

Lab 02 Manejo de mapas en páginas web usando OSM (OpenStreetMaps) y Leaflet.

Angelo Ramírez Ortega
e-mail: angramirez@ic-itcr.ac.cr

RESUMEN: El objetivo principal del presente laboratorio es interactuar con nuevas tecnologías web mediante la utilización de APIs para la creación y manipulación de mapas. Se pretende generar dos pruebas distintas utilizando Leaflet, una librería JavaScript codificada encima de OpenStreetMaps, Inicialmente se añadirán puntos en OSM y luego se realizarán pruebas. La primera de las pruebas es hacer una ruta de planificación de un viaje turístico. El segundo es colocar marcadores sobre diferentes puntos y agruparlos para poder modificarlos en conjunto. Se documentarán todas las observaciones realizadas.

PALABRAS CLAVE: OpenStreetMaps, Leaflet, Rutas, APIs

1 INTRODUCCIÓN

OpenStreetMap es una herramienta gratuita de código abierto que permite la utilización de mapas interactivos en páginas web mediante la utilización de tags de HTML y scripts que modifican el DOM de manera dinámica. OSM es mantenido por la comunidad ya que cualquier persona puede hacer acceso a la base de datos y realizar modificaciones para mantenerla actualizada. Hay muchas aplicaciones que se basan sobre OSM que cubren diferentes necesidades, desde la creación de rutas hasta análisis de topografía [1]. Leaflet es una librería de JavaScript codificada sobre OSM que de manera liviana permite generar mapas con puntos, figuras y diferentes "skins" para darle apariencias distintas [2]. Leaflet cuenta también con una serie de plugins que extienden su funcionalidad para poder cumplir con distintos fines. Los dos plugins en los que nos concentraremos son Routing Machine y MarkerCluster. Routing machine permite la creación de rutas a partir de puntos de coordenadas compuestas por longitud y latitud. Encuentra rutas que cubren todos los puntos que se le presenten y puede ser modificado de manera de que los puntos sean interactivos y se puedan mover a través del mapa. Además de esto, genera una ventana con los pasos a seguir desde el inicio del viaje hasta el fin [3]. MarkerCluster cubre la necesidad de manejar grandes cantidades de puntos y agruparlos para poder así interactuar con ellos más sencillamente. Permite modificar tanto las características definitorias de grupos como su apariencia [4].

2 OPENSTREETMAPS

Para poder utilizar OpenStreetMaps lo primero que se debe hacer es crear una cuenta, bajo esta cuenta se registrarán las interacciones que se realicen con el sistema como lo son las ediciones del mapa. Al acceder a la aplicación se presenta un mapamundi en el cual uno puede hacer selecciones de áreas, líneas, generar anotaciones y hacer modificaciones a elementos existentes. Para el presente laboratorio se creó una ruta turística que pasa recogiendo turistas por puntos estratégicos saliendo de Ochomogo, Cartago. Se recogerá a todos, luego se pasará por un restaurante en Jacó centro y por un supermercado para que compren todas las amenidades necesarias. En este punto será necesario visitar una gasolinera y de aquí se pasará a Palo Seco para que los turistas disfruten del sol con un almuerzo en la playa. Finalmente, se les dejará en su Villa privada en las inmediaciones de Parrita. La ruta es la siguiente:

- Plantel de RECOPE, Ochomogo, Cartago.
- Momentum Pinares, Curridabat, San José.
- Plaza del Sol, Curridabat, San José.
- Condominio Benicarlo, Montes De Oca, San José.
- Jacó Walk, Jacó, Puntarenas.
- Mas x Menos, Jacó, Puntarenas.
- Gasolinera Delta, Jacó, Puntarenas.
- Playa Palo Seco, Parrita, Puntarenas.
- Villa Casa Altamar, Parita, Puntarenas.

Se agregaron los puntos que no estaban previamente en el mapa: el Condominio Benicarlo y la Villa Casa Altamar, como se aprecia en la siguiente captura de pantalla. Además de esto se recolectaron las coordenadas de los puntos y se generó un GeoJSON [3] para poder generar las rutas entre los puntos y desplegar su información conforme sea necesario.

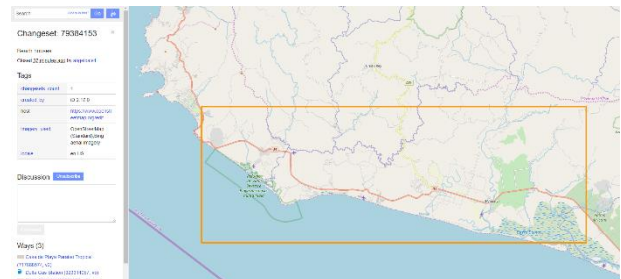


Figura 1. Generación de puntos.

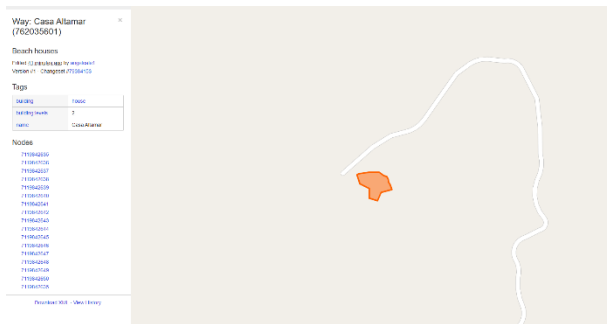


Figura 2. Generación de puntos.

3 LEAFLET

Leaflet nos permite interactuar con los mapas de OpenStreetMap de manera sencilla y ligera. Lo primero que se debe hacer es incluir en la cabecera del HTML los scripts que incluyen los manipuladores del mapa y los estilos básicos. Una vez que hemos incluido esto se requieren tres pasos, crear un tag de <div> en la cual se insertará el mapa, definir su tamaño mediante estilos de CSS y luego generar el mapa mediante JavaScript. Para crear el mapa se utiliza la función `map` del objeto `L` de leaflet. A este le pasamos el id que le colocamos al tag de <div> en el cual colocaremos el mapa. Adicional a esto debemos especificar dónde iniciará el mapa mediante el método `setView` que recibe las coordenadas por utilizar. Una vez que tenemos la estructura del mapa hay que escoger cómo se llenará mediante la utilización de un tile layer. Estos pueden venir de diferentes lugares en línea tanto pagados como gratuitos mediante un URL y definirán la apariencia de nuestro mapa. Una vez tenemos el mapa podemos agregar puntos utilizando marcadores, polígonos y líneas. Podemos agregar a los diferentes elementos popups, estos tienen la finalidad de desplegar información relevante.

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.2.0/dist/leaflet.css" />
<link rel="stylesheet" href="https://unpkg.com/leaflet-routing-machine@lat-est/dist/leaflet-routing-machine.css" />
<script src="https://unpkg.com/leaflet@1.2.0/dist/leaflet.js"></script>
```

Figura 3. Código HTML Leaflet.

```
var mymap = L.map('map').setView([9.7889, -84.2308], 10);
L.tileLayer('http://(s).tile.osm.org/{z}/{x}/{y}.png', {
  attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
}).addTo(mymap);
function onEachFeature(feature, layer) {
  var popupContent = ""
  if (feature.properties && feature.properties.popupContent) {
    popupContent = feature.properties.popupContent;
  }
  layer.bindPopup(popupContent);
}
```

Figura 4. Código Javascript Leaflet.

3.1 PLUGINS

3.1.1 ROUTING MACHINE

El routing machine es un plugin de Leaflet diseñado para extender leaflet con pocas líneas de código y la

capacidad de agregar varios puntos y generar rutas 100% personalizables en tanto funcionamiento y apariencia. Para poder utilizarlo solo debemos incluir una serie de scripts después de incluir Leaflet y generar nuestros puntos cardinales de las posiciones en las que queremos hacer paradas. Para esto utilizaremos el GeoJSON generado previamente a la hora de colocar los puntos que deseábamos de la ruta turística. Iteramos sobre todos estos puntos y generamos la ruta mediante `L.Routing.control` y esto nos generará un mapa con rutas.

```
/*routing*/
L.geoJSON([jsonobj], {
  style: function (feature) {
    return feature.properties && feature.properties.style;
  },
  onEachFeature: onEachFeature,
  pointToLayer: function (feature, latlng) {
    return L.circleMarker(latlng, {
      radius: 8,
      fillColor: "#ff7800",
      color: "#000",
      weight: 1,
      opacity: 1,
      fillOpacity: 0.8
    });
  }
}).addTo(mymap);

var mapWayPoints = [];
for (index = 0; index < jsonobj.features.length; index++) {
  mapWayPoints = mapWayPoints.concat(L.latLng(jsonobj.features[index].geometry.coordinates[1], jsonobj.features[index].geometry.coordinates[0]))
}

L.Routing.control({
  waypoints: mapWayPoints
}).addTo(mymap);
```

Figura 5. Código Javascript Routing Machine.

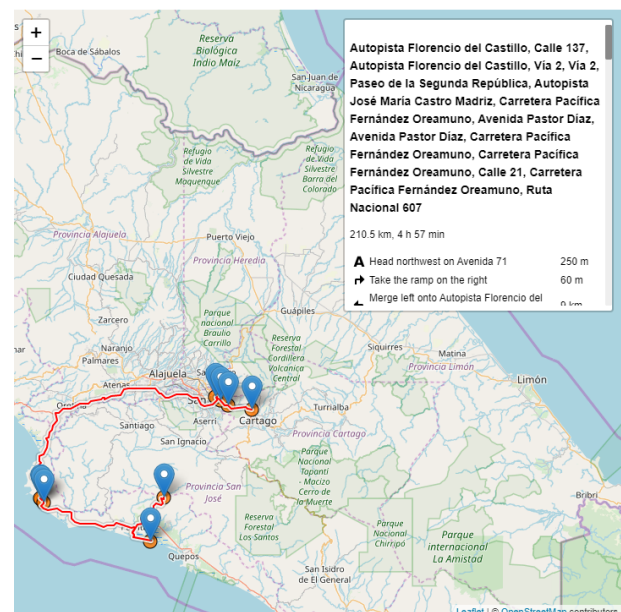
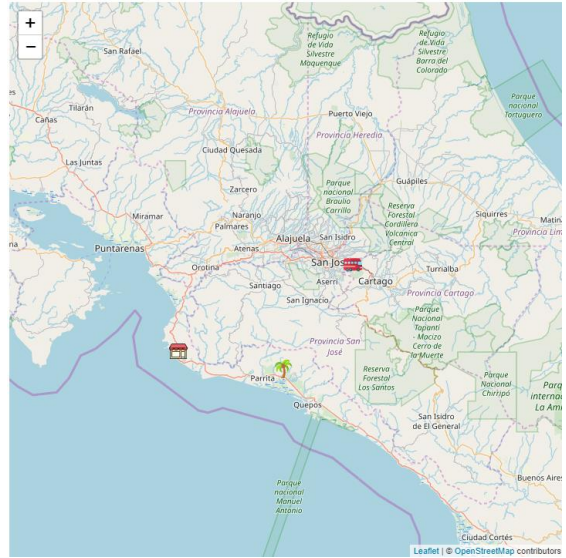


Figura 6. Ruta generada con direcciones.

Nota: las rutas no siempre serán generadas por limitaciones inherentes al servidor por su naturaleza

El `markerCluster` es una librería de JavaScript diseñada para manejar grandes conjuntos de marcadores sobre mapas de manera que sea más sencillo agruparlos por tipos y darles el comportamiento deseado, así como hacerlos más dinámicos a la hora de interactuar con el mapa, especialmente a la hora de hacer zoom tanto hacia adentro como hacia afuera. Para poder utilizarla se descarga el código de la distribución de su GitHub oficial [4], y se sacan los estilos y los scripts y se agregan a la cabecera del HTML. Una vez hecho esto creamos un `markerClusterGroup`, el cual contiene todos los hijos de un grupo. Con solo esto automáticamente ya contamos con grupos al añadirlo al mapa, sin embargo, se tiene la flexibilidad de modificar el código de los marcadores, así como su apariencia mediante estilos y el atributo `iconCreateFunction`. Se generaron tres grupos de marcadores: los puntos de la ciudad, los puntos de jacó y los puntos de la playa. Se les puso tres clases distintas para poder modificar su estilo y colocarles imágenes representativas.



```

var markersRecoGer = L.markerClusterGroup({
    iconCreateFunction: function (cluster) {
        var childCount = cluster.getChildCount();
        var c = ' marker-cluster-recoGer';
        return new L.DivIcon({ html: '<div><span>' + childCount + '</span></div>',
            className: 'marker-cluster' + c, iconSize: new L.Point(40, 40) });
    }
});

var markersJaco = L.markerClusterGroup({
    iconCreateFunction: function (cluster) {
        var childCount = cluster.getChildCount();
        var c = ' marker-cluster-jaco';
        return new L.DivIcon({ html: '<div><span>' + childCount + '</span></div>',
            className: 'marker-cluster' + c, iconSize: new L.Point(40, 40) });
    }
});

var markersPlaya = L.markerClusterGroup({
    iconCreateFunction: function (cluster) {
        var childCount = cluster.getChildCount();
        var c = ' marker-cluster-playa';
        return new L.DivIcon({ html: '<div><span>' + childCount + '</span></div>',
            className: 'marker-cluster' + c, iconSize: new L.Point(40, 40) });
    }
});

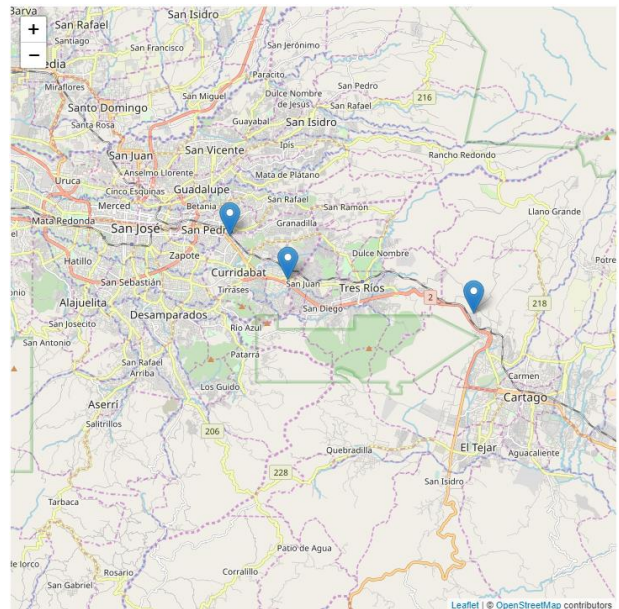
for (var i = 0; i < 3; i++) {
    var marker = L.marker(new L.LatLng(jsonobj.features[i].geometry.coordinates[1], jsonobj.features[i].geometry.coordinates[0]));
    markersRecoGer.addLayer(marker);
}

for (i = 4; i < 7; i++) {
    var marker = L.marker(new L.LatLng(jsonobj.features[i].geometry.coordinates[1], jsonobj.features[i].geometry.coordinates[0]));
    markersJaco.addLayer(marker);
}

for (i = 7; i < jsonobj.features.length; i++) {
    var marker = L.marker(new L.LatLng(jsonobj.features[i].geometry.coordinates[1], jsonobj.features[i].geometry.coordinates[0]));
    markersPlaya.addLayer(marker);
}

map2.addLayer(markersRecoGer);
map2.addLayer(markersJaco);
map2.addLayer(markersPlaya);

```



- [1] OpenStreetMap (2020, Enero 7). OpenStreetMap [En línea]. Disponible en: <https://www.openstreetmap.org/>.
- [2] Leaflet (2020, Enero 7). Leaflet Quick Start Guide [En línea]. Disponible en: <https://leafletjs.com/examples/quick-start/>.
- [3] Leaflet (2020, Enero 7). Using GeoJSON with Leaflet [En línea]. Disponible en: <https://leafletjs.com/examples/quick-start/>.
- [4] Leaflet (2020, Enero 7). Leaflet.markercluster [En línea]. Disponible en: <http://leaflet.github.io/Leaflet.markercluster/>.