



Toma lo que necesites,
deja lo que puedas.

Google™ APIs

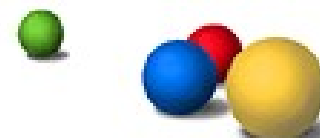
Trucos, tips y consejos para desarrollar aplicaciones Web sin morir en el intento

Por: Mauricio Angulo S.



LAMP

y desarrollo
en plataformas
libres





Toma lo que necesites,
deja lo que puedas.

Puntos de la presentación

A lo largo de la presentación cubriremos los siguientes puntos:

- Qué es una **API** y como funciona
- Cuáles **APIs** ofrece Google a los webmasters
- **Google Search** (servicio de búsqueda Web)
- **Google AdWords** (servicio de publicidad en Internet)
- **Google Sitemaps** (servicio de actualización al buscador)
- **Google Maps** (servicio de mapas satelitales)
- Sobre **Webmasters México**

Tiempo aproximado: **120 minutos**





Toma lo que necesites,
deja lo que puedas.

Qué es un API

API son las siglas de **Application Programming Interface** (Interface de programación de **aplicaciones**), y es por lo general una librería de funciones, objetos y/o aplicaciones que se utilizan para tener acceso a las funcionalidades de otros sistemas o aplicaciones, o para poder hacer intercambio de datos entre ellas.

Una **API** crea una capa de abstracción extra que facilita el desarrollo y reduce el tiempo de creación de un proyecto.

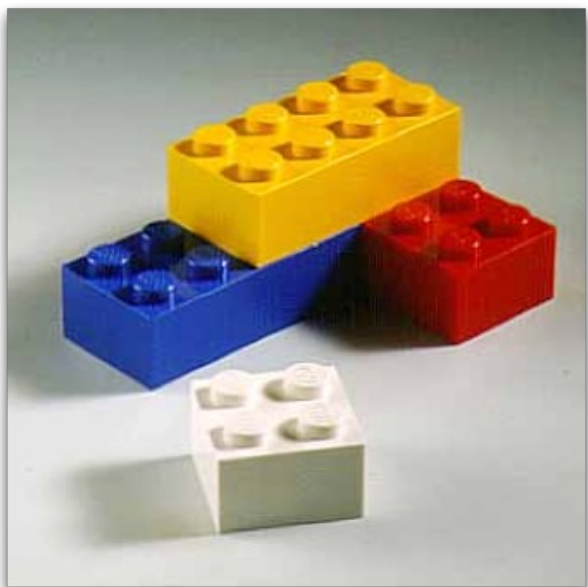




Toma lo que necesites,
deja lo que puedas.

Qué es un API

El desarrollo con **APIs** es como trabajar con **Legos**:

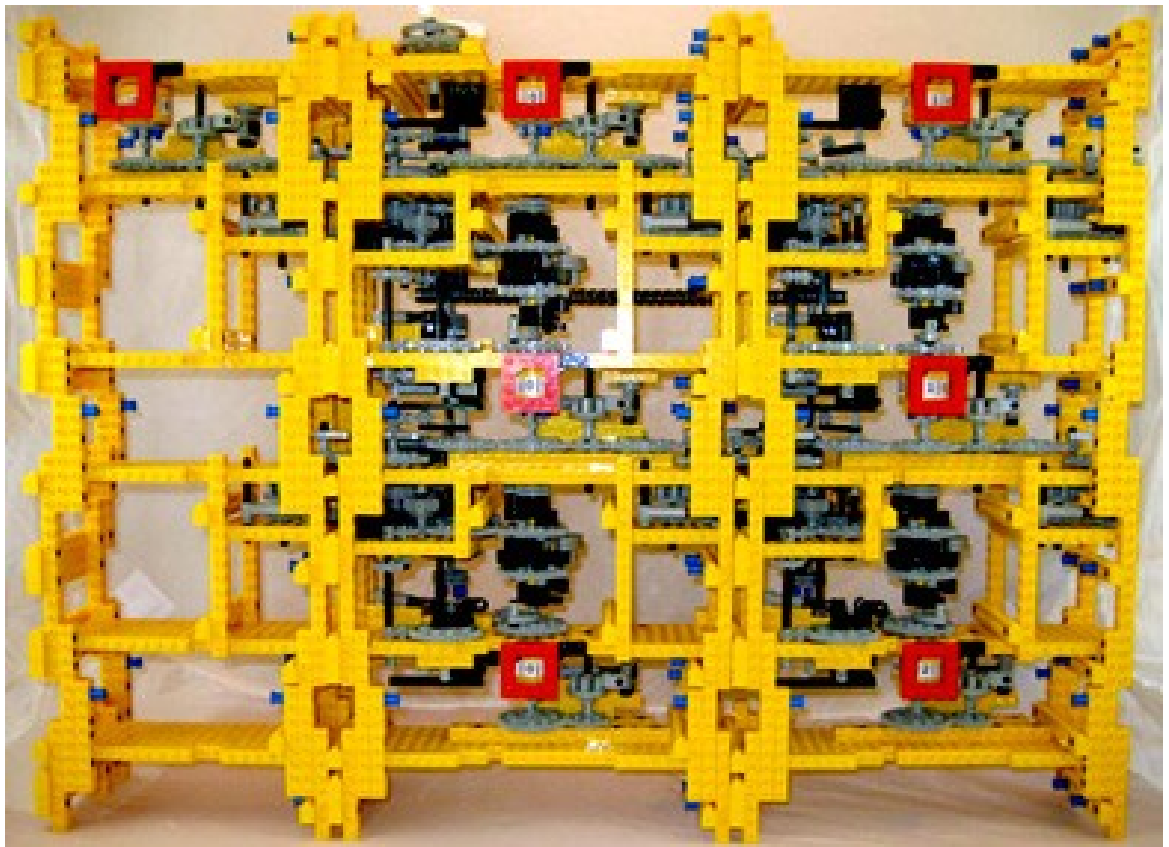




Toma lo que necesites,
deja lo que puedas.

Qué es un API

Entre más piezas tengamos, podremos hacer cosas más complejas y detalladas con menos esfuerzo.





Toma lo que necesites,
deja lo que puedas.

APIs de Google

Google ofrece una gran variedad de APIs para uso de programadores de aplicaciones y webmasters:



Búsqueda web



Mapas satelitales



Noticias y eventos



Compras en línea



Búsqueda en escritorio



Gmail – correo electrónico



Blogger – publicación web

¡... y muchos más!

Google Reader, Google Earth,





Toma lo que necesites,
deja lo que puedas.

APIs de Google

Las diferentes **APIs** de **Google** tienen diferentes niveles de funcionalidad y complejidad, y en su mayoría son de acceso independiente de plataforma y no tiene ningún costo utilizarlas.

Para llevar un seguimiento al día sobre las implementaciones, ejemplos y novedades alrededor de estos desarrollos, vale la pena revisar:

- **Google Code:** <http://code.google.com/apis.html>
- Los sitios y *blogs* de cada servicio en **Google**
- Grupos de desarrollo de **Google**





Toma lo que necesites,
deja lo que puedas.

Google Search

La principal funcionalidad de **Google** es la de **búsqueda Web**. A través de la **API** de búsqueda es posible adquirir datos del buscador:

- en diferentes idiomas
- con filtros para quitar resultados repetidos o no aptos para menores
- localizados por dominios o en sitios específicos
- de sitios que ligan a un sitio o página (backlinks)
- en el caché del buscador
- en rangos de búsquedas

... y muchas, muchas más cosas. En serio. **Muchas**





Toma lo que necesites,
deja lo que puedas.

Google Search

Entre las cosas que podemos hacer con esta **API** están:

- Crear un buscador personalizado para un sitio web o intranet
- Crear una aplicación para dispositivos móviles
- Desarrollar aplicaciones de escritorio con conectividad web
- Utilizar el buscador para filtrar y analizar datos
- Añadir corrector ortográfico a una aplicación

¡El límite es la imaginación!





Toma lo que necesites,
deja lo que puedas.

Google Search

Para tener acceso al **API** del buscador debemos seguir tres pasos:

1. Regístrate para crear una **Cuenta en Google**. El registro es gratuito.
2. **Obtener la llave de acceso para usar el sistema**. Esta llave nos da acceso a 1,000 consultas diarias.
3. **¡Escribir la aplicación!** Aquí ayuda mucho conocer la documentación sobre cómo escribir consultas avanzadas, cómo delimitar los rangos de búsqueda, etc.





Toma lo que necesites,
deja lo que puedas.

Google Search

La página de resultados de **Google** se ve así:

The screenshot shows the Google search interface with the following elements labeled:

- A**: Navigation links (Web, Images, Groups, News, Froogle, Local, more »)
- B**: Search button
- C**: Advanced Search link
- D**: Search input field containing 'google address mountain view'
- E**: Preferences link
- F**: Search results header: 'Web Results 1 - 10 of about 87,500 for google address +mountain view. (0.31 seconds)'
- G**: Tip: 'Find maps by searching for a street address with city or zip code'
- H**: News results for 'google', including links like 'Google Slices and Dices by Locality' and 'Google Rolls Out System To Improve Local Search Results'
- I**: Google logo
- J**: Footer links: 'Advertise with Us - Business Solutions - Services & Tools - Jobs, Press, & Help ©2004 Google - Searching 4,285,199,774 web pages.'
- K**: 'http://www.google.com/'
- L**: '3k'
- M**: 'Cached'
- N**: 'Similar pages'
- O**: 'Google Corporate Information: Google Offices'
- P**: '[More results from www.google.com]'



Toma lo que necesites,
deja lo que puedas.

Google Search

El **API** del buscador web está diseñado para utilizarse con **servicio web**. En el caso particular de **aplicaciones Web** en **LAMP** lo que necesitamos es

SOAP



*"With enough SOAP you can blow anything" -
Tyler*





Toma lo que necesites,
deja lo que puedas.

Google Search

SOAP (siglas de *Simple Object Access Protocol*) es un protocolo estándar actualmente bajo el auspicio de la **W3C** que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. **SOAP** es uno de los protocolos utilizados en los Servicios Web.

SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples pilas de protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes **SOAP** interactuando entre sí.

```
<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  ...
  Message information goes here
  ...
</soap:Envelope>
```

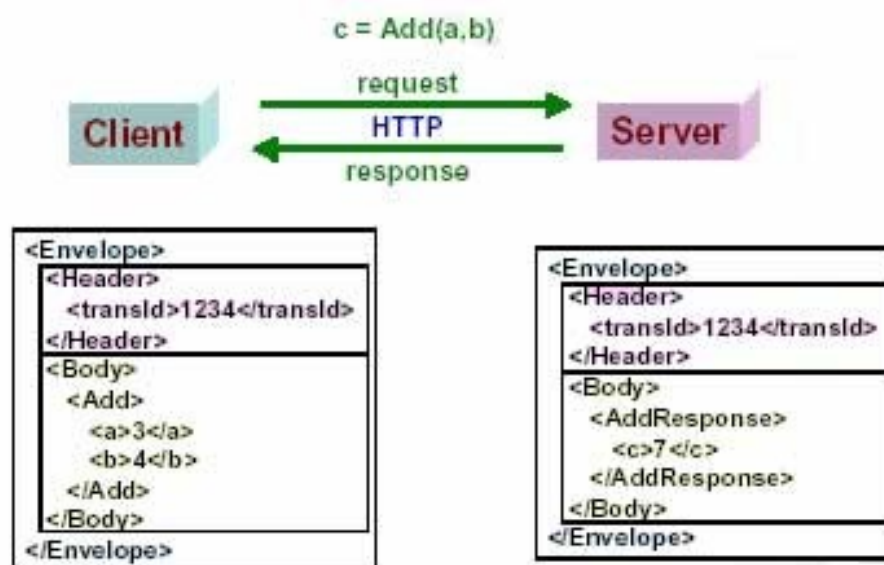


Toma lo que necesites,
deja lo que puedas.

Google Search

SOAP corre sobre cualquier protocolo de Internet, generalmente **HTTP**, que es el único homologado por el **W3C**. **SOAP** tiene como base **XML**, con un diseño que cumple el patrón **Cabecera-Desarrollo de diseño de software**, como otros muchos diseños, v.g. **HTML**.

La cabecera *Header* es opcional y contiene metadatos sobre enrutamiento (*routing*), seguridad o transacciones. El desarrollo *Body* contiene la información principal, que se conoce como carga útil (*payload*). La carga útil se acoge a un *XML Schema* propio..



SOAP



Toma lo que necesites,
deja lo que puedas.

Google Search

Esta es una solicitud para el servicio de búsqueda de Google:

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:si="http://soapinterop.org/xsd"
  xmlns:galactivism="urn:GoogleSearch">
  <SOAP-ENV:Body>
    <galactivism:doGoogleSearch>
      <key xsi:type="xsd:string">your-google-license-key-xxxxxxx</key>
      <q xsi:type="xsd:string">melonfire</q>
      <start xsi:type="xsd:int">0</start>
      <maxResults xsi:type="xsd:int">10</maxResults>
      <filter xsi:type="xsd:boolean">0</filter>
      <restrict xsi:type="xsd:string"></restrict>
      <safeSearch xsi:type="xsd:boolean">0</safeSearch>
      <lr xsi:type="xsd:string"></lr>
      <ie xsi:type="xsd:string"></ie>
      <oe xsi:type="xsd:string"></oe>
    </galactivism:doGoogleSearch>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



Toma lo que necesites,
deja lo que puedas.

Google Search

Esta es la respuesta de **Google**. Los datos ya obtenidos los podemos manipular según nuestra conveniencia:

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
<ns1:doGoogleSearchResponse xmlns:ns1="urn:GoogleSearch"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<return xsi:type="ns1:GoogleSearchResult">
<documentFiltering xsi:type="xsd:boolean">false</documentFiltering>
<estimatedTotalResultsCount
xsi:type="xsd:int">3880</estimatedTotalResultsCount>
<directoryCategories
xmlns:ns2="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="ns2:Array" ns2:arrayType="ns1:DirectoryCategory[0]">
</directoryCategories>
<searchTime xsi:type="xsd:double">0.247468</searchTime>
<resultElements xmlns:ns3="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="ns3:Array" ns3:arrayType="ns1:ResultElement[10]">
<item xsi:type="ns1:ResultElement">
<cachedSize xsi:type="xsd:string">6k</cachedSize>
<hostName xsi:type="xsd:string"></hostName>
.
```



Toma lo que necesites,
deja lo que puedas.

Google Search

Nosotros podemos tomar esta respuesta y formatearla de acuerdo a nuestras necesidades:

```
Array
(
  [documentFiltering] => false
  [estimatedTotalResultsCount] => 3880
  [directoryCategories] =>
  [searchTime] => 0.05159
  [resultElements] => Array
  (
    [0] => Array
    (
      [cachedSize] => 6k
      [hostName] =>
      [snippet] => <b>...</b> Check out. what we've
      cooked up in our brand-new PHP section. Copyright 1998-2002<br>
      <b>Melonfire</b>. All rights reserved Terms and Conditions | Feedback.

      [relatedInformationPresent] => true
      [directoryTitle] => <b>Melonfire</b>
      [summary] => A production house with business
      divisions focusing on content production and Web development.
      [URL] => http://www.melonfire.com/
      [title] => <b>melonfire</b>!
```



Toma lo que necesites,
deja lo que puedas.

Google Search

Para el usuario final todo este proceso es transparente, y el resultado final se genera como **HTML**:





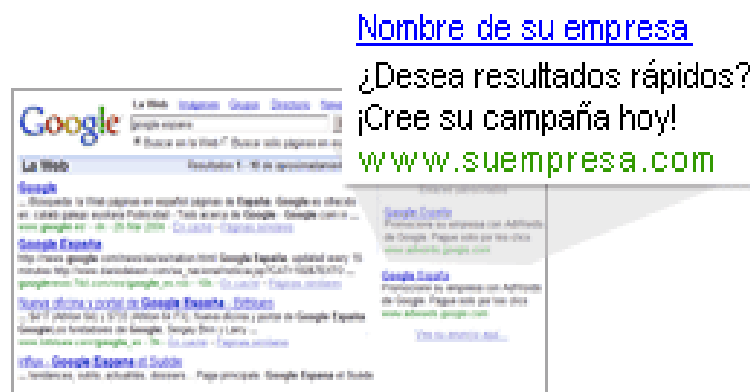
Toma lo que necesites,
deja lo que puedas.

Google AdWords

Google ofrece a personas y empresas que desean anunciarse en su buscador el servicio de Google AdWords. Los anuncios de AdWords les permiten llegar a nuevos clientes en el momento en que éstos buscan sus productos o servicios en Internet.

Con AdWords de Google es posible crear sus propios anuncios, elegir las palabras clave que nos ayuden a orientar los anuncios a sus usuarios y pagar sólo cuando se haga clic en ellos.

El servicio de AdWords es uno que requiere de la evaluación constante de diversos factores para mantener una campaña de publicidad exitosa, y cuando se llevan muchas campañas -como en el caso de los resellers de AdWords- es necesario automatizar en la medida de lo posible las altas de clientes, anuncios, palabras, y la generación de reportes.





Toma lo que necesites,
deja lo que puedas.

Google AdWords

Para utilizar el **API** de **Google AdWords** es necesario tener al menos **una cuenta activa de anuncios** en el momento de registrarse como desarrollador. Una vez registrado, se tiene acceso a:

- **Una clave (token) de programador** para acceder a '**Mi Centro de Clientes**' (*My Client Center* o *MCC*).
- **Una cuota de unidades para consultas.** Cada consulta al servicio de MCC de AdWords tiene un costo diferente, y hay un límite a la cantidad de puntos que se asignan dependiendo de la cantidad de campañas que se manejen con el mismo token.
- **Un Centro API de AdWords** dentro del mismo MCC para revisar el saldo de puntos, y tener acceso a los blogs, grupos y foros de desarrollo relacionados con AdWords.





Toma lo que necesites,
deja lo que puedas.

Google AdWords

Una vez registrado es posible empezar a escribir aplicaciones que tengan conexión con este servicio de publicidad.

Al igual que el servicio de búsqueda Web, el **API** de **Google AdWords** funciona bajo un servicio Web basado en **SOAP**, por lo podemos escribir aplicaciones tanto de escritorio como basadas en web en una amplia variedad de lenguajes, que incluyen a **Java**, **Python** y **PHP**.





Toma lo que necesites,
deja lo que puedas.

Google AdWords

Este es un ejemplo de una consulta para reporte de **palabras clave** (*keywords*) de una campaña de **Google AdWords**. Primero es necesario identificarse:

```
// Set up the authentication headers
$email = "<email>youremail@suempresa.com</email>";
$password = "<password>password</password>";
$userAgent = "<useragent>EMPRESA -- PHP Demo de reportes de palabras clave</useragent>";
$token = "<token>token-google-adwords</token>";
$header = $email . $password . $userAgent . $token;

// Connect to the WSDL for the ReportService
require_once('nusoap.php');
$wsdl = "https://adwords.google.com/api/adwords/v2/ReportService?wsdl";
$client = new soapclient($wsdl, 'wsdl');

// Set the headers; they are needed for authentication
$client->setHeaders($header);
```





Toma lo que necesites,
deja lo que puedas.

Google AdWords

A continuación definimos los datos que requerimos:

```
// Specify the report criteria
$reportName = "<name>Prueba de reporte de Keywords</name>";
$aggregation = "<aggregationType>Summary</aggregationType>";
$startDate = "<startDate>2006-02-01T00:00:00Z</startDate>";
$endDate = "<endDate>2006-02-30T00:00:00Z</endDate>";

// Could optionally specify campaigns, but the default is to get all campaigns
// <campaigns>1234567</campaigns>

$reportjobparams = "<job xsi:type='KeywordReportJob'>" . $reportName . $aggregation . $startDate . $endDate .
"</job>";

// Construct the XML string for the parameters
// It's a nusoap thing that the param string needs to include the operation name too
$reportJob = "<scheduleReportJob xmlns='https://adwords.google.com/api/adwords/v2'>" . $reportjobparams .
"</scheduleReportJob>";
```



Toma lo que necesites,
deja lo que puedas.

Google AdWords

Enviamos la solicitud y esperamos la respuesta. Dependiendo del tráfico el tiempo de respuesta puede variar:

```
// Make the request to schedule the report job
$jobid = $client->call("scheduleReportJob", $reportJob);
$jobid = $jobid['scheduleReportJobReturn'];

// Handle any SOAP faults.
if($client->fault) {
    echo "<P>FAULT: {"$client->fault}<br>\n";
    echo "<P>Code: {"$client->faultcode}<br>\n";
    echo "<P>String: {"$client->faultstring}<br>\n";
    echo "<P>Detail: {"$client->faultdetail}<br>\n";
    return;
}

// If we got to here, the report was scheduled successfully
echo "<P>There are " . count($jobid) . " elements in the response array.<br>\n";
echo "<P>The reportJob id is $jobid<br>\n";

// Now we need to wait until the report job is ready
$jobidparam = "<reportJobId>$jobid</reportJobId>";
```



Toma lo que necesites,
deja lo que puedas.

Google AdWords

```
// It's a nusoap thing that the param string needs to include the operation name too
$statusxml = "<getReportJobStatus xmlns='https://adwords.google.com/api/adwords/v2'> $jobidparam
</getReportJobStatus>";

$jobstatus = $client->call("getReportJobStatus", $statusxml);
$jobstatus = $jobstatus['getReportJobStatusReturn'];
// echo "<P>There are " . count($jobstatus) . " elements in the status array.<br>\n";
echo "Status for report job $jobid is $jobstatus<br>\n";

while((strcmp($jobstatus, "InProgress") == 0) || (strcmp($jobstatus, "Pending") == 0)) {
    // check the status every 30 seconds
    sleep(30);
    $jobstatus = $client->call("getReportJobStatus", $statusxml);
    $jobstatus = $jobstatus['getReportJobStatusReturn'];
    echo "Status for report job $jobid is $jobstatus<br>\n";
}

echo "<P>No longer waiting, check if the report is ready or failed</P><br>\n";

if(strcmp($jobstatus, "Failed") == 0) {
    echo "The report generation process failed, sorry!<br>\n";
    return;
}
```



Toma lo que necesites,
deja lo que puedas.

Google AdWords

Finalmente y si no hubo errores o tiempos de espera agotados, el *script* recoge el reporte y lo guarda en disco.

```
// The report job completed so get the URL to the report and download it
// NOTE: You must have the CURL package installed for this to work
// see http://curl.haxx.se/
if(strcmp($jobstatus, "Completed") == 0) {
    $downloadURLxml = "<getReportDownloadUrl xmlns='https://adwords.google.com/api/adwords/v2'>
$jobidparam </getReportDownloadUrl>";
    $reportURL = $client->call("getReportDownloadUrl", $downloadURLxml);
    $reportURL = $reportURL['getReportDownloadUrlReturn'];
    echo "<P>Download URL is $reportURL<br>\n";

    // Make sure the specified filename exists and is writable
    $fp = fopen("report1.xml", "w");

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $reportURL);
    curl_setopt($ch, CURLOPT_FILE, $fp);

    // Download the report
    $result = curl_exec($ch);
    curl_close($ch);
}
?>
```



Toma lo que necesites,
deja lo que puedas.

Google Sitemaps

Google Sitemaps es un experimento en el ramo de búsqueda en Internet. Al usar **Sitemaps** para notificar al buscador de **Google** directamente cuando hay cambios en un sitio web se acelera la indexación y la adición de esas páginas en el buscador.

<http://www.google.com/> [About Google Sitemaps](#) [Sitemaps FAQ](#) [About statistics we show](#)

[Sitemaps](#) [Stats](#) [Errors](#) [robots.txt](#)

[Query stats](#) | [Crawl stats](#) | [Page analysis](#) | [Index stats](#)

These tables provide information about Google search queries that returned pages from your site during the last three weeks.

Top search queries are the queries that most often returned pages from your site and **top search query clicks** are the top queries that directed traffic to your site (based on the number of clicks to your pages). **Average top position** is the highest position any page from your site ranked for that query, averaged over the last three weeks. Since our index is dynamic, this may not be the same as the current position of your site for this query.

Top searches

[Download as .csv file](#)

Top search queries	Average top position
1. google	1
2. google maps	3
3. google video	8
4. traductor	3
5. translator	5

[Download as .csv file](#)

Top search query clicks	Average top position
1. google	1
2. google talk	1
3. adsense	1
4. translate	4
5. "www.google.com"	2



Toma lo que necesites,
deja lo que puedas.

Google Sitemaps

Básicamente hay tres pasos a tomar para utilizar el servicio de **Google Sitemaps**:

1. Crear un archivo en formato **Sitemap** del sitio web a indexar.
2. Añadir el sitio y el URL del archivo a una cuenta de **Sitemaps**
3. Actualizar el archivo **Sitemap** cuando el sitio sufra cambios.

No hay límite a la cantidad de sitios que se pueden añadir (aunque es una cuenta por dominio único) y el servicio es completamente gratuito.

Lo que si hay es un máximo de 50,000 páginas por sitio y un límite de 10 Mb por archivo **Sitemap**.





Toma lo que necesites,
deja lo que puedas.

Google Sitemaps

El archivo de formato Sitemap (o protocolo Sitemap, como lo llaman algunos) es un sencillo archivo de texto en formato XML que describe los URLs absolutos de las páginas que deseamos que sean indexadas y la última fecha en que fueron actualizadas.

```
- <urlset>
  - <url>
    <loc>http://www.google.com/BUILD</loc>
    <lastmod>2005-04-30T03:45:08+00:00</lastmod>
  </url>
  - <url>
    <loc>http://www.google.com/about.html</loc>
    <lastmod>2005-05-07T00:45:09+00:00</lastmod>
  </url>
  - <url>
    <loc>http://www.google.com/addurl.html</loc>
    <lastmod>2005-03-11T23:45:16+00:00</lastmod>
  </url>
```



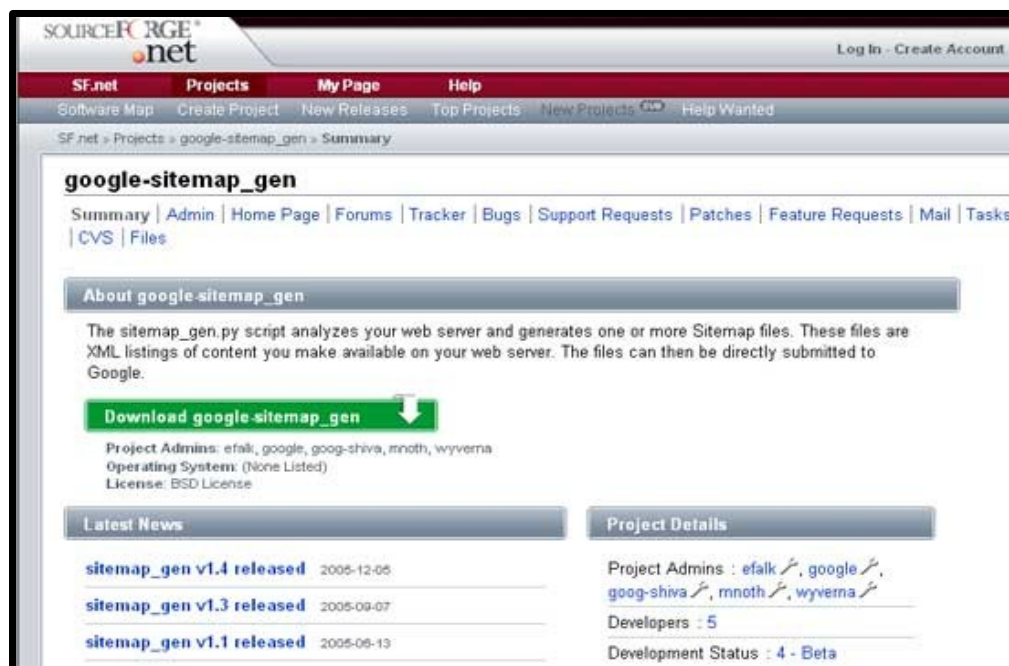
Hay un ejemplo en <http://www.google.com/sitemap.xml>.



Toma lo que necesites,
deja lo que puedas.

Google Sitemaps

Aunque cada webmaster puede implementar la generación del archivo **Sitemap** como mejor convenga a su criterio (desde editarlo a mano hasta la generación con un pequeño crawler o un analizador de *logs*), **Google** a creado una pequeña aplicación escrita en **Python** llamada **Google Sitemap Generator** bajo licencia **BSD** disponible para su descarga en **Sourceforge.net**



<http://code-sitemapgen.sourceforge.net/>

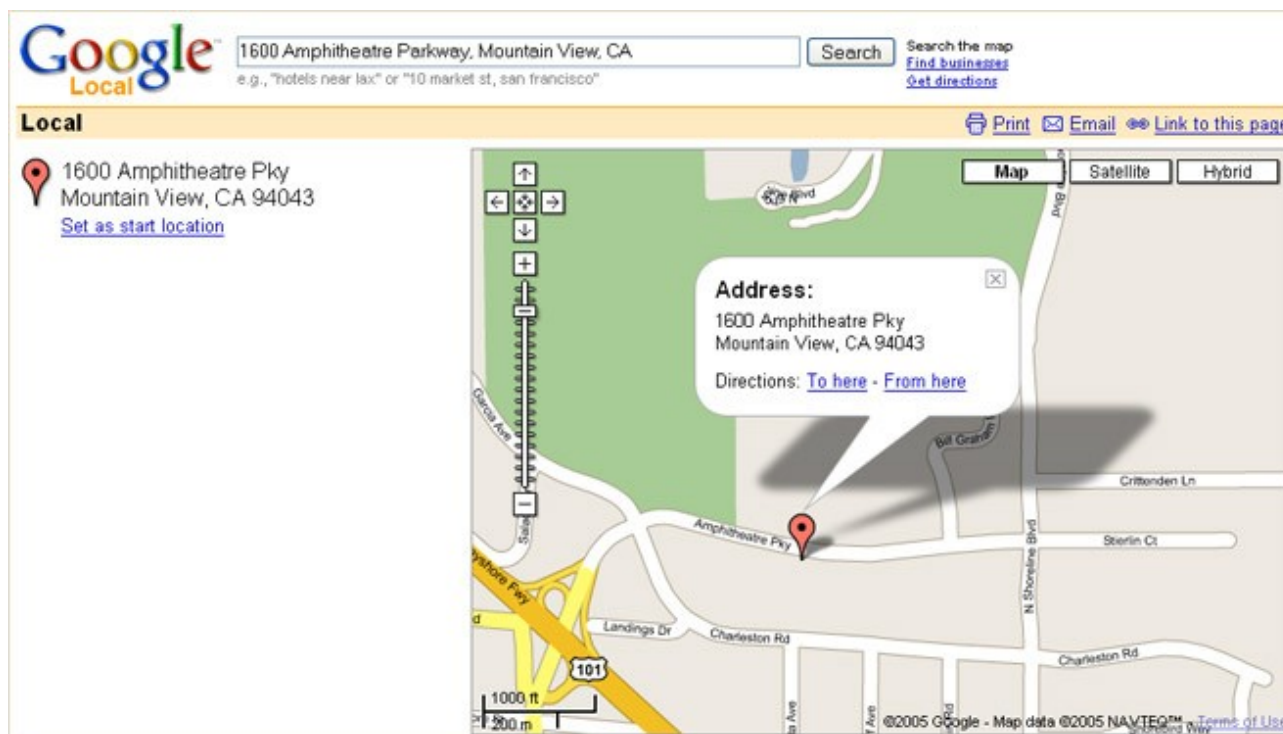
<http://www.webmasters.org.mx/>



Toma lo que necesites,
deja lo que puedas.

Google Maps

Google Maps (llamado también **Google Local**) es un servicio de búsqueda y navegación de mapas aéreos y que cubre prácticamente todo el mundo (*¡y parte de la Luna!*).



El **API de Google Maps** permite a los webmasters y desarrolladores web integrar estos mapas en sus propias páginas, y añadir marcadores, líneas, áreas de color y cajas de diálogo personalizadas.



Toma lo que necesites,
deja lo que puedas.

Google Maps

Google Maps es un servicio todavía en beta y disponible para uso gratuito, aunque Google se reserva el derecho de en el futuro colocar publicidad en los mapas.

Google Maps nos ofrece dos versiones del servicio: **la vista de calle** y **la vista en fotografía real**.

Cualquiera de estos está sujeto a disponibilidad por zonas. En el caso de México, aún no está disponible la vista de calle, y en algunos casos las fotografías tienen poca resolución, sobre todo en áreas rurales.

Para tener acceso al servicio es necesario registrarse previamente para obtener una clave que nos permite tener acceso a la librería de mapas.

El registro se hace por dominio a utilizar el servicio y no hay límite a los dominios que se pueden registrar.





Toma lo que necesites,
deja lo que puedas.

Google Maps

Construiremos un ejemplo con un mapa en la ciudad de México.

Primero cargamos los archivos JavaScript utilizando la llave de nuestro registro:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>

    <title>Google Maps JavaScript API Example: simple</title>
    <script src="http://maps.google.com/maps?file=api&v=1&key=ABQIAAAAEfCuQGsNiSWxRgf_vfNwaRQjsk11-
YgiA_BGX2yRrf7htVrbmBTEB0IH-F489GrwP8-dHLib7cKKIQ"
      type="text/javascript">
    </script>

    <script type="text/javascript">.
```





Toma lo que necesites,
deja lo que puedas.

Google Maps

A partir de ahí creamos una rutina para configurar nuestro mapa: la coordenada central, el nivel de zoom, si queremos colocar marcadores, cajas, eventos, etc.

```
<script type="text/javascript">
//

function onLoad() {
  // The basics.
  //
  // Creates a map and centers it on Palo Alto.

  if (GBrowserIsCompatible()) {
    var map = new GMap(document.getElementById("map"));
    map.centerAndZoom(new GPoint(-99.15023, 19.3030), 4);
  }
}

//]]&gt;
&lt;/script&gt;
&lt;/head&gt;</pre></div><div data-bbox="50 832 910 867" data-label="Text"><p>Hay que recordar que <b>Google Maps</b> requiere las coordenadas geográficas en <b>radianes</b>, no en <b>grados</b>.</p></div><div data-bbox="63 878 175 941" data-label="Image"><img alt="Three small spheres in green, blue, and yellow."/></div><div data-bbox="662 935 891 962" data-label="Page-Footer"><p><a href="http://www.webmasters.org.mx/">http://www.webmasters.org.mx/</a></p></div>
```



Toma lo que necesites,
deja lo que puedas.

Google Maps

Para convertir grados de longitud y latitud se utiliza la siguiente equivalencia:

$$180 \text{ grados} = \pi \text{ radianes}$$

De la que podemos deducir que:

$$\frac{\pi}{180^\circ} \theta$$

Para las longitudes al oeste del Meridiano de **Greenwitch** y al sur del **Ecuador** se utiliza la notación negativa para las mediciones.

Para el **Estadio Azteca**, en la **Cd. de México**, utilizaremos las coordenadas:

Latitud: 19.30304299778644

Longitud: -99.15023803710938





Toma lo que necesites,
deja lo que puedas.

Google Maps

Finalmente, agregamos la caja en el cuerpo de la página HTML en el tamaño que deseamos mostrar el mapa, y ya está.

```
<body onload="onLoad()">  
    <div id="map" style="width: 500px; height: 300px"></div>  
</body>  
</html>
```

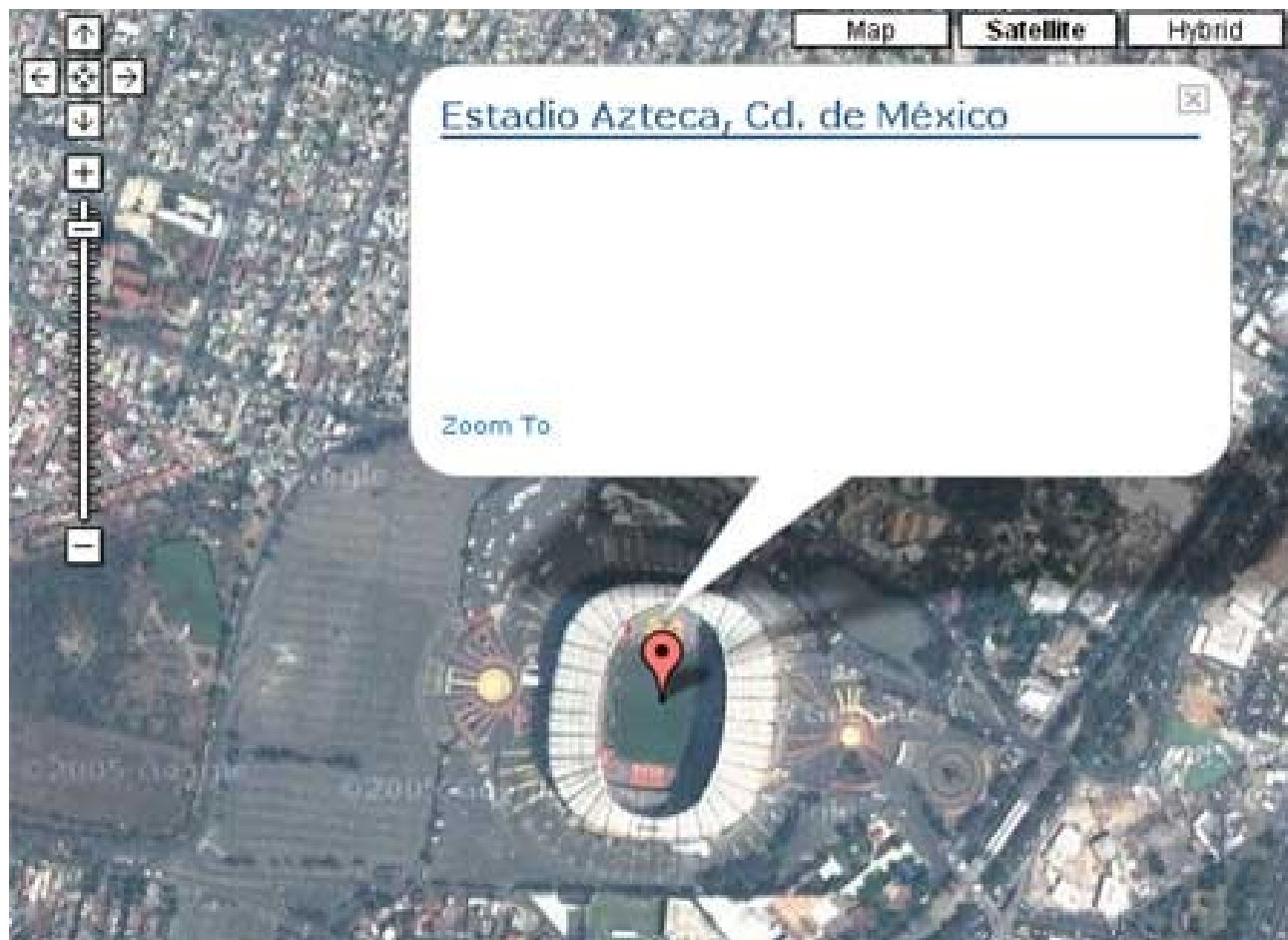




Toma lo que necesites,
deja lo que puedas.

Google Maps

El resultado final se algo como esto:





Toma lo que necesites,
deja lo que puedas.

¡Eso es-eso es- es todo, amigos!

Ahora las preguntas...





Toma lo que necesites,
deja lo que puedas.

Una última cosa...



Toma lo que necesites,
deja lo que puedas.

Webmasters.org.mx

Objetivo y misión:

- Generar espacios para intercambio de experiencias, opiniones, conceptos, soluciones y proyectos del mundo de Internet mexicano con la intención de que esto conduzca a un crecimiento, a una mejora continua, a un estándar de calidad cada vez más alto, a un mayor nivel de competitividad internacional en un mundo globalizado y a dar una mayor difusión a las actividades, capacidades y habilidades de todo el gremio, creando sinergia entre sus integrantes.



Toma lo que necesites,
deja lo que puedas.

Webmasters.org.mx

Metas:

- Servir como medio de contacto entre quienes requieren desarrollos para Internet en todos sus conceptos (diseño, programación, planeación, puesta en marcha, sostén tecnológico) y quienes los proveen en México.
- Establecer un mecanismo de retroalimentación, apoyo y reconocimiento entre colegas e integrantes de la comunidad.
- Generar una comunidad que marque la pauta del Internet mexicano, que construya opiniones de peso sobre ésta y a la vez que logre el mayor y mejor aprovechamiento de esta tecnología en nuestro país.



Toma lo que necesites,
deja lo que puedas.

Webmasters.org.mx

Más información en

<http://www.webmasters.org.mx/manifiesto.php>

Suscripción libre y gratuita en

<http://www.webmasters.org.mx/forma.php>

- Foros Web:
<http://www.webmasters.org.mx/forosweb>
- Lista de correo:
<http://www.webmasters.org.mx/lista>
- Proyectos:
<http://www.webmasters.org.mx/proyectos>
- Blog:
<http://www.webmasters.org.mx/blog>



Toma lo que necesites,
deja lo que puedas.

¡Muchas Gracias!

A sus órdenes en Internet:

<http://www.webmasters.org.mx/>

Ing. Mauricio Angulo S.

Coordinación y programación

mauricioangulo@webmasters.org.mx

Blog: <http://www.thebookoffate.com/>