

Guía de uso de mapas Leaflet

Obtener mi ubicación.....	2
Introducción	2
Conocer mi posición con JS	2
Incorporar mapas Leaflet	3
Funcionalidad JS	4
Crear un mapa	4
Agregar marcadores	5
Agregar un marcador en el mismo punto donde hace centro el mapa	5
Agregar más marcadores en otros puntos.....	5
Agregar un popup a cada marcador.....	6
Crear un área de radio a un marcador	6
Crear un polígono.....	7
Capturar el click del mapa	8
Código	8
Html.....	8
JS.....	9

Obtener mi ubicación.

Introducción

El trabajo con mapas no depende directamente de poder incorporar el uso de la ubicación, pero poder conocer la geolocalización permite explotar el recurso de los mapas a otro nivel, ya que podremos realizar cálculos en tiempo real de distancias, rutas etc.

Los dispositivos que pueden localizarse son los que disponen del hardware de geolocalización. Los teléfonos casi en su totalidad lo pueden hacer.

La ubicación se expresa con valores de latitud y longitud y tienen un formato como este:

-34.9063947,-56.2004327

Conocer mi posición con JS

Si el dispositivo soporta geolocalización, podremos obtenerla utilizando javascript mediante navigator / geolocation.

Creamos un proyecto web, con un archivo JS enlazado.

En la primera línea definimos:

```
navigator.geolocation.getCurrentPosition(SetearCoordenadas);
```

En este ejemplo SetearCoordenadas es la función que se pasa por parámetros a getCurrentPosition. Debemos definirla, sabiendo que setearCoordenadas, recibirá también por parámetros la posición, que posee las coordenadas de la ubicación.

```
function SetearCoordenadas(position){  
    console.log(position.coords.latitude);  
    console.log(position.coords.longitude);  
}
```

Incorporar mapas Leaflet

La incorporación de mapas en esta guía está basada en la [documentación oficial](#).

Incorporar librerías en el documento HTML.

1) Incluir Leaflet CSS en el head.

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.3/dist/leaflet.css"
      integrity="sha256-kLaT2GOSpHechhsozzB+flnD+zUyjE2L1fWPGU04xyI="
      crossorigin=""/>
```

2) Incluir Leaflet JS a continuación del CSS.

```
<script src="https://unpkg.com/leaflet@1.9.3/dist/leaflet.js"
        integrity="sha256-WBkoXOWTeyKc10HuWtc+iz2uENFpDZ9YPdf5Hf+D7ewM="
        crossorigin=""></script>
```

3) Crear un div con id “map” en el body.

```
<div id="map"></div>
```

4) Agregar el siguiente estilo al mapa.

```
#map { height: 680px; }
```

5) Crear y enlazar un archivo para el código javascript.

El html debe quedar similar a esto:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8      <link rel="stylesheet" href="https://unpkg.com/leaflet@1.8.0/dist/leaflet.css"
9          integrity="sha512-hoalWLoI8r4UszCkZ5kL8vayOGVae1oxXe/2A4AO6J9+580uKHD03JdHb7NzwwzK5xr/Fs0
10         crossorigin=""/>
11      <script src="https://unpkg.com/leaflet@1.8.0/dist/leaflet.js"
12          integrity="sha512-BB3hKbKWOc9Ez/TAwybXNXeoV9c1v6FIeYiBieIWkpLJauysF18Nzgr1MBNBXf8/KABdlk
13          crossorigin=""></script>
14      <style>#map { height: 680px; }</style>
15
16 </head>
17 <body>
18     <div id="map"></div>
19     <script src="codigo.js"></script>
20 </body>
21 </html>
```

Funcionalidad JS

Crear un mapa

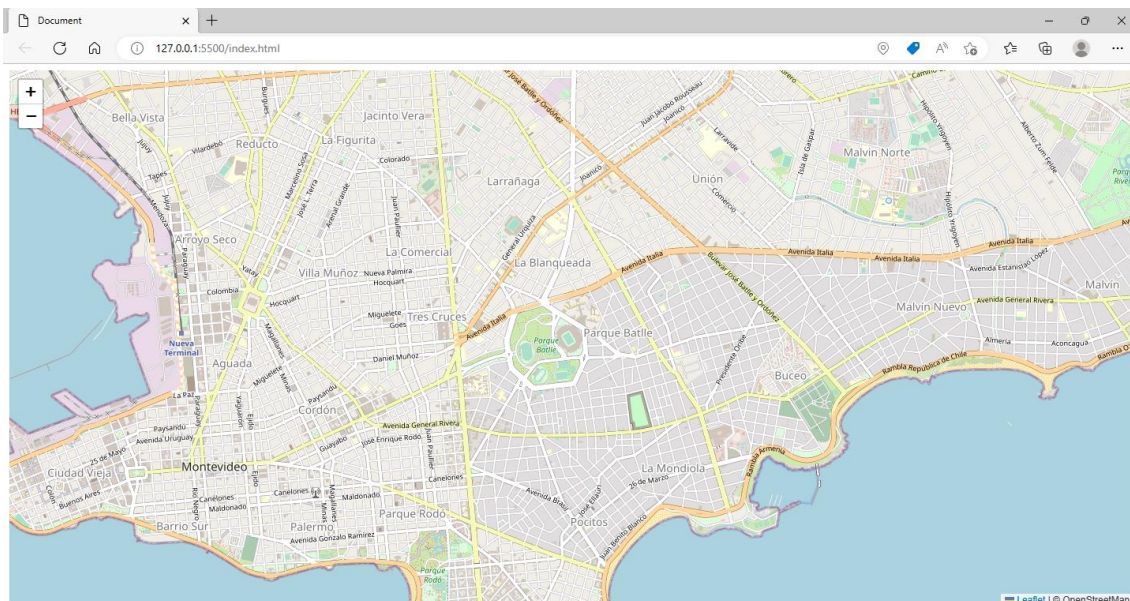
Para poder visualizar un mapa se necesitan como mínimo 2 datos: las coordenadas sobre las que hará centro el mapa, y el nivel de zoom que tendrá sobre ese punto.

La creación del mapa la haremos dentro de una función CrearMapa y el centro lo haremos con coordenadas -34.894517716076805, -56.15253906600995.

```
function CrearMapa(){  
  var map = L.map('map').setView([-34.894517716076805, -56.15253906600995], 14);  
  L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {  
    maxZoom: 19,  
    attribution: '© OpenStreetMap'  
  }).addTo(map);  
}
```

En este caso, creamos un mapa con centro en las coordenadas antes mencionadas, con un zoom por defecto de 14 niveles, y un máximo de 19 niveles de zoom.

Llamamos la función CrearMapa y probamos en el navegador.



Agregar marcadores

Los marcadores pueden estar en cualquier parte del mapa, incluso en zonas no visibles.

Cada marcador deberá tener sus propias coordenadas y estas junto con el nivel de zoom por defecto del mapa, hará que se vean o no.

Agregar un marcador en el mismo punto donde hace centro el mapa

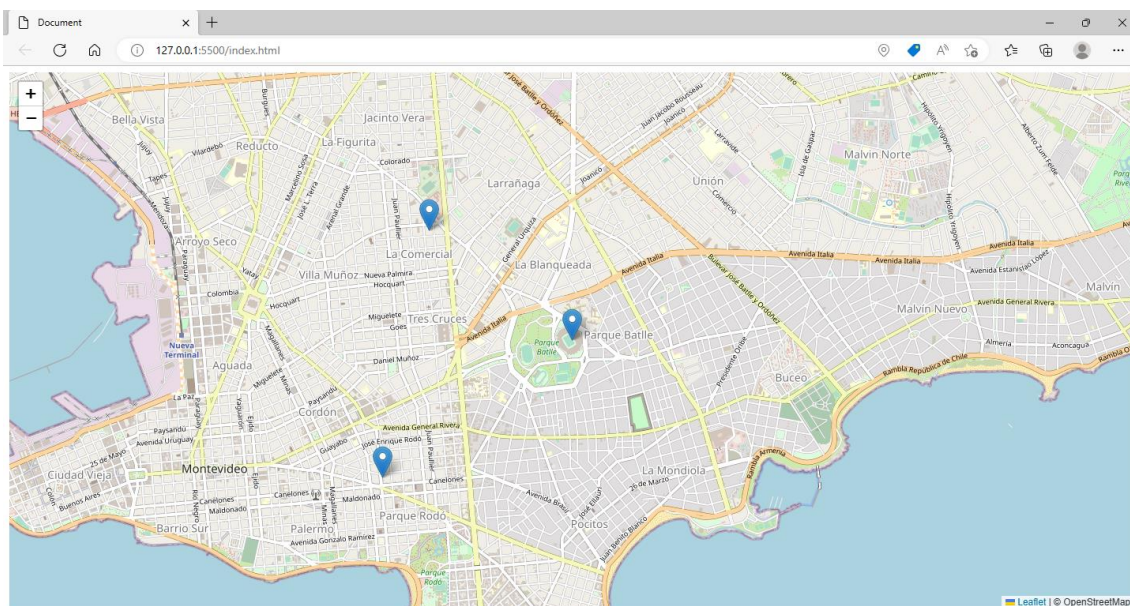
```
function CrearMapa(){  
    var map = L.map('map').setView([-34.894517716076805, -56.15253906600995], 14);  
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {  
        maxZoom: 19,  
        attribution: '© OpenStreetMap'  
    }).addTo(map);  
  
    var marker = L.marker([-34.894517716076805, -56.15253906600995]).addTo(map);  
}
```

Agregar más marcadores en otros puntos.

Las coordenadas se pueden obtener en Maps de Google, o cualquier aplicación de mapas.

Se agregan dos marcadores más.

```
var marker = L.marker([-34.894517716076805, -56.15253906600995]).addTo(map);  
var marker = L.marker([-34.8847165448899, -56.16800846177113]).addTo(map);  
var marker = L.marker([-34.90678678949317, -56.17306564853651]).addTo(map);
```



Agregar un popup a cada marcador.

Un marcador puede tener un popup que muestre información ampliada del punto cuando se clickea sobre él. Es posible pasar código html para resaltar el texto del popup.

```
var marker = L.marker([-34.894517716076805, -56.15253906600995]).addTo(map);
marker.bindPopup("<b>Hola!</b><br>Soy el punto 1").openPopup();|
var marker = L.marker([-34.8847165448899, -56.16800846177113]).addTo(map);
marker.bindPopup("<b>Hola!</b><br>Soy el punto 2").openPopup();
var marker = L.marker([-34.90678678949317, -56.17306564853651]).addTo(map);
marker.bindPopup("<b>Hola!</b><br>Soy el punto 3").openPopup();
```

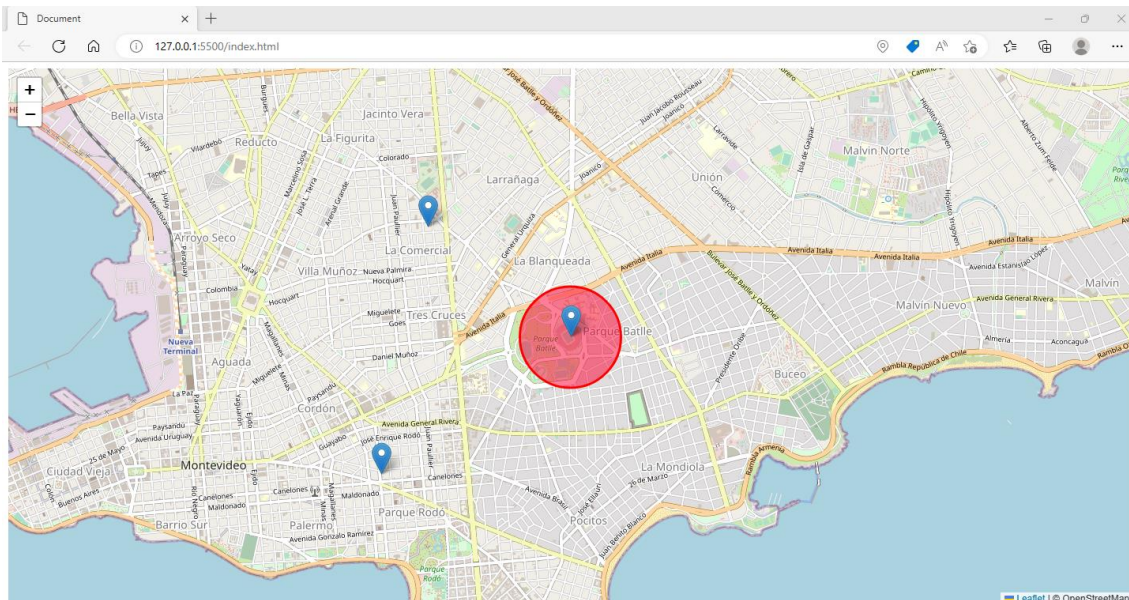
Crear un área de radio a un marcador

Un área de radio crea un círculo con centro en las coordenadas y su tamaño esta expresado en metros. Se pueden dar aspectos de diseño de color y opacidad.

```
var circle = L.circle([-34.894517716076805, -56.15253906600995], {
  color: 'red',
  fillColor: '#f03',
  fillOpacity: 0.5,
  radius: 500
}).addTo(map);
```

Se le puede además agregar un popup al radio, diferente al popup del marcador.

```
circle.bindPopup("Área seleccionada");|
```



Crear un polígono.

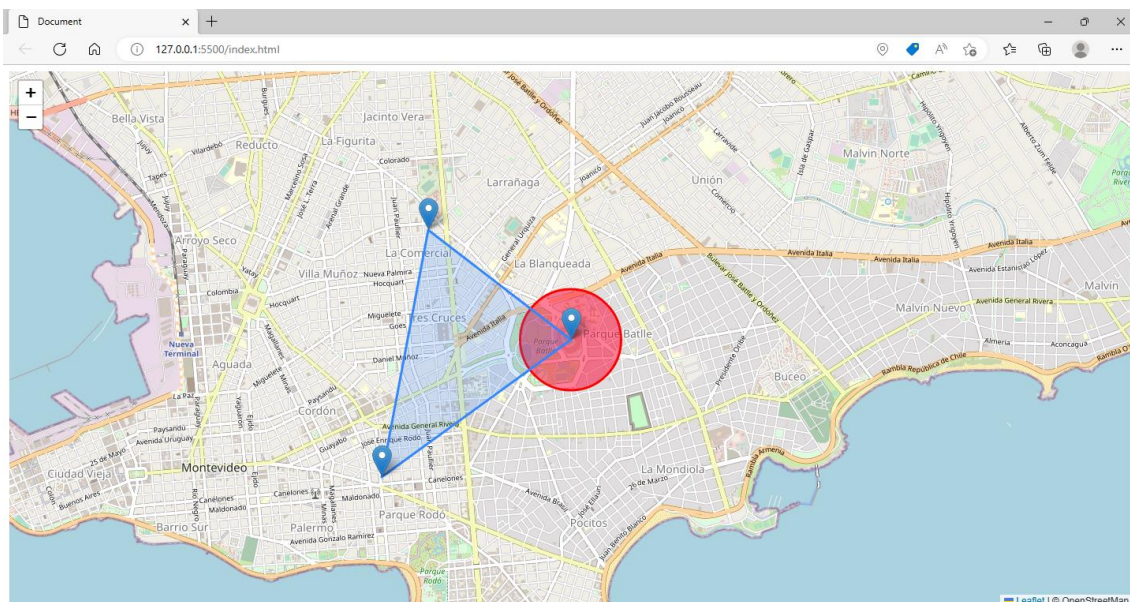
Un polígono une con una línea recta todos los puntos (expresados en coordenadas).

El último punto se une automáticamente con el primero para cerrarlo.

```
var polygon = L.polygon([  
  [-34.894517716076805, -56.15253906600995],  
  [-34.8847165448899, -56.16800846177113],  
  [-34.90678678949317, -56.17306564853651],  
]).addTo(map);
```

Se puede agregar también un popup a un polígono.

```
polygon.bindPopup("Polígono seleccionado");
```



Capturar el click del mapa

Captura el evento de click sobre el mapa. Nos brinda las coordenadas del punto clickeado.

Es necesario dejar escuchando el evento de click sobre el mapa.

```
map.on('click', onMapClick);
```

En este caso, onMapClick es una función que debemos definir fuera de la función sobre la que hemos venido trabajando el mapa.

La función onMapClick recibe por parámetros el evento y este contiene información útil, en este caso utilizaremos las coordenadas exactas donde se tocó.

```
function onMapClick(e) {  
    alert(`Has hecho click sobre Latitud: ${e.latlng.lat} Longitud: ${e.latlng.lng}`);  
}
```

Código

Html

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Document</title>  
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.8.0/dist/leaflet.css"  
        integrity="sha512-  
hoalWLoI8r4UszCkZ5kL8vayOGVae1oxXe/2A4A06J9+580uKHD03JdHb7NzwwzK5xr/Fs0W40kiNHxM9vyTtQ=="  
        crossorigin=""/>  
    <script src="https://unpkg.com/leaflet@1.8.0/dist/leaflet.js"  
        integrity="sha512-  
BB3hKbKWOc9Ez/TAwyWxNXeoV9c1v6FIeYiBieIWkpLjauysF18Nzgr1MBNBXf8/KABdlkX68nAhlwcDFLGPCQ=="  
        crossorigin=""/></script>  
    <style>#map { height: 680px; }</style>  
</head>  
<body>  
    <div id="map"></div>  
    <script src="codigo.js"></script>  
</body>  
</html>
```


JS

```
Inicio();
let MiLat = null;
let MiLong = null;

function Inicio() {
    getLocation();
    CrearMapa();
}

function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(SetearCoordenadas);
    } else {
        console.log("Geolocation is not supported by this browser.");
    }
}

function SetearCoordenadas(position){
    console.log(position.coords.latitude);
    console.log(position.coords.longitude);
}

function CrearMapa(){
    var map = L.map('map').setView([-34.894517716076805, -56.15253906600995], 14);
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
        maxZoom: 19,
        attribution: '@ OpenStreetMap'
    }).addTo(map);

    var marker = L.marker([-34.894517716076805, -56.15253906600995]).addTo(map);
    marker.bindPopup("<b>Hola!</b><br>Soy el punto 1").openPopup();
    var marker = L.marker([-34.8847165448899, -56.16800846177113]).addTo(map);
    marker.bindPopup("<b>Hola!</b><br>Soy el punto 2").openPopup();
    var marker = L.marker([-34.90678678949317, -56.17306564853651]).addTo(map);
    marker.bindPopup("<b>Hola!</b><br>Soy el punto 3").openPopup();

    var circle = L.circle([-34.894517716076805, -56.15253906600995], {
        color: 'red',
        fillColor: '#f03',
        fillOpacity: 0.5,
        radius: 500
    }).addTo(map);

    circle.bindPopup("Área seleccionada");
```

```
var polygon = L.polygon([
    [-34.894517716076805, -56.15253906600995],
    [-34.8847165448899, -56.16800846177113],
    [-34.90678678949317, -56.17306564853651],

]).addTo(map);

polygon.bindPopup("Polígono seleccionado");
map.on('click', onMapClick);
}

function onMapClick(e) {
    alert(`Has hecho click sobre Latitud: ${e.latlng.lat} Longitud: ${e.latlng.lng}`);
}
```