

Práctica 12

1. En esta práctica se creará un visualizador web más avanzado con datos en formato GeoJSON.
2. Puede utilizar el siguiente código *ejemplo.html* para crear el mapa:

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Mapa Web</title>
  <link rel="stylesheet"
    href="http://cdn.leafletjs.com/leaflet-0.5/leaflet.css"/>
  <link rel="stylesheet" href="http://netdna.bootstrapcdn.com/font-awesome/
    3.0.2/css/font-awesome.css">

  <!--[if lte IE 8]>
    <link rel="stylesheet" href="http://cdn.leafletjs.com/
      leaflet-0.5/leaflet.ie.css" />

  <![endif]>-->
  <script src="http://cdn.leafletjs.com/leaflet-0.5/leaflet.js"></script>
  <script
    src="https://raw.githubusercontent.com/lvoogdt/Leaflet.awesome-markers/
      master/dist/leaflet.awesome-markers.js"></script>
  <style type="text/css">
    #map { width: 600px; height: 400px }

    .awesome-marker {
      background: url("https://raw.githubusercontent.com/lvoogdt/Leaflet.awesome-markers/
        master/dist/images/markers-soft.png") no-repeat 0 0;
      width: 35px; height: 46px;
      position: absolute;
      left: 0; top: 0;
      display: block;
      text-align: center;
    }

    .awesome-marker-shadow {
      background: url("https://raw.githubusercontent.com/lvoogdt/Leaflet.awesome-markers/
        master/dist/images/markers-shadow.png") no-repeat 0 0;
```

```

        width: 36px; height: 16px;
    }

    .awesome-marker i {
        color: #333; margin-top: 10px;
        display: inline-block; font-size: 14px;
    }

    .awesome-marker .icon-white { color: #fff; }

    .awesome-marker-icon-red { background-position: 0 0; }

    .awesome-marker-icon-darkred { background-position: -180px 0; }

    .awesome-marker-icon-orange { background-position: -36px 0; }

    .awesome-marker-icon-green { background-position: -72px 0; }

    .awesome-marker-icon-darkgreen { background-position: -252px 0; }

    .awesome-marker-icon-blue { background-position: -108px 0; }

    .awesome-marker-icon-darkblue { background-position: -216px 0; }

    .awesome-marker-icon-purple { background-position: -144px 0; }

    .awesome-marker-icon-darkpurple { background-position: -288px 0; }

    .awesome-marker-icon-cadetblue { background-position: -324px 0; }

</style>
</head>
<body>
<div id="map"></div>
<script type="text/javascript">

    var redMarker = L.AwesomeMarkers.icon({
        icon: "food",
        color: "green",
        iconColor: "black"
    });

    function onEachFeature(feature, layer) {
        layer.bindPopup(feature.properties.Nombre);
    };

```

```

function pointToLayer(feature, latlng) {
    return L.marker(latlng, {icon: redMarker});
};

var xhr = new XMLHttpRequest() || new ActiveXObject("Msxml2.XMLHTTP")
    || new ActiveXObject("Microsoft.XMLHTTP");
xhr.open("GET","puntos.geojson",false); xhr.send(null);
var puntos = JSON.parse(xhr.responseText);

xhr.open("GET","lineas.geojson",false); xhr.send(null);
var lineas = JSON.parse(xhr.responseText);

xhr.open("GET","poligonos.geojson",false); xhr.send(null);
var poligonos = JSON.parse(xhr.responseText);

var styleLineas = {
    "color": "#999999",
    "weight": 2,
    "opacity": 0.65
};

var stylePoligonos = {
    "color": "#ff7800",
    "weight": 1,
    "opacity": 0.65
};

var puntosLayer = L.geoJson(puntos, {
    pointToLayer: pointToLayer, onEachFeature: onEachFeature
});

var lineasLayer = L.geoJson(lineas, {
    style: styleLineas
});

var poligonosLayer = L.geoJson(poligonos, {
    style: stylePoligonos
});

var map = L.map("map", {
    center: new L.LatLng(0.450,-0.450),
    zoom: 8,
    layers: [puntosLayer, lineasLayer, poligonosLayer]
});

var overlayMaps = {

```

```

        "Restaurantes": puntosLayer,
        "Calles": lineasLayer,
        "Edificios": poligonosLayer
    };

    L.control.layers(null, overlayMaps).addTo(map);
</script>
</body>
</html>

```

3. El archivo de datos *puntos.geojson* utilizado por este ejemplo, y tal como lo genera *QGIS*, es el siguiente:

```

{ "type": "FeatureCollection", "features": [
  { "type": "Feature", "id": 0, "properties": {
    "id": 1, "Nombre": "McDonald's" },
    "geometry": { "type": "Point", "coordinates":
      [ -0.646113458359572, 0.611079915774559 ] } }
  ,
  { "type": "Feature", "id": 1, "properties": {
    "id": 2, "Nombre": "Taco Bell" },
    "geometry": { "type": "Point", "coordinates":
      [ 0.06295268222607, 0.311090394757557 ] } }
  ,
  { "type": "Feature", "id": 2, "properties": {
    "id": 3, "Nombre": "Pizza Hut" },
    "geometry": { "type": "Point", "coordinates":
      [ -0.818834697732997, 0.180185876495592 ] } }
  ]
}

```

4. El archivo de datos *lineas.geojson* utilizado por este ejemplo, y tal como lo genera *QGIS*, es el siguiente:

```

{ "type": "FeatureCollection", "features": [
  { "type": "Feature", "id": 0, "properties": {
    "id": 1, "nombre": "Avenida 1" }, "geometry":
    { "type": "LineString", "coordinates":
      [ [ -1.258819328557935, 0.440176794710327 ],
        [ -0.726110663964106, 0.425631848236776 ],
        [ -0.146130923331234, 0.381997008816121 ], [
          0.366578439861461, 0.212912006061083 ] ] } }
  ,
  { "type": "Feature", "id": 1, "properties": {
    "id": 2, "nombre": "Calle 1" }, "geometry":
    { "type": "LineString", "coordinates":

```

```

[ [ -0.444302326039043, 1.100153740947733 ],
[ -0.724292545654912, 0.423813729927582 ],
[ -0.949739215994962, -0.188892140270781 ] ] } }
,
{ "type": "Feature", "id": 2, "properties":
{ "id": 3, "nombre": "Calle 3" }, "geometry":
{ "type": "LineString", "coordinates":
[ [ -0.046134416325567, 1.083790676164987 ],
[ -0.186129526133501, 0.520174000314861 ],
[ -0.146130923331234, 0.380178890506927 ],
[ -0.053406889562343, -0.267071227566121 ] ] } }
]}

```

5. El archivo de datos *poligonos.geojson* utilizado por este ejemplo, y tal como lo genera *QGIS*, es el siguiente:

```

{ "type": "FeatureCollection", "features": [
{ "type": "Feature", "id": 0, "properties": { "id": 1 },
"geometry": { "type": "Polygon", "coordinates":
[ [ [ -1.205859375, 0.5775 ], [ -1.117265625, 1.10578125 ],
[ -1.097578125, 1.10578125 ], [ -1.097578125, 1.10578125 ],
[ -0.533203125, 1.07296875 ], [ -0.795703125, 0.55453125 ],
[ -1.205859375, 0.5775 ] ] ] } },
{ "type": "Feature", "id": 1, "properties": { "id": 2 },
"geometry": { "type": "Polygon", "coordinates":
[ [ [ -0.018046875, 1.0040625 ], [ 0.411796875, 1.010625 ],
[ 0.401953125, 0.4725 ], [ -0.119765625, 0.58078125 ],
[ -0.018046875, 1.0040625 ] ] ] } },
{ "type": "Feature", "id": 2, "properties": { "id": 3 },
"geometry": { "type": "Polygon", "coordinates":
[ [ [ -0.693984375, 0.29859375 ], [ -0.467578125, 0.275625 ],
[ -0.155859375, 0.18703125 ], [ -0.132890625, -0.2690625 ],
[ -0.874453125, -0.19359375 ], [ -0.693984375, 0.29859375 ]
] ] } }
] ] } }

```

6. Dibuje, mediante *QGIS*, los mismos elementos que se presentaron en el mapa de la ciudad de la práctica anterior (incluyendo datos puntuales). Incluya en la base de datos los nombres de restaurantes, las calles y avenidas, e invente algunos nombres para los elementos poligonales (plaza, iglesia, banco, etc.)
7. Exporte desde *QGIS* las capas creadas (restaurantes, calles, avenidas, quebradas y edificios) hacia formato *GeoJSON*.
8. Modifique el código del mapa web para que incluya las elementos que usted creó junto con el selector de capas. Debe aparecer la etiqueta de cada

elemento cuando esta se selecciona con el cursor. También debe cambiar los estilos de colores para cada clase de elementos, así como los iconos de los restaurantes.

9. Publique su código en Github de forma que se pueda visualizar el mapa sin tener que descargarlo.

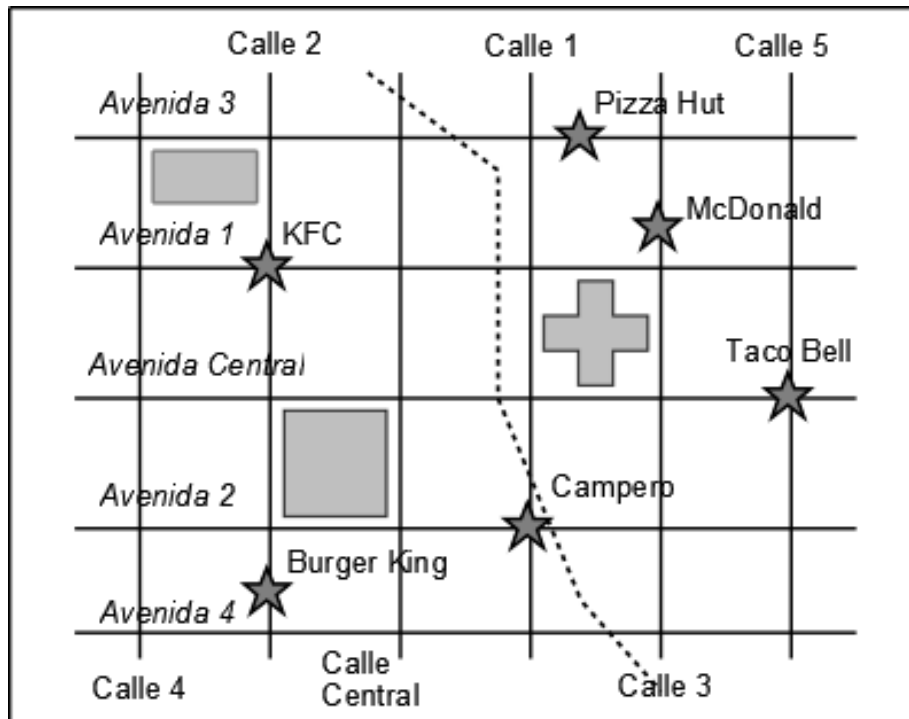


Figure 1: Mapa de ciudad