



Universidad Cenfotec

Proyecto de Ingeniería de Software 2



Nombre del equipo

TM3

Estudiantes:

Manuel Delgado Rodríguez

Miguel Gutiérrez Barrantes

Marlon Agüero Vargas

Tracy Mora Cordero

Investigación

GMaps

Profesor

Dennis Córdoba López

Período lectivo

I cuatrimestre, 2020

Índice

Introducción	3
¿Qué es GMaps?	4
¿Dónde se encuentra GMaps?	5
¿Para qué se utiliza GMaps?	6
Implementación	7

Introducción

La visualización de mapas como recurso para localizar lugares se ha vuelto una tendencia mucho más valorada para la experiencia de usuario. Con la ayuda de satélites, la generación de aplicaciones que faciliten el acceso a estos datos geográficos se ha vuelto mayor, tomando en cuenta APIs como Google Maps o OSM (OpenStreetMap).

Dicho todo esto, se le conoce a esta tendencia como webmapping, el cual debe involucrar un navegador web para realizar interacciones cliente-servidor y así hacer uso de esos datos geográficos, por medio de HTML, CSS y JavaScript. De aquí parten las librerías webmapping, que funcionan como complementos para facilitar la implementación de mapas, como es el caso de GMaps, que será explicado a detalle en el presente documento.

¿Qué es GMaps?

GMaps consiste en una librería JavaScript, con licencia MIT, de fácil asociación con el API de Google Maps para la exploración de distintas funciones centradas en la geolocalización. Es por esto que GMaps busca la reducción de código por medio de una programación en jQuery entendible para los desarrolladores.

La principal función de GMaps es utilizar el API de Google Maps para crear y visualizar mapas con una amplia gama de componentes como marcadores, rutas, geolocalización, perfiles longitudinales, polígonos, eventos, entre muchos otros. Cabe decir que generar este tipo de funciones únicamente con API generaría un esfuerzo y programación mucho mayor.

Por último, queda claro que GMaps no es un API por sí solo, por lo que no depende de sí mismo para funcionar, ya que existen otras librerías de funciones similares que se complementan con Google Maps, como Leaflet y OpenLayers.

¿Dónde se encuentra GMaps?

Al ser GMaps una librería JavaScript externa de Google Maps. Se debe descargar desde su sitio Web:

<https://hpneo.dev/gmaps/>



Seguido de la descarga de los archivos, se debe agregar el archivo gmaps.js a la lista de scripts en el html, como se explicará más adelante en la implementación.

¿Para qué se utiliza GMaps?

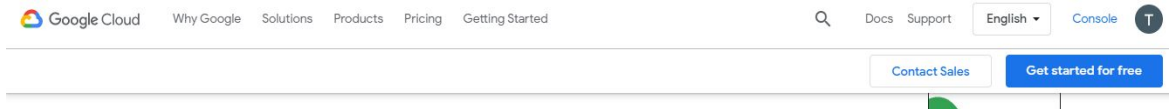
Antes de entender cómo funciona la librería, no está de más recalcar los diferentes objetivos de GMaps. Como se mencionó anteriormente, se creó la librería con el propósito de facilitar el uso de Google Maps con el menor código posible, pero, además:

- La intención de ser un complemento de Google Maps hace que sea fácil consultar lugares y direcciones en el mapa.
- Funciona como un GPS ya que su motor, siendo Google Maps, extrae mapas enteros a nivel satelital y puede asociar direcciones formando rutas de un sitio a otro además del desplazamiento de una persona dentro de la ruta.
- Puede hacer uso de la visualización del terreno con ese modo seleccionado desde Google Maps, por lo que puede verse con mayor detalle cualquier mapa.
- Se puede hacer uso del proyecto Business View de Google en el mapa, por el cual se enfatiza la localización de locales comerciales como medio de promoción, con la ayuda de fotografías, calificaciones, etc.

Implementación

Para poder implementar GMaps lo que se necesita es obtener el API_KEY de Google Cloud Plataform,a continuación se detallaran los pasos a seguir:

1. Ingresar a Google Cloud Platform y selecciona la opción empezar gratis.






2. Completa el formulario para obtener la prueba gratis.



Prueba Google Cloud Platform de manera gratuita

Paso 2 de 2

Información del cliente

 Tipo de cuenta  

Empresa

 Nombre y dirección 

Nombre de la empresa

Nombre

Tray Mora

Línea 1 de la dirección

Línea 2 de la dirección

3. Selecciona o crea un proyecto para el que desea agregar una clave API.

Seleccionar un proyecto NUEVO PROYECTO

Buscar en proyectos y carpetas

RECIENTES TODOS

Nombre	ID
prueba ?	prueba-269702

4. Haga click en el botón de menú y seleccione API y servicios > Credenciales.

The screenshot shows the Google Cloud console interface. On the left, a sidebar menu lists various services: 'Página principal', 'Mercado', 'Facturación', 'API y servicios' (highlighted in blue), 'Asistencia', 'IAM y administración', 'Comenzar', and 'Seguridad'. To the right of the 'API y servicios' item, a dropdown menu is open, displaying a list of options: 'Panel', 'Biblioteca', 'Credenciales' (highlighted), 'Pantalla de consentimiento de OAuth', 'Verificación del dominio', and 'Acuerdos de uso de páginas'. In the background, a blurred view of the 'Credenciales' page is visible, showing a '+ CREAR CREDENCIAL' button and a warning message: 'Recuerda configurar la pantalla de consentimiento de OAuth'.

5. Al ingresar a la página de credenciales, seleccione crear credenciales > Clave API.

[+ CREAR CREDENCIALES](#) [BORRAR](#)

Clave de API

Identifica tu proyecto con una clave de API simple para verificar la cuota y el acceso

ID de cliente de OAuth

Solicita el consentimiento del usuario para que tu app pueda acceder a sus datos

Cuenta de servicio

Habilita la autenticación de servidor a servidor en el nivel de la app mediante cuentas robot

Ayúdame a elegir

Responde algunas preguntas que te ayudarán a decidir qué tipo de credencial usar

6. El cuadro de diálogo muestra la clave API recién generada.

7. Por último, haz clic en cerrar.

8. En la sección de claves de API se mostrará la clave creada recientemente.

Es importante saber que antes se debe agregar el archivo gmaps.js con extensión JavaScript al proyecto donde desea implementar la funcionalidad.

A continuación, se detallarán los pasos a seguir para lograr llevar a cabo el uso de GMaps.

1. En el archivo con extensión .html en la sección del head se debe incluir las referencias de la librería JQuery, la referencia al API de Google Maps.

```
<head>
  <!-- Referencia al BOOTSTRAP 4 -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" int
  <!-- Referencia a la librería de JQUERY -->
  <script src="https://code.jquery.com/jquery-3.4.1.js"></script>
  <!-- Referencia al API de GOOGLE MAPS -->
  <script src="http://maps.google.com/maps/api/js?key=API_KEY"></script>
  <script src="js/gmaps.js"></script>
  <link rel="stylesheet" href="./style/style.css">
</head>
```

2. En el archivo con extensión .html se debe incluir un div el cual debe de tener un identificador map (id) para poder mostrar el mapa.

```
<!-- div para mostrar el mapa -->  
<div id="map" class="map"></div>
```

3. Luego se llama el archivo JavaScript que nosotros creamos.

```
<!-- Referencia al js necesario para renderizar lo que se busca -->  
<script src="js/maps.js"></script>
```

4. Ya estando en el archivo JavaScript se inicializa el objeto tipo mapa con los siguientes atributos:

- ID, el cual indica la referencia del mapa a través del id del div en HTML.
- Zoom, hace referencia al zoom que la persona tiene disponible hacer.
- Latitud y longitud, es el punto central del mapa.

```
//Inicializacion del objeto tipo mapa  
var map = new GMaps({  
  //busca con el id #map  
  el: '#map',  
  //el zoom que la persona tiene permitido hacer  
  zoom: 14,  
  //latitud del centro del mapa  
  lat: 9.9333296,  
  //longitu del centro del mapa  
  lng: -84.0833282,  
  //se añade el evento click y se referencia a la funcion addMarker  
  click: addMarker  
});
```

5. Lo siguiente es agregar los puntos o los markers, gracias a GMaps que ya trae una la funcionalidad de addMarkers, esta nos permite añadir los puntos a lista y decidir cuántos queremos dejar añadir a ella.

```

//la funcion recibe un parametro
function addMarker(e) {
    //se revisa la lista y solo va a permitir crear dos puntos
    if (map.markers.length <= 2) {
        //en el caso que la lista este vacia se añade el punto inicial
        if (map.markers.length == 0) {
            //llama a la funcion del punto inicial
            addStartPoint(e);
        }
        //Si la lista de puntos tiene solamente uno [1] se añade el punto final
        else if (map.markers.length == 1) {
            //llama a la funcion del punto inicial
            addEndPoint(e);
            //llama a la funcion que dibuja la ruta
            drawRoute();
            //por ultimo calcula la ruta
            calculateDistance();
        }
    }
}

```

6. Dependiendo de la condición se puede iniciar la señalización de la ruta con un primer marker o punto inicial (addStartPoint) y después el punto final adonde debe terminar la ruta (addEndPoint). Con esto, GMaps puede hacer el cálculo de las rutas asociando las dos coordenadas mediante el GPS de Google Maps.

```

//Se añade el punto inicial
function addStartPoint(e) {
    map.addMarker({
        //el objeto e que se paso por parametro trae la latitud y la longitud
        lat: e.latLng.lat(),
        lng: e.latLng.lng(),
        title: "Start Point",
        icon: {
            path: google.maps.SymbolPath.CIRCLE, //icono
            scale: 5, //tamaño
            strokeColor: '#f00', //color del borde
            strokeWeight: 10
        }
    });
}

//Se añade el punto final
function addEndPoint(e) {
    map.addMarker({
        //el objeto e que se paso por parametro trae la latitud y la longitud
        lat: e.latLng.lat(),
        lng: e.latLng.lng(),
        title: "End Point",
        icon: {
            path: google.maps.SymbolPath.CIRCLE, //icono
            scale: 5, //tamaño
            strokeColor: '#009975', //color del borde
            strokeWeight: 10
        }
    });
}

```

7. Por último, el dibujo de la ruta se define en la función drawRoute, que, por medio de polylines, que son líneas sobrepuestas en el mapa, dibujan el recorrido de un punto al otro.

```

//Se dibuja la ruta entre los dos puntos
function drawRoute() {
    //En la lista de puntos se toma la posicion cero como el inicial y el 1 como final
    var start = map.markers[0].position;
    var end = map.markers[1].position;
    //llamada a funcion de Gmaps --> drawRoute
    map.drawRoute({
        //Se toma la lat y long del punto inicial
        origin: [start.lat(), start.lng()],
        //Lo mismo con el punto final
        destination: [end.lat(), end.lng()],
        //el tipo de modo de traslado ya sea manejando o en bicicleta por ejemplo
        travelMode: 'driving',
        //se distingue por un color
        strokeColor: '#131540',
        strokeOpacity: 0.6,
        strokeWeight: 6
    });
}

```

