

# IDC – Universidad Nacional de Cuyo White Paper

Simposio de Informática y Modernización del Estado

Buenos Aires – Setiembre de 2018

## Guía para el diseñador de sitio web - Uso de biblioteca leaflet map en la pagian de un sitio

Ricardo R. Palma <sup>A1</sup>, Valentina Milan <sup>A2</sup>, Alejandro Martinez <sup>A3</sup>, Sabrina La Malfa <sup>B4</sup>, Mario Drake <sup>B5</sup>

(1) rpalma@uncu.edu.ar, (2) valentinasmilan@gmail.com,  
(3) alejandrodanielmartinez93@gmail.com, (4) slamalfa@id-c.org.ar, (5) mariodrake@gmail.co

A -Facultad de Ingeniería - Universidad Nacional de Cuyo - Práctica Profesional Supervisada

B IDC Instituto de Desarrollo Comercial - Gobierno de Mendoza - Argentina

## 1 Resumen Ejecutivo

Este document señala como es la estructura mínima necesaria para crear una página que contenga los scripts de los mapas asociados. Representa el relato de un caso de estudio de modernización del estado utilizando herramientas de fuentes abiertas (Open Source)

Documento básico

```
<!DOCTYPE html>
<html>
<head>
<title>Leaflet Web Map</title>
<style>
</style>
```

Jornadas Argentinas de Informática e Investigación Operativa

Simposio de Informática e Modernización de Estado – Buenos Aires Setiembre de 2018

```
</head>
<body>
<script>
</script>
</body>
</html>
```

## 2 Referencias a la hoja de estilos y los java script

Para hacer uso de la biblioteca, se necesitan referencias a la CSS y el script de java dentro de la estructura de archivo html.

Para ello debería referenciarse a las bibliotecas localizadas dentro de la carpeta que alberga el host con los mapas.

Para ello inserte este código en la cabecera <head> del archivo html.

```
<link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
```

```
<script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
```

## 3 Antecedentes y fuentes

Leaflet es una librería javascript para la creación de mapas interactivos. Nos permite utilizar los tiles del servicio que prefiramos para pintar el mapa. Para este tutorial utilizaremos los tiles de OpenStreetMap.

Se recomienda crear un pequeño proyecto con el que podremos probar la librería. Se puede clonar el repositorio <https://github.com/DaniOtero/leaflet-demo>

## 4 Entorno de desarrollo

El tutorial está escrito usando el siguiente entorno:

Hardware: Portátil EXO Pro 14' (2.5 GHz Intel Core i3, 8 GB 1600 MHz DDR3).

Sistema Operativo: Linux Ubuntu 14:04 LTS

Google Chrome 50.0.2661.102

## 5 Añadir leaflet a nuestro proyecto

Para añadir leaflet a nuestro proyecto basta con incluir la librería javascript y el css que nos proporciona leaflet. Por comodidad, añadiremos ambas desde el CDN tal como se señaló en el resumen inicial.

```
<html>
  <head>
    <title>Tutorial leaflet.js</title>
    <link rel="stylesheet" href="http://cd-
n.leafletjs.com/leaflet/v0.7.7/leaflet.css" />
  </head>
  <body>
    <h1>Leaflet.js Demo</h1>
    <div id="map" style="height: 100%"></div>
```

```
        <script type="text/javascript" src="http://cd-
n.leafletjs.com/leaflet/v0.7.7/leaflet.js"></script>
    </body>
</html>
```

## 6 Carga del del mapa

Para poder cargar el mapa, necesitamos indicarle a leaflet el identificador un elemento div cargado en el DOM de nuestra página. Le indicaremos a leaflet que utilice el elemento con id “map”.

```
var demoMap = L.map('map').setView([40.453191,
-3.509236], 6);
```

La llamada a la función “setView” nos centrará el mapa en las coordenadas que le indiquemos (lat, lng) y con el nivel de zoom que deseemos.

A continuación, tenemos que añadirle un “tileLayer” a nuestro mapa. Un “tile” es una imagen que representa un área determinada. Cada nivel de zoom cargará los “tiles” asociados a ese nivel de zoom, los cuales tendrán más o menos detalle en función de la escala del nivel de zoom.

Para cargar los tiles de OpenStreetMap, utilizaremos el siguiente código

```
var osmUrl = 'http://{s}.tile.openstreetmap.org/
{z}/{x}/{y}.png';
var osmAttrib = 'Map data © <a href="http://open-
streetmap.org">OpenStreetMap</a> contributors';
var osm = new L.TileLayer(osmUrl, {
```

```
        minZoom: 5,  
        maxZoom: 16,  
        attribution: osmAttrib  
    });
```

En cuanto a este código comentar un par de detalles:

En cuanto a la URL, el parámetro “s” representa subdominios (para cargar de distintos subdominios si el servicio dispone de ellos y aligerar la carga), el parámetro “z” el nivel de zoom, y los parámetros “x” e “y” las coordenadas del “tile”.

La atribución añadirá una nota al pie del mapa indicando el origen de los mismos.

Para crear la capa, se debe indicar la URL del servicio de mapas, y entre las opciones podemos indicar los niveles de zoom máximo y mínimo que queremos permitir.

Con esto ya tendríamos nuestro mapa centrado en el punto que deseamos

## 7 Leaflet vacío

### 7.1 Añadir marcadores a nuestro mapa

Leaflet nos permite añadir marcadores de forma sencilla:

```
L.marker([39.190878,  
-5.7806939999999996]).addTo(demoMap);  
1
```

```
L.marker([39.190878,  
-5.7806939999999996]).addTo(demoMap);
```

Podemos añadir tantos marcadores como queramos a nuestro mapa.

## 7.2 Leaflet Marcadores

### Mostrar un popup con información de un marcador.

Si además de mostrar el marcador, queremos que nos muestre alguna información adicional, podemos emplear un popup para representar esta información en forma de HTML. Para ello podemos refactorizar el código anterior ligeramente

```
var location = [39.190878, -5.7806939999999996];  
var marker = L.marker(location)  
  
marker.bindPopup('<p>Latitud: '+location[0]+'</p><p>Longitud: '+location[1]+'</p>');  
marker.addTo(demoMap);
```

Si ahora hacemos click sobre algún de los marcadores podremos ver la latitud y longitud de dicho marcador

## 7.3 Optimización

En el caso de que necesitemos una mayor fluidez para nuestro mapa (por ejemplo, si es una aplicación web que va a ejecutarse en smartphones con algún año a sus espaldas..), podemos mejorar la experiencia de usuario desac-

ativando las animaciones. Para ello podemos pasarle un objeto con opciones a la llamada de creación del mapa:

```
var demoMap = L.map('map', {
    fadeAnimation: false,
    zoomAnimation: false,
    markerZoomAnimation: false
}).setView([40.453191, -3.509236], 6);
```

También podemos indicarle al “TileLayer” que reutilice tiles, de forma que no tenga que crear nuevos objetos y reservar memoria por cada tile creado, y que se actualice cuando esté en estado “idle” para evitar que se congele la pantalla mientras carga.

```
var osm = new L.TileLayer(osmUrl, {
    minZoom: 5, maxZoom: 16,
    attribution: osmAttrib,
    updateWhenIdle: true,
    reuseTiles: true
});
```

## 8 Clustering

Si necesitamos representar una gran cantidad de marcadores en nuestro mapa, es posible que a un nivel de zoom bajo nuestro mapa se vea sobrecargado. Además el renderizar una gran cantidad de marcadores también es bastante costoso (en nuestro navegador de escritorio probablemente no tendremos problemas, pero en el caso de smartphones la cosa se complica...).

Para ello podemos agrupar los marcadores en clusters, de forma que los marcadores que se aglutinan en un área determinada se representarán como un

único marcadores, y se separarán cuando se alcance un nivel de zoom determinado (o al hacer click, se cambiará el nivel de zoom). Para ello leaflet cuenta con un plugin llamado Leaflet.markercluster. Para instalarlo podemos utilizar npm (en este caso no existe CDN):

```
npm install --save leaflet.markercluster
1
npm install --save leaflet.markercluster
```

Y lo cargaremos en nuestro html:

```
<html>
  <head>
    <title>Demo leaflet.js</title>
    <link rel="stylesheet" href="http://cd-
n.leafletjs.com/leaflet/v0.7.7/leaflet.css" />
    <link rel="stylesheet"
href="node_modules/leaflet.markercluster/dist/Mark-
erCluster.Default.css" />
  </head>
  <body>
    <h1>Leaflet.js Demo</h1>
    <div id="map" style="height: 100%"></div>
    <script type="text/javascript" src="http://cd-
n.leafletjs.com/leaflet/v0.7.7/leaflet.js"></script>
    <script type="text/javascript" src="node_mod-
ules/leaflet.markercluster/dist/leaflet.markerclus-
ter-src.js"></script>
  </body>
</html>
```

Refactorizaremos el código para añadir nuestros marcadores a un cluster. En vez de añadir un marcador directamente al mapa, se debe añadir a una nueva capa:



```
var markerCluster = L.markerClusterGroup();

var location = [39.190878, -5.7806939999999996];
var marker = L.marker(location)

marker.bindPopup('<p>Latitud:'+location[0]+'</p><p>Longitud:'+location[1]+'</p>');
markerCluster.addLayer(marker);

demoMap.addLayer(markerCluster);
```

## 9 – Materiales complementarios

<http://themys.sid.uncu.edu.ar/IDC/>

```
<html>
<head>
<h2>Test de Mapas </h2>
</head>
<body>
```

Links Directos

```
<hr>
```

```
<pre>
```

Todos los links se encuentran anclados a esta sitio:

<http://themys.sid.uncu.edu.ar/IDC/>

En caso de inseguridad o tráfico elevado del DNS se puede referencias como:

<http://200.12.138.4/IDC>

```
[ICO] Name Last modified Size Description
[PARENTDIR] Parent Directory -
[DIR] Chacrasrev1/2017-12-07 11:44 -
[DIR] Guaymallén/ 2017-11-13 15:37 -
[DIR] Las Heras/2017-11-08 23:23 -
[DIR] Luján de Cuyo/ 2017-11-08 23:23 -
[DIR] LujanCarrodillarev1/ 2017-12-07 11:30 -
[DIR] LujanCentrorev1/ 2017-12-07 11:36 -
[DIR] MAIPU/ 2017-11-26 14:52 -
[DIR] Tunuyán/ 2017-11-08 23:23 -
Apache/2.4.18 (Ubuntu) Server at themys.sid.uncu.edu.ar Port 80
```

</pre>

<hr>

<a href="http://themys.sid.uncu.edu.ar/IDC/Chacrasrev1"Chacras de Coria</a>

<p>

<a href="http://themys.sid.uncu.edu.ar/IDC/Guaymallén"Guaymallén</a>

<p>

<a href="http://themys.sid.uncu.edu.ar/IDC/Las Heras"Las Heras</a>

<p>

<a href="http://themys.sid.uncu.edu.ar/IDC/Luján de Cuyo"Lujan de Cuyo</a>

<p>

<a href="http://themys.sid.uncu.edu.ar/IDC/LujanCarrodillarev1"Lujan de Cuyo (Carrodilla)></a>

<p>

<a href="http://themys.sid.uncu.edu.ar/IDC/LujanCentrorev1"Lujan de Cuyo (Centro Revisión 1)></a>

```
<p>
<a
href="http://themys.sid.uncu.edu.ar/IDC/MAIPU"Maip&u
acute></a>
```

```
<p>
<a
href="http://themys.sid.uncu.edu.ar/IDC/Tunuyán"Tunu
y&aacute;n></a>
```

```
<p>
Datos ordenados alfabeticamente
```

```
<hr>
```

```
<strong> Métodos alternativos iframe </strong>
```

```
<p>
```

```
<pre>
```

```
Se incrusta en un frame el mapa de tunuyan con un
marco de760 por 500 pixels de ancho / alto
```

```
<p>
```

```
iframe
```

```
src="http://themys.sid.uncu.edu.ar/IDC/Tunuyán"
width="760" height="500" frameborder="0" margin-
height="0" marginwidth="0">Cargando.../iframe
```

```
<p>
```

```
</pre>
```

```
<iframe
```

```
src="http://themys.sid.uncu.edu.ar/IDC/Tunuyán"
width="760" height="500" frameborder="0" margin-
height="0" marginwidth="0">Cargando...</iframe>
```

```
<hr>
```

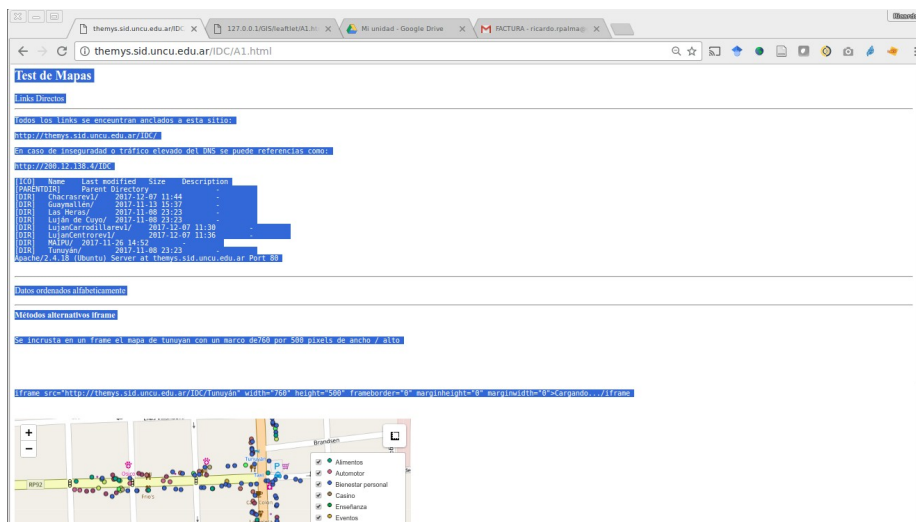
```
<h2>
```

```
acceso al withe paper</h2>
```

```
<a href="http://themys.sid.uncu.edu.ar/IDC/withe
%20paper/"></a>
```

```
</body>
```

```
</html>
```



## 10 - Conclusiones

Leaflet.js es una librería bastante completa. Es bastante intuitiva y su API es bastante simple. Además, nos permite personalizar muchos aspectos, como el icono de los marcadores o el comportamiento de los mismos. También dispone de bastantes plugins como en el caso de los clusters.

El hecho de que podamos utilizar cualquier servicio de mapas que queramos también es un punto a favor, ya que no nos "casamos" con un servicio determinado.

También está preparada para funcionar en smart-phones, reconociendo los gestos típicos como pan o pinch sin tener que programar nada adicional.

Como desventaja, en smartphones con algún año a sus espaldas, o cuyo motor javascript del webview nativo sea algo lento (cof, cof windows phone...), la experiencia de usuario puede ser no todo lo agradable que desearíamos, especialmente cuando cargamos una cantidad elevada de marcadores.

## **11. Referencias**

<http://leafletjs.com/reference.html>

<https://github.com/Leaflet/Leaflet.markercluster>