

Tutorial API Google maps

Juan Ramón Pérez Pérez

¿Qué es el API de Google Maps?

- ▶ El API de Google Maps te permite **incrustar Google Maps** en tus **propias páginas web** mediante **JavaScript**.
- ▶ El API proporciona:
 - Utilidades para manipular los mapas
 - Añadir contenido al mapa mediante diversos servicios
- ▶ Crear potentes aplicaciones para mapas en tu sitio web.
- ▶ Servicio gratuito disponible para cualquier sitio web que sea gratuito para los usuarios.
- ▶ <http://code.google.com/intl/es/apis/maps/index.html>

Clave de Google Maps

- ▶ Es necesario (teóricamente) el uso de una clave
- ▶ La clave está ligada a un dominio y un directorio.
- ▶ La clave se recibe a través de un cuenta en Google.
- ▶ <http://code.google.com/intl/es/apis/maps/signup.html>

Recomendaciones generales

- ▶ Compatibilidad con navegadores
 - GBrowserIsCompatible()
- ▶ Se recomienda utilizar código XHTML compatible con los estándares
- ▶ Para mostrar polilíneas en un mapa se debe incluir el espacio de nombres VML en el documento XHTML para que funcione correctamente en Internet Explorer.
- ▶ Versión de la API:
 - El parámetro v en la dirección URL `http://maps.google.com/maps?file=api&v=2` hace referencia al número de *versión* que debe emplear el API de Google Maps
 - Actual (octubre 2008), versión 2
- ▶ Reducción de fugas de memoria del navegador
 - GUnload()

```

<!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>Ejemplo del API JavaScript de Google Maps</title>
    <script
      src="http://maps.google.com/maps?file=api&v=2&key=la_clave"
      type="text/javascript"></script>
    <script type="text/javascript">

      function load() {
        if (GBrowserIsCompatible()) {
          var map = new GMap2(document.getElementById("map"));
          map.setCenter(new GLatLng(43.354810,-5.851805), 14);
        }
      }

    </script>
  </head>
  <body onload="load()" onunload="GUnload()">
    <div id="map" style="width: 500px; height: 300px"></div>
  </body>
</html>

```

Carga la API
de Google
Maps

Crea un
objeto mapa

Inicializa el
mapa

Carga el
mapa

Elemento *div* para
alojar el mapa



GMap2: el objeto fundamental

- ▶ Es la clase JavaScript que representa a los mapas.
- ▶ Creamos nuevas instancias mediante el operador new.
- ▶ Al crear una nueva instancia, se especifica un nodo DOM como contenedor para el mapa.
- ▶ Todos nodos HTML son hijos del objeto document de JavaScript y obtenemos una referencia a este elemento mediante el método `document.getElementById()`


Latitud y longitud

- ▶ ¿Cómo hacer referencia a ubicaciones dentro del mapa?
- ▶ El objeto `GLatLng` proporciona esta posibilidad
- ▶ Se construyen pasando sus parámetros en el orden habitual en cartografía {latitud, longitud}

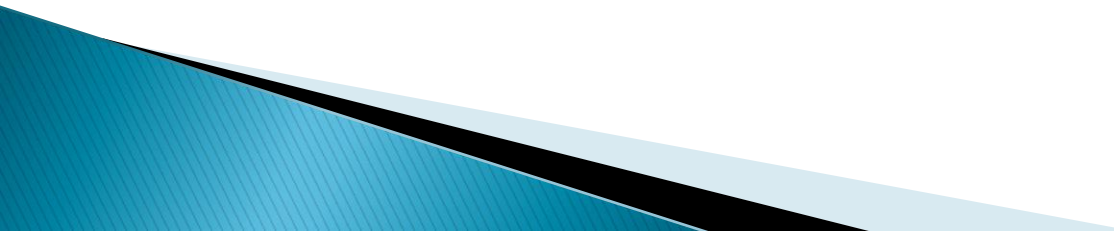
Atributos y ventanas de información del mapa

- ▶ Tipos de mapas:
 - G_HYBRID_MAP: mezcla de vistas normales y de satélite.
 - Se puede definir el tipo de mapa mediante el método `setMapType()` del objeto `GMap2`.
- ▶ Interacción
 - `GMap2.disableDragging()`
- ▶ Cada mapa puede mostrar una única ventana:
 - Cada mapa puede mostrar una única ventana de información del tipo `GInfoWindow`.
 - El objeto `GMap2` proporciona un método `openInfoWindow()`, que requiere un punto y un elemento DOM HTML como argumentos.
 - `openInfoWindowHtml()`

Eventos

- ▶ El navegador genera evento ante las interacciones del usuario y JavaScript permite detectar estos eventos.
 - ▶ JavaScript permite registrar detectores de eventos y cuando estos suceden se ejecuta el código asociado.
 - ▶ Los eventos del API de Google Maps son específicos y distintos de los eventos DOM estándar.
 - ▶ Cada objeto del API de Google Maps exporta una determinada cantidad de eventos con nombres.
- 

Ejemplo de eventos

- ▶ El objeto GMap2 exporta los eventos click, dblclick y move, entre muchos otros
 - ▶ Cada evento se produce en un contexto determinado y pasa argumentos que identifiquen dicho contexto.
 - ▶ El evento mousemove se activa cuando el usuario mueve el ratón sobre un objeto de mapa y pasa el objeto GLatLng de la ubicación geográfica en la que se encuentra el ratón.
- 

Registro, uso y eliminación de detectores de eventos

- ▶ Registro: `GEvent.addListener()`
- ▶ Parámetros:
 - Objeto
 - Evento a detectar
 - Función para llamar cuando se produce el evento
- ▶ Argumentos: se pasan los símbolos especificados a las funciones de captura del evento.
- ▶ Invocar un método sobre todas las instancias de un objeto como respuesta a un evento: `GEvent.bind()`
- ▶ Eliminación: `removeListener(myEventListener)`



Controles

- ▶ Son elementos de interfaz de usuario que permiten interactuar con el map.
- ▶ Se pueden incluir variaciones de ellos en tu aplicación.
- ▶ Se pueden crear controles personalizados mediante subclases de Gcontrol.
- ▶ Adición de controles: `map.addControl()`

Controles integrados

GSmallMapControl



GMapTypeControl

GHierarchicalMapTypeControl

GOverviewMapControl

GLargeMapControl

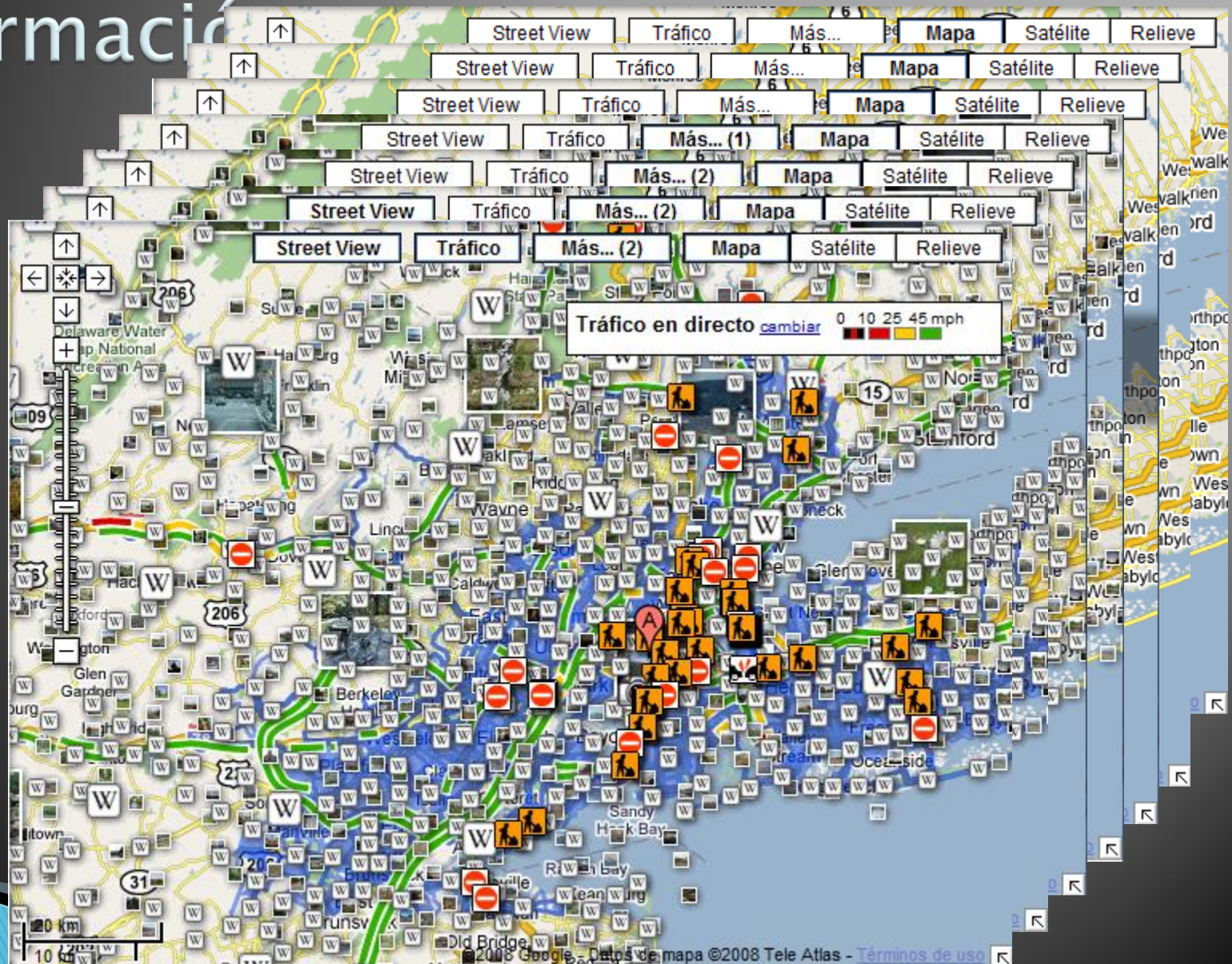
GScaleControl



Superposiciones

- ▶ Objetos del mapa vinculados a coordenadas de latitud y longitud
- ▶ Se mueven al arrastrar o aplicar el acercamiento sobre el mapa
- ▶ Se añaden al mapa para designar puntos, líneas o zonas.
- ▶ Tipos:
 - Puntos, GMarker o GIcon.
 - Líneas, GPolyline
 - Zonas de mapa, polígonos o superposiciones de terreno.
 - Mapa, superposición de mosaico
 - Ventana de información

Superposiciones = capas de información



Codificación geográfica (*geocoding*)

- ▶ Transformar direcciones postales en coordenadas geográficas
- ▶ Servicio de codificación geográfica de Google Maps se puede acceder:
 - Acceso HTTP (servicio Rest)
 - Objeto GClientGeocoder
- ▶ Consume tiempo y recursos
 - Caché de codificación geográfica



Formas de acceder al servicio de geolocalización

► Solicitud HTTP:

- <http://maps.google.com/maps/geo?>
- q: la dirección que SE desea codificar.
- key: la clave del API.
- output: el formato en que deben generarse los resultados. Las opciones son xml, kml, csv o json.

► Ejemplo:

<http://maps.google.com/maps/geo?q=Vald%E9s+Salas,+Oviedo&output=xml&key=abcdefg>

► Objeto GClientGeocoder

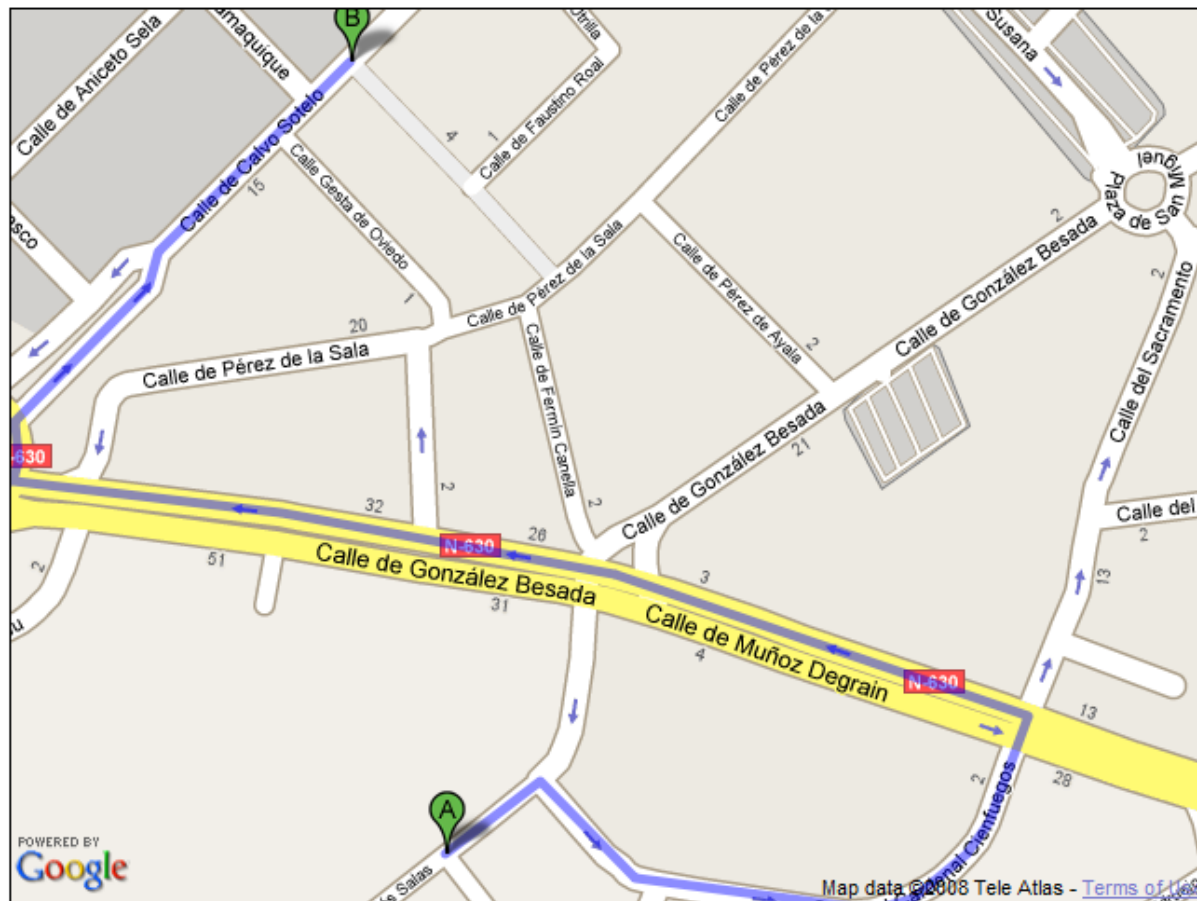
- Método GClientGeocoder.getLatLng()
- Parámetros:
 - Dirección
 - Función callback con un parámetro donde recibe:
 - Si se ubica correctamente, GLatLng
 - Si no se ubica null
 - En caso de direcciones ambiguas sólo se pasa el de mejor coincidencia

Rutas

- ▶ La API de Google Maps permite consultar rutas entre dos puntos.
- ▶ La salida puede ser de dos formas:
 - Dibujada sobre un mapa con una polilinea
 - Descrita textualmente
- ▶ Para crear rutas: GDirections
- ▶ Punto de origen y punto destino, descritos mediante direcciones o coordenadas.
- ▶ También podemos pedir rutas con puntos intermedios.



Visualización de rutas



Calle de Valdés Salas,
33007, Oviedo, Spain

1.1 km (about 3 mins)

1. Head **northeast** on **Calle de Valdés Salas** toward **Calle del Cardenal Cienfuegos** 56 m
2. Turn **right** at **Calle del Cardenal Cienfuegos** 0.3 km
3. Turn **left** at **N-630/ Calle de Muñoz Degrain** 0.5 km
Continue to follow N-630
4. At **Plaza de Castilla**, take the **1st** exit onto **Calle de Calvo Sotelo** 0.3 km



Calle de Calvo Sotelo,
33007, Oviedo, Spain

Map data ©2008 Tele Atlas