

## U7. Proyecto DWEC

### APIs Geolocalización

#### Índice

1.	Geolocalización con navigator:.....	2
2.	Geolocalización con API Google Maps .....	2
•	API estática .....	2
•	API (dinámica) de Google Map .....	2
•	Gestión de eventos en el mapa .....	4
•	Función createMarker: .....	5
•	Medir distancias con la librería GMaps Geometry .....	6

### 1. Geolocalización con navigator:

La localización consiste en dos componentes de coordenadas: latitud, que es la distancia hacia el norte (positiva) o sur (negativa) a partir del ecuador. Y longitud, que es la distancia hacia el este (positiva) o al oeste (negativa) a partir del Meridiano de Greenwich.

Usamos `navigator.geolocation.getCurrentPosition` para geolocalizarnos. Este método es asíncrono (se ejecuta en segundo plano), por tanto no nos devuelve nada de forma inmediata.

Le pasaremos una función que será ejecutada cuando el navegador encuentre finalmente la localización. Esta función recibirá un parámetro que contendrá la información de dicha localización.

Una segunda función para gestionar errores.

Ver ejemplo 1

### 2. Geolocalización con API Google Maps

La [API de Google Maps](#) nos permite integrar un mapa interactivo en una web.

Para poder usar gran parte de la API de Google, debemos estar registrados como desarrollador y obtener una clave (key) de desarrollo para nuestra web o app.

- **API estática**

Lo más sencillo, una vez que tenemos la localización, es usar [la API estática de Google](#) y obtener una imagen (estática) centrada de nuestra posición con marcadores (opcional). Vamos a ver un ejemplo y su resultado:

```
navigator.geolocation.getCurrentPosition(function(pos) { var latlon =  
pos.coords.latitude + "," + pos.coords.longitude;
```

```
var imgUrl = "http://maps.googleapis.com/maps/api/staticmap?center=" +  
latlon+"&zoom=14&size=400x300&sensor=false&&markers=color:red%7Clabel:C%7C" +latlon;
```

// Ya sólo queda modificar la imagen en nuestro documento donde queramos mostrar el mapa

```
document.getElementById("map").setAttribute('src', imgUrl); });
```

- **API (dinámica) de Google Map**

En primer lugar, debemos crear una clave válida para un proyecto (podemos crear varios proyectos de forma gratuita) en [esta página](#). Necesitamos añadir también un dominio válido que nos permita usar la clave para acceder (podemos usar localhost, plunker, o algún dominio válido donde nuestra página web esté alojada).

## Credenciales

### Crear clave de API de navegador

#### Usar esta clave en la aplicación web

Las solicitudes de API se envían a Google directamente desde el navegador del usuario. [Más información](#)

#### Nombre

#### Aceptar solicitudes de estas URL de referencia HTTP (sitios web) (Opcional)

Usa asteriscos como caracteres comodín. Si lo dejas en blanco, se aceptarán solicitudes de cualquier URL de referencia. Añade URLs de referencia antes de usar esta clave en producción.

✕

✕

Nota: Pueden pasar hasta 5 minutos antes de que se aplique la configuración



Una vez que hemos establecido nuestra clave, debemos copiarla para empezar a usar el API de Google Maps. Para incluir la API con nuestra clave añadimos la ruta al JavaScript con la librería en nuestro documento principal. Debemos hacer algo como esto (¡importante!: No olvides poner la clave):

```
<!DOCTYPE html> <html>
```

```
<head> <title>Ejemplo de obtención de la localización</title> <link
rel="stylesheet" href="style.css">
```

```
</head>
```

```
<body> <div>
```

```
<p id="coordinates">Obteniendo la información de la localización...</p>
```

```
<div id="gmap"></div> </div> <script src="script.js"></script> <script async
defer
```

```
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY">
</script>
```

```
</body> </html>
```

No debemos olvidar asignar el ancho y alto del elemento <div> donde irá el mapa. Este será el tamaño que ocupará el mapa.

```
#map { height: 480px; width: 640px;
```

```
}

function showMap(pos) { document.getElementById("coordinates").textContent
= "Latitude" +

pos.coords.latitude + ". Longitude: " + pos.coords.longitude; // Primero,
necesitamos crear el objeto LatLng (coordenadas para Google) var gLatLng =
new google.maps.LatLng(pos.coords.latitude, pos.coords.longitude);

// Creamos un objeto de opciones con la información para cargar el mapa

var opt = { zoom: 17, // Zoom del mapa (min: 0, max: 20) center: gLatLng, //
Centramos el mapa en nuestra posición mapTypeId:
google.maps.MapTypeId.ROADMAP // Tipo del mapa (Pueden ser:
ROADMAP, SATELLITE, HYBRID, TERRAIN)

};

var mapDiv = document.getElementById("gmap");

var map = new google.maps.Map(mapDiv, opt); // Por último, creamos y
mostramos el mapa. }

// Esperamos a que el DOM esté cargado antes de obtener la posición y cargar el
mapa

window.addEventListener('load', function() {
navigator.geolocation.getCurrentPosition(showMap);

});
```

- **Gestión de eventos en el mapa**

Podemos añadir manejadores de eventos al mapa (tenemos que usar los métodos y los eventos que el API de Google Maps nos facilita). Por ejemplo, eventos como click, dblclick, dragstart, dragend, mousemove, resize, zoom\_changed, etc...

Para añadir un manejador de evento usamos `google.maps.event.addListener` que recibe 3 parámetros: el objeto mapa, el nombre del evento (string) y una función (o referencia a una función) que manejará el evento. Vamos a añadir un evento click en nuestro mapa, por tanto cuando el usuario haga clic en él, el mapa se centrará donde el usuario haya pulsado.

```
google.maps.event.addListener(map, 'click', function(event) { var coordsP =
document.getElementById("coordinates"); coordsP.textContent = "Me muevo a
lat: " + event.latLng.lat() + ", lng: " + event.latLng.lng();
map.panTo(event.latLng);

});
```

Vamos a ver cómo añadir marcadores por defecto en el mapa. Los marcadores recibirán un conjunto de opciones en formato JSON. Las opciones obligatorias son el objeto del mapa, y el objeto latLng donde se pondrá el marcador. Vamos a añadir un marcador al hacer clic (función showMap):

```
// Creamos un marcador rojo en nuestra posición
```

```
createMarker(map, gLatLng, 'red');
```

```
google.maps.event.addListener(map, 'click', function(event) { var coordsP =  
document.getElementById("coordinates"); coordsP.textContent = "Creado un  
marcador en lat: " + event.latLng.lat() +
```

```
", lng: " + event.latLng.lng(); map.panTo(event.latLng);
```

```
// Creamos un marcador verde cada vez que el usuario haga click sobre el mapa
```

```
createMarker(map, event.latLng, 'green'); });
```

- **Función createMarker:**

```
// Recibes un objeto de mapa y un objeto latLng, que será donde se cree el  
marcador
```

```
function createMarker(map, latLng, color) { var opts = {
```

```
position: latLng, map: map, icon: 'http://maps.google.com/maples/ms/icons/' +  
color + '-dot.png'
```

```
};
```

```
var marker = new google.maps.Marker(opts);
```

```
// Añadimos el evento click al marcador también. Este nos mostrará una ventana  
con la información de las coordenadas
```

```
google.maps.event.addListener(marker, 'click', function(event) { var  
infoWindow = new google.maps.InfoWindow();  
infoWindow.setContent("Marker at lat: " + event.latLng.lat().toFixed(6) +
```

```
", lng: " + event.latLng.lng().toFixed(6)); infoWindow.open(map, marker);
```

- Medir distancias con la librería GMaps Geometry

En el API de Google Maps, hay muchas librerías adicionales que podemos usar para geometría, localización de sitios en un mapa o para dibujar. Hay también otras APIs auxiliares como [Google Maps Directions API](#) que podemos usarlas para calcular las distancias andando, en bici, en transporte público o en coche.

Vamos a ver cómo se calcula la distancia entre dos puntos en el mapa (sólo las distancias rectas, no las reales si viajamos). Para incluir esta librería en nuestro script sólo tenemos que incluir un parámetro con las librerías al final, separadas por coma si hay más de una:

```
<script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&libraries
=geometry">

</script>
```

El método `google.maps.geometry.spherical.computeDistanceBetween` (que ya podremos usar) calcula la distancia entre dos objetos `LatLng`:

```
google.maps.event.addListener(map, 'click', function(event) { var coordsP =
document.getElementById("coordinates"); var dist =
google.maps.geometry.spherical.computeDistanceBetween(

gLatLng, // Nuestra posición

event.latLng // Esta posición );

coordsP.innerHTML = "Creado un marcador en: " + event.latLng.lat() + ", lng: "
+ event.latLng.lng() + "<br>" +

"Distancia desde donde estás: " + (Math.round(dist)/1000) + "km";
map.panTo(event.latLng);

// Crea un marcador verde cada vez que hace click

createMarker(map, event.latLng, 'green'); });
```