http://openclipart.org/detail/by-jeronimo</a>.</p>
	<p>Autoría: shokunin / shokunin.      Licencia: Dominio público.      Procedencia:  <a href="http://openclipart.org/detail/76297/gps-navigation-by-shokunin">http://openclipart.org/detail/76297/gps-navigation-by-shokunin</a>.</p>		<p>Autoría: agone.      Licencia: Dominio público.      Procedencia:  <a href="http://openclipart.org/detail/agone">http://openclipart.org/detail/agone</a>.</p>
	<p>Autoría: Ausis.      Licencia: Dominio público.      Procedencia:  <a href="http://openclipart.org/detail/1697/pencil-and-note-pad-by-ausis">http://openclipart.org/detail/1697/pencil-and-note-pad-by-ausis</a>.</p>		<p>Autoría: midkiffaries.      Licencia: Dominio público.      Procedencia:  <a href="http://www.openclipart.org/c">http://www.openclipart.org/c</a></p>
	<p>Autoría: krypt / krypt.      Licencia: Dominio público.      Procedencia:  <a href="http://openclipart.org/detail/3325/ruffled-map-by-midkiffaries">http://openclipart.org/detail/3325/ruffled-map-by-midkiffaries</a>.</p>		<p>Autoría: Studio-Hades / stud      Licencia: Dominio público.      Procedencia:  <a href="http://openclipart.org/detail/glyph-dingbat-by-studio_had">http://openclipart.org/detail/glyph-dingbat-by-studio_had</a></p>

# Condiciones y términos de uso de los materiales

Materiales desarrollados inicialmente por el Ministerio de Educación, Cultura y Deporte y actualizados por el profesorado de la Junta de Andalucía bajo licencia Creative Commons BY-NC-SA.



Antes de cualquier uso leer detenidamente el siguiente [Aviso legal](#)

## Histórico de actualizaciones

Versión: 02.00.03

Fecha subida: 22/03/18

Autoría: Jesús Manuel Marín Navarro

**Ubicación:** 3.1

**Mejora (tipo 1):** arreglado el enlace al vídeo para que se vea bien desde la plataforma según las nuevas indicaciones

### Ajustes realizados en la versión

Silvia Castillo Miranda

**Mejora:** Eliminación del idevice de licencia

Versión: 02.00.02

Fecha subida: 19/06/17

Autoría: Jesús Manuel Marín Navarro

**Ubicación:** 3 , 3.1

**Mejora (tipo 1):** Eliminados vídeos por no tener licencia libre y puesto otras explicaciones con capturas de pantalla

Versión: 02.00.01

Fecha subida: 08/06/17

Autoría: Jesús Manuel Marín Navarro

**Ubicación:** 3 , 3.1, mapa conceptual

**Mejora (tipo 1):** descargados y añadidos los vídeos al elp, además de poner enlace al vídeo de Youtube. mapa conceptual con todos los ficheros html, png, etc.

**Versión: 02.00.00**

**Fecha subida: 29/05/17**

**Autoría: Jesús Manuel Marín  
Navarro**

**Ubicación:** 3, 3.1, 4.3, 4.6

**Mejora (tipo 1):** Cambiados los vídeos para adaptarse al nuevo interfaz del portal de desarrollo de Google en 3 y 3.1. Añadido un para saber más con más info sobre la política de precios de Google Map. Añadida la libreria gmaps.js en para saber más del 4.3. Añadida la sección 4.6 con OpenStreetMap

**Ubicación:** Completo

**Mejora (tipo 3):** Las APIs de Google son ahora de pago, hay que volver a redactar dichos apartados y buscar otras alternativas libres.

**Versión: 01.02.01**

**Fecha subida: 14/04/15**

**Autoría: Gloria Ortiz**

**Ubicación:** Primera página

**Mejora (tipo 1):** Materiales --> Se ha puesto la licencia correctamente tal y como establece el procedimiento de Actualización de Materiales (F14/04/15)

**Versión: 01.02.00**

**Fecha subida: 14/04/15**

**Autoría: Carlos Ríos Ruiz**

**Ubicación:** Sección 4.5.3

**Mejora (tipo 2):** Añadido recurso fichero pdf explicando detalladamente la forma de configurar todo lo necesario para la ejecución de la aplicación web ejemplo, así como su funcionamiento.

**Ubicación:** Sección 4.5.3

**Mejora (tipo 2):** Modificación de la aplicación web de ejemplo Repartos para adaptarla a los cambios de las APIs de Google.

**Versión: 01.01.02**

**Fecha subida: 09/04/15**

**Autoría: Carlos Ríos Ruiz**

**Ubicación:** Sección 4.4

**Mejora (tipo 1):** Arreglar enlace Google Tasks.

**Ubicación:** Sección 4.3

**Mejora (tipo 1):** Arreglar los 3 enlaces que aparecen de Google Directions.

**Ubicación:** Sección 4.1

**Mejora (tipo 1):** Arreglado enlace Google Geocoding.

**Ubicación:** Sección 1

**Mejora (tipo 1):** Cambio la entrada de glosario métodos HTTP por una definición.

**Versión: 01.01.01**

**Fecha subida: 03/04/15**

**Autoría: Manuel Alberto Domínguez  
Vega**

**Mejora:** Pequeñas erratas, enlaces rotos, glosario en nuevo formato, credenciales en imágenes y videos (S)

**Versión: 01.01.00**

**Fecha subida: 20/04/14**

**Autoría: Sonia Amate Garrido**

Adaptación de los contenidos a las nuevas librerías de google.

**Versión: 01.00.00**

**Fecha subida: 20/04/14**

**Autoría: MECD MECD**

**Mejora:** *No especificada.*