

# Programación WEB (PWEB)

## Tema 4 OpenLayers v2 y v3

Miguel Ángel Manso  
Ramón Alcarria



1

## Contenido

- Introducción
- OpenLayers v3:
  - Referencias y Documentación
  - Primer mapa
- OpenLayers v2
  - Referencias y Documentación
  - Primer mapa

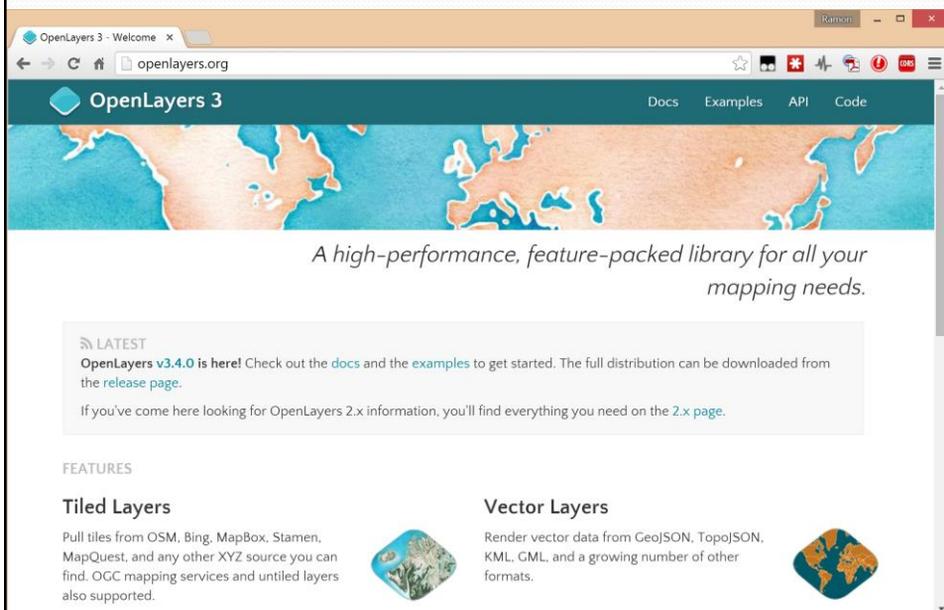
# Introducción

- OpenLayers es una librería JavaScript de código abierto para mostrar mapas interactivos en los navegadores web. Licencia derivación de BSD
- OpenLayers ofrece un API para acceder a diferentes fuentes de información cartográfica en la red: Web Map Services, Mapas comerciales (Google Maps, Bing, Yahoo), OSM, Web Feature Services, distintos formatos vectoriales.
- Desde noviembre del 2007 el proyecto forma parte de los proyectos de Open Source Geospatial Foundation
- Web oficial: <http://www.openlayers.org/>

PWEB 2014-2015

3

# Introducción



OpenLayers 3 - Welcome x

openlayers.org

OpenLayers 3 Docs Examples API Code

*A high-performance, feature-packed library for all your mapping needs.*

**LATEST**  
**OpenLayers v3.4.0 is here!** Check out the [docs](#) and the [examples](#) to get started. The full distribution can be downloaded from the [release page](#).  
 If you've come here looking for OpenLayers 2.x information, you'll find everything you need on the [2.x page](#).

**FEATURES**

**Tiled Layers**  
 Pull tiles from OSM, Bing, MapBox, Stamen, MapQuest, and any other XYZ source you can find. OGC mapping services and untiled layers also supported.

**Vector Layers**  
 Render vector data from GeoJSON, TopoJSON, KML, GML, and a growing number of other formats.

# OL3: Referencias y docs.

- Documentacion General OpenLayers:
  - <http://trac.osgeo.org/openlayers/wiki/Documentation>
- Ejemplos con código fuente:
  - <http://openlayers.org/en/v3.4.0/examples/>
- Referencia (API Reference)
  - <http://openlayers.org/en/v3.4.0/apidoc/>
- Como crear un mapa con OpenLayers 3:
  - <http://mappinggis.com/2013/04/como-crear-un-mapa-con-openlayers-3/>
- Antes de iniciar:
  - Conocimientos de Javascript
  - jQuery
  - JSON y AJAX

PWEB 2014-2015

5

# API Reference

The screenshot shows the OpenLayers 3 API Documentation website. The browser address bar displays `openlayers.org/en/v3.4.0/apidoc/ol.Map.html`. The page title is "OpenLayers 3 API Documentation" with a "Stable Only" indicator. A search bar is visible at the top left. The main content area is titled "ol.Map" and includes a description: "The map is the core component of OpenLayers. For a map to render, a view, one or more layers, and a target container are needed:". Below the description is a code snippet for creating a map:

```
var map = new ol.Map({
  view: new ol.View({
    center: [0, 0],
    zoom: 1
  }),
  layers: [
    new ol.layer.Tile({
      source: new ol.source.MapQuest({layer: 'osm'})
    })
  ],
  target: 'map'
});
```

The text below the code explains: "The above snippet creates a map using a `ol.layer.Tile` to display `ol.source.MapQuest` OSM data and render it to a DOM element with the id `map`." It also states: "The constructor places a viewport container (with CSS class name `ol-viewport`) in the target element (see `getViewport()`), and then two further elements within the viewport: one with CSS class name `ol-overlaycontainer-stopevent` for controls and some overlays, and one with CSS class name `ol-overlaycontainer` for other overlays (see the `stopEvent` option of `ol.Overlay` for the difference). The map itself is placed in a further element within the viewport, either DOM or Canvas, depending on the renderer."

At the bottom of the page, it says "new ol.Map(options)" and "src/ol/map.js, line 162".

## Creando el primer mapa

- <http://openlayers.org/en/v3.0.0/doc/quickstart.html>
- La API de OpenLayers tiene dos conceptos que son importantes para entender cómo construir el primer Visor: '**Map**' y '**Layer**'
- **Map** almacena la información del *canvas*: proyección, extensión geográfica, tamaño, unidades, máximas escalas, zooms..
- Sobre el mapa se visualizan las capas '**Layer**'
- Cada **Layer** es una fuente de datos (ráster, vector), y contiene la lógica particular de cada fuente de información (WMS, WFS, Google, Bing, Bing, ..., KML)

## Creando el primer mapa

- Los pasos a seguir para la creación paso a paso del primer mapa empleando OpenLayers son los siguientes:
  - 1) Incluir las librerías de OpenLayers
  - 2) Etiqueta HTML para el Mapa
  - 3) Definición de elemento Mapa
  - 4) Añadir control de Zoom
  - 5) Añadir otras capas
  - 6) Añadir elementos externos a OL

## Creando el primer mapa

- **Paso 1)** Incluir las librerías OpenLayers
- La construcción de un visor OpenLayers requiere la elaboración de HTML en el que se incluirá el visor
- OpenLayers soporta el posicionamiento de un mapa en el interior de cualquier elemento de nivel bloque - lo que significa que se puede utilizar casi cualquier elemento del código HTML de la página para poner un mapa
- Además de un elemento de nivel bloque, se requiere también incluir una etiqueta de `<script>` que incorpore la biblioteca OpenLayers a la página.

## Creando el primer mapa

- **Paso 1)** Incluir las librerías OpenLayers

```
<link rel="stylesheet"
href="http://openlayers.org/en/v3.2.0/css/ol.css"
type="text/css">
```

```
<script
src="http://openlayers.org/en/v3.2.0/build/ol.js"
type="text/javascript"></script>
```

## Creando el primer mapa

- **Paso 2)** Etiqueta HTML para el Mapa
- Dentro del `<body>` tenemos que crear un elemento HTML para que contenga el Mapa

```
<div style="width:100%; height:100%" id="map"></div>
```

## Creando el primer mapa

- **Paso 3)** Definición de elemento Mapa
- Debe crearse un elemento Map

```
var map = new ol.Map({
  layers: [
    new ol.layer.Tile({
      source: new ol.source.OSM()
    })
  ],
  renderer: 'canvas',
  target: 'map',
  view: new ol.View({
    center: [0, 0],
    zoom: 2
  })
});
```

- El constructor **ol.Map** admite varios argumentos: layers, renderer, target, view, etc

## Creando el primer mapa

- **Paso 4)** Añadimos control de Zoom

```
map.addControl(new ol.control.ZoomSlider());
```

## Creando el primer mapa

- **Paso 5)** Añadimos otras capas, por ejemplo de un WMS

```
var wms = new ol.layer.Tile({  
  title: "Satélite global",  
  source: new ol.source.TileWMS({  
    url: 'http://maps.opengeo.org/geowebcache/service/wms',  
    params: {LAYERS: 'blumarble', VERSION: '1.1.1'}  
  })  
});  
  
map.addLayer(wms);
```

## Creando el primer mapa

- **Paso 6)** Añadimos un layer switcher
- En OpenLayers v2 existe un layer switcher oficial, para OpenLayers v3, como es nuevo, hay que usar layer\_switcher creado por desarrolladores externos:
- <https://github.com/walkermatt/ol3-layerswitcher>

```
var layerSwitcher = new ol.control.LayerSwitcher({
  tipLabel: 'Capas' // Optional label for button
});
map.addControl(layerSwitcher);
```

## Ejercicio

- **Añadimos la capa: *Population density***
- Dirección del WMS:
  - <http://sedac.ciesin.columbia.edu/geoserver/wms>
- Nombre de la capa:
  - gpw-v3:gpw-v3-population-density\_2000

## Tipos de capas

- **ol.layer.Tile** is for layer sources that provide pre-rendered, tiled images in grids that are organized by zoom levels for specific resolutions.
- **ol.layer.Image** is for server rendered images that are available for arbitrary extents and resolutions.
- **ol.layer.Vector** is for vector data that is rendered client-side.

## OL2: Referencias y docs.

- Documentación General OpenLayers v2:
  - <http://openlayers.org/two/>
- Ejemplos con código fuente:
  - <http://dev.openlayers.org/examples/>
- Referencia (API Reference)
  - <http://dev.openlayers.org/releases/OpenLayers-2.13.1/doc/apidocs/files/OpenLayers-js.html>
- Manual de OpenLayers v2 en Español:
  - <http://openlayers.bicimap.es/manualOpenLayers.html>

## Creando primer mapa en OL2

- Paso 1) Buscar librería de OpenLayers (2.13.1)
  - <http://cdnjs.com/libraries/openlayers>
- Encontramos:
  - <https://cdnjs.cloudflare.com/ajax/libs/openlayers/2.13.1/OpenLayers.js>

## Creando primer mapa en OL2

- Paso 2) Escribimos aplicación básica

```
<html> <head><title>OpenLayers Simplest Example</title></head>
  <body>
    <div id="demoMap" style="height:800px"/>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/openlayers/2.13.1/OpenLayers.js"/>
    <script>
map = new OpenLayers.Map("demoMap");
map.addLayer(new OpenLayers.Layer.OSM());
map.zoomToMaxExtent();
    </script>
  </body> </html>
```

## Creando primer mapa en OL2

- Paso 3) Añadimos otros elementos
  - Transformación de coordenadas

```
var fromProjection = new OpenLayers.Projection("EPSG:4326");
// Transforma coordenadas desde WGS84
var toProjection = new OpenLayers.Projection("EPSG:900913");
// a Spherical Mercator

var position = new OpenLayers.LonLat(13.41, 52.52).transform(
  fromProjection, toProjection);

var zoom = 15;
```

## Creando primer mapa en OL2

- Paso 3) Añadimos otros elementos
  - Más controles

```
var options = {
  restrictedExtent : extent,
  controls: [
    new OpenLayers.Control.Navigation(),
    new OpenLayers.Control.PanZoomBar(),
    new OpenLayers.Control.Attribution()
  ]
};
map = new OpenLayers.Map("Map", options);
```

## Creando primer mapa en OL2

- Paso 3) Añadimos otros elementos
- ¿Cómo añadimos un selector de capas?
  - Ver Layerswitcher:
  - <http://dev.openlayers.org/docs/files/OpenLayers/Control-js.html#OpenLayers.Control>
- ¿Cómo añadimos otra capa WMS?
  - Ver OpenLayers.Layer.WMS

## Creando primer mapa en OL2

- Paso 3) Añadimos otros elementos
- Añadimos geometrías
  - Geometría Punto, con latitud y logitud

```
var vectorLayer = new OpenLayers.Layer.Vector ("Geometrías");
```

```
var feature = new OpenLayers.Feature.Vector (new
OpenLayers.Geometry.Point(-3.72,40.44).transform( fromProjection,
toProjection), {some: "dato interesante"});
```

```
vectorLayer.addFeatures (feature);
```

## Creando primer mapa en OL2

- Paso 3) Añadimos otros elementos
- Añadimos un marcador

```
var markers = new OpenLayers.Layer.Markers( "Marcadores" );  
  
markers.addMarker(new OpenLayers.Marker(new  
OpenLayers.LonLat(-3.74,40.46).transform( fromProjection,  
toProjection)));
```