Ajuntament de Barcelona

Diseña mapas interactivos con OpenLayers.

Día 1 de 4, Modulo 2h, 2021

Barcelonactiva

Barcelona Activa: Qui som?

Barcelona Activa, integrada en l'àrea d'Economia, Empresa i Ocupació, és l'organització executora de les polítiques de promoció econòmica de l'Ajuntament de Barcelona.

Des de fa 25 anys impulsa el creixement econòmic de Barcelona i el seu àmbit d'influència donant suport a les empreses, la iniciativa emprenedora i l'ocupació, alhora que promociona la ciutat internacionalment i els seus sectors estratègics; en clau de proximitat al territori.

Barcelona Activa va ser guanyadora del Gran Premi del Jurat 2011, atorgat per la DG d'Empresa i Indústria de la Comissió Europea en el marc dels *European Enterprise Awards*, per la iniciativa empresarial més creativa i inspiradora d'Europa.

Àrees d'activitat de Barcelona Activa

Barcelona Activa s'estructura en tres grans blocs de serveis a les **Empreses**, a l'**Emprenedoria** i a la **Ocupació**. La **Formació** és un instrument transversal present en els tres blocs, així com també tot el relacionat amb l'economia social.

Una xarxa d'Equipaments Especialitzats

Xarxa de Proximitat

13 antenes Cibernàrium a biblioteques
10 punts d'atenció en Ocupació

Índice

- **Día 1: Conceptos básicos y primeros mapas con capas ráster**
- Día 2: Capas vectoriales de GeoJSON y TopoJSON y su diseño
- Día 3: Controles, estilos dinámicos, eventos, animaciones y capas WMS
- Día 4: Overlays, proyecciones, plugins y teselas vectoriales

Índice día 1

- Presentación
- Generación de mapas desde QGIS
- Introducción OpenLayers
- Primer mapa con OpenLayers usando OpenStreetMap
- Pirámide de teselas
- Sistemas de coordinadas
- Tipos de capas
- Capas ráster

Proyectos hechos con OpenLayers

Algunos proyectos personales hechos con OpenLayers:
http://geraldo.github.io/?tag=Openlayers

https://tracesmap.org/

http://cartahistorica.muhba.cat/

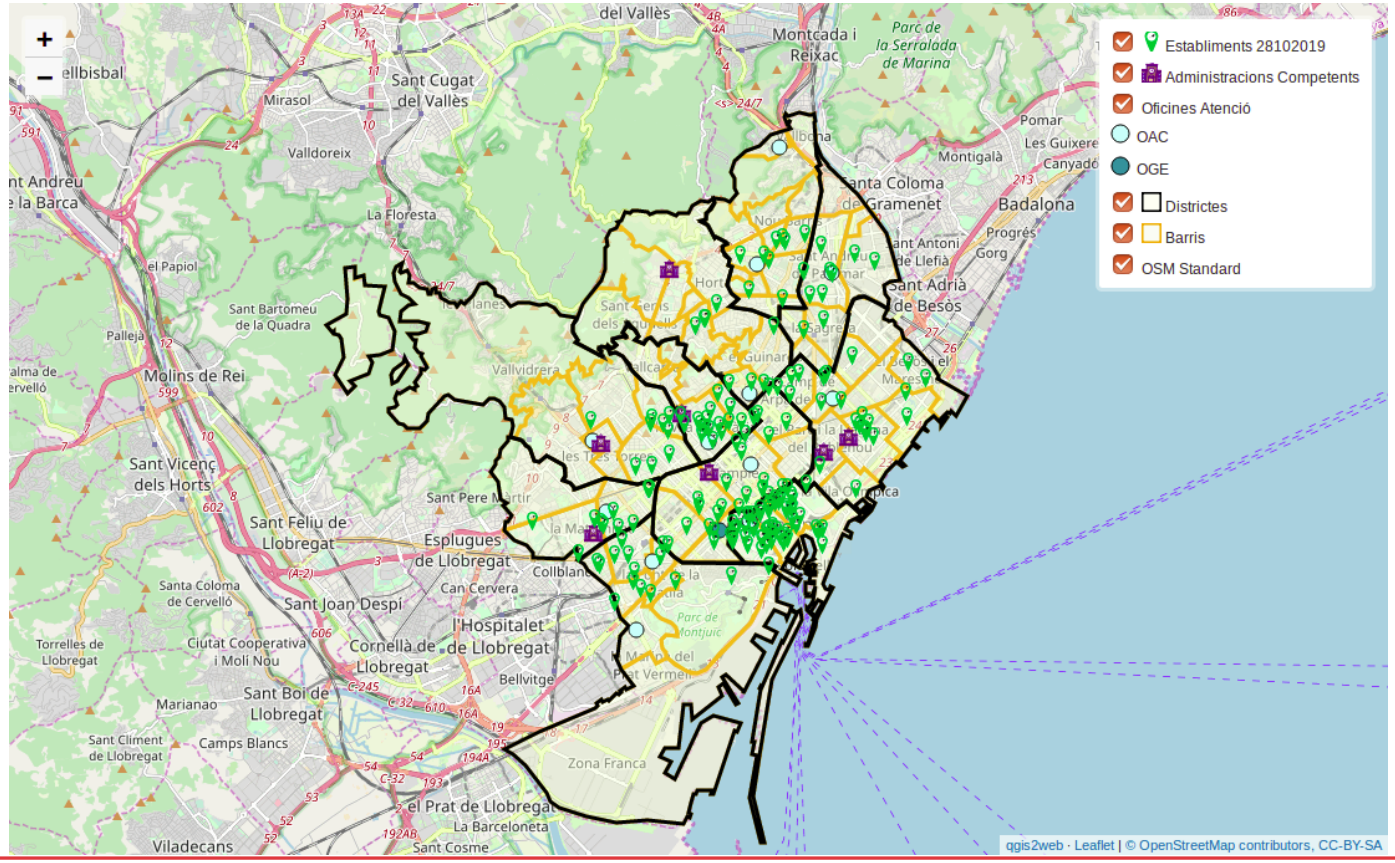
Mapas OpenLayers (casi) sin tocar código

Comparación soluciones de exportación desde QGIS

- qgis2web
- QGIS Cloud
- Lizmap
- g3wsuite
- GeoNode

Ejemplo real de mapa con qgis2web

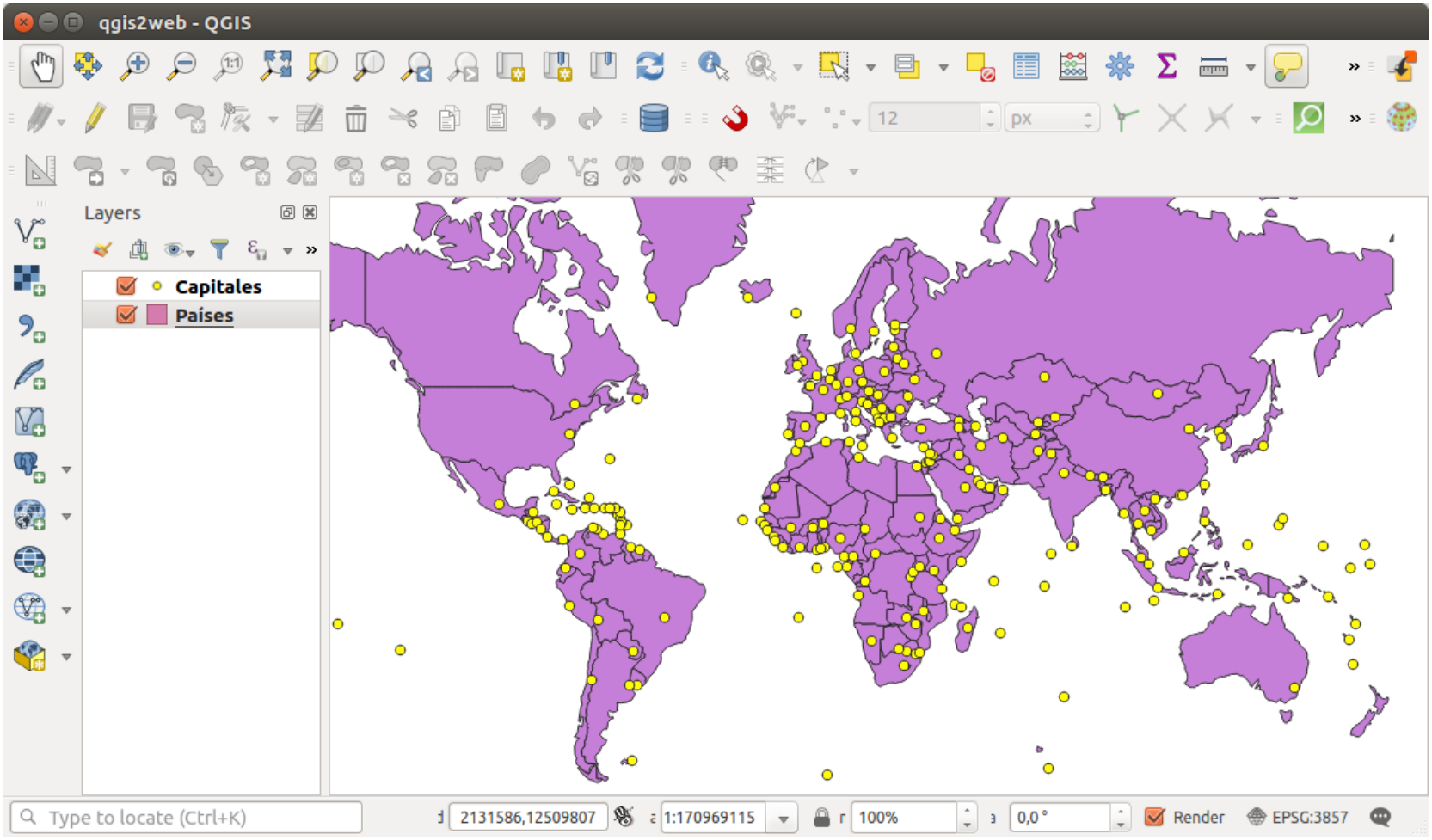
Agència de Salut Pública de Barcelona: Cens i mapa d'establiments de Tatuatges



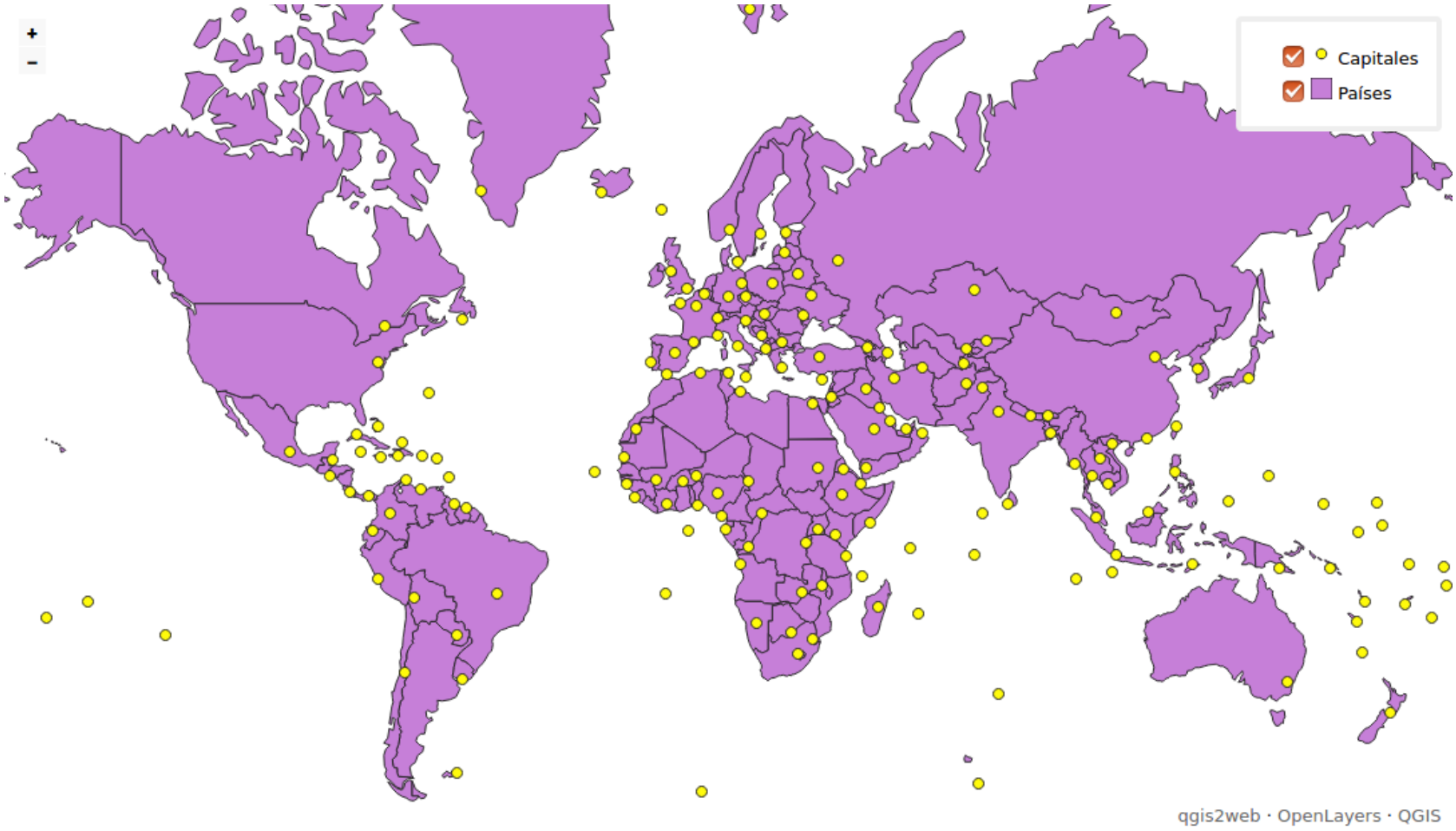
Primer mapa con qgis2web

1. Abrir QGIS
2. Instalar qgis2web: *Plugins* > *Manage and Install plugins…* > *qgis2web* (en el caso que no se muestra el plugin, hace falta activar opción *Show also experimental plugins* en *Settings*
3. Descomprimir y abrir proyecto QGIS *qgis2web.zip*
4. Exportar mapa menú *Web* > *qgis2web* > *Create web map*
5. Cambiar vista previa a OpenLayers (las otras opciones son Leaflet y Mapbox GL JS) y clickar *Update Preview*
6. Ajustar opciones:
   - *Layers and Groups*: Marcar *Visible* y *Popup* para *capitals* y *countries*
   - *Appearance*: Se puede limitar nivel de zoom, cambiar colores de capas y íconos, añadir herramientas, etc.
   - *Export*: Seleccionar carpeta donde exportar contenido

Primer mapa con qgis2web



Primer mapa con qgis2web



Características principales de OpenLayers

- Web: https://openlayers.org/
- Software libre con licencia BSD desde 2006
- Ahora bajo el paraguas de OSGeo
- Gran comunidad de desarrolladores y usuarios
- Excelente documentación de la API
- Muchos ejemplos online

Características principales de OpenLayers

- Proveedores de mapas como OSM, Bing, MapBox y Stamen preconfiguradas
- Capas ráster en formato XYZ
- Servicios OGC como WMS y WFS
- Capas vectoriales como GeoJSON, TopoJSON, KML, GML, Mapbox vector tiles y otros
- Rendering para Canvas 2D (por definición), WebGL y adaptada para todas las tecnologías de HTML5 y CSS3
- Gran soporte para proyecciones usando códigos EPSG
- Sistema modular para crear una librería ligera y rápida

Primer mapa Openlayers: holamundo.html

```
<!doctype html>
<html lang="en">
  <head>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master/en/v6.5.0/css/ol.css" type="text/css">
    <style>
      .map {
        height: 600px;
        width: 100%;
      }
    </style>
    <script src="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master/en/v6.5.0/build/ol.js"></script>
    <title>Hola mundo, hola OpenLayers</title>
  </head>
  <body>
    <h2>Cataluña en OpenStreetMap</h2>
    <div id="map" class="map"></div>
    <script type="text/javascript">
      var map = new ol.Map({
        target: 'map',
        layers: [
          new ol.layer.Tile({
            source: new ol.source.OSM()
          })
        ],
        view: new ol.View({
          center: ol.proj.fromLonLat([1.466667, 41.816667]),
          zoom: 8
        })
      });
    </script>
  </body>
</html>
```

Primer mapa de Openlayers con OSM

Basado en ejemplo Quickstart de OpenLayers

Conceptos básicos
= Clases de OpenLayers:

- Map
- View
- Source
- Layer

Primer mapa de Openlayers con OSM

Para crear un mapa en una página web usando OpenLayers hacen falta 3 cosas:

1. Incluir librería de Javascript OpenLayers
2. Contenedor mapa `<div>`
3. JavaScript para crear un mapa sencillo

1. Incluir librería de JavaScript OpenLayers

```
<script
src="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master/en/v6
.5.0/build/ol.js"></script>
```

Primero tienes que incluir la librería de JavaScript de OpenLayers. Para esta clase simplemente vamos a incluir una versión de OpenLayers online. En un entorno de producción incluiríamos una versión customizada conteniendo solamente los módulos necesarios para la aplicación.

```
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master/en/v
6.5.0/css/ol.css" type="text/css">
```

Además incluimos el la parte CSS de la librería, con eso damos estilos a los elementos básicos del mapa como por ejemplo los botones Zoom + y -.

2. Contenedor mapa

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

El mapa de la aplicación se muestra en un contenedor de HTML. A través de este se puede controlar propiedades como anchura, altura y marco usando CSS. Aquí el código CSS para crear un mapa con una altura de 600px y una anchura del 100% de la ventana del navegador.

```
<style>
  .map {
    height: 600px;
    width: 100%;
  }
</style>
```

3. JavaScript para crear un mapa

```
var map = new ol.Map({
  target: 'map',
  layers: [
    new ol.layer.Tile({
      source: new ol.source.OSM()
    })
  ],
  view: new ol.View({
    center: ol.proj.fromLonLat([1.466667, 41.816667]),
    zoom: 8
  })
});
```

Con este código creamos un mapa con una capa de OpenStreetMap mostrando Cataluña. Vamos paso por paso:

3. JavaScript para crear un mapa

```
var map = new ol.Map({ ... });
```

Usamos el parámetro `target` para añadir este objeto mapa al elemento . El valor es la `id` del `<div>`:

```
target: 'map'
```

`layers: [ ... ]` define un array de capas para el mapa. En este ejemplo usamos una única capa, cual es una capa teselada:

```
layers: [
  new ol.layer.Tile({
    source: new ol.source.OSM()
  })
]
```

Las capas en OpenLayers se definen con un tipo (Image, Tile o Vector) cual contiene una fuente. La fuente es el protocolo para conseguir los teselas del mapa.

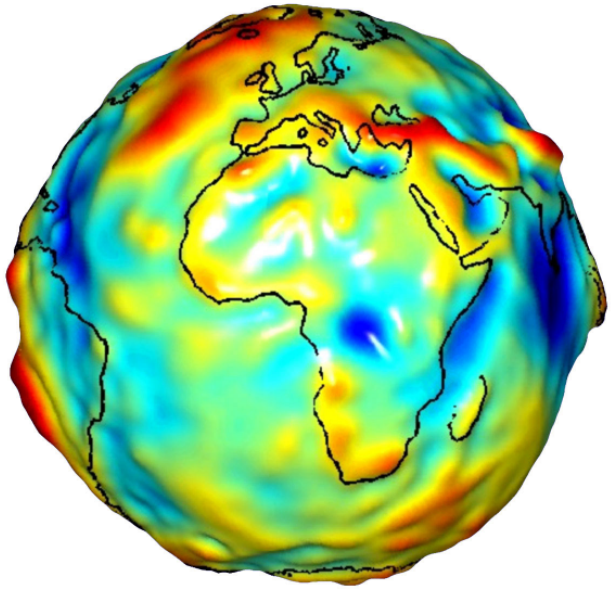
3. JavaScript para crear un mapa

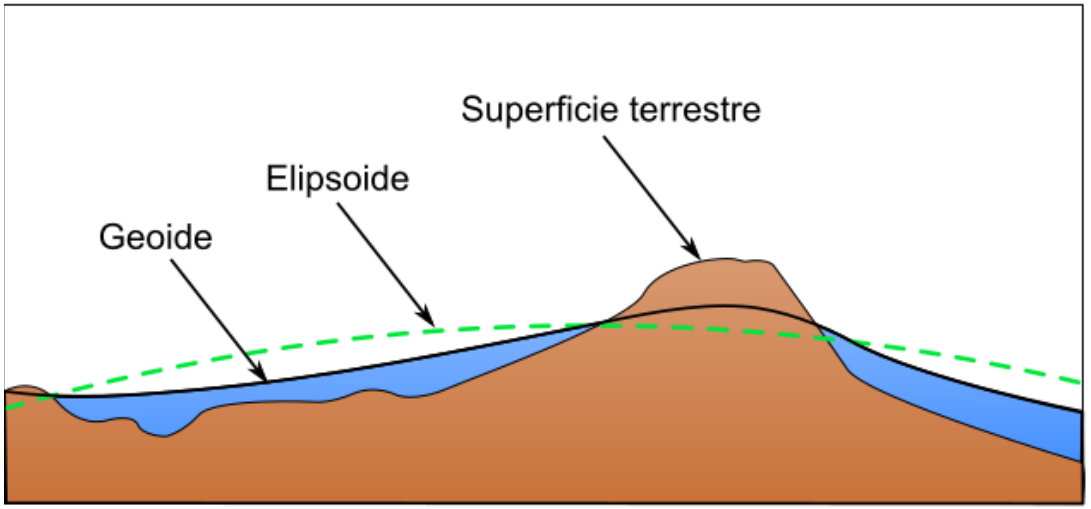
La siguiente parte del mapa es la vista, llamada *View*. La vista permite definir el centro, la resolución y la rotación del mapa. La forma más sencilla de definir una vista es poniendo un punto central y un nivel de zoom. Ten en cuenta que nivel 0 de zoom es zoomed out.

```
view: new ol.View({
  center: ol.proj.fromLonLat([1.466667, 41.816667]),
  zoom: 8
})
```

Notarás que el centro del mapa esta definido en coordinadas de longitud y latitude (EPSG:4326). Como la única capa que usamos esta en la proyección esférica de Mercator (Spherical Mercator projection, EPSG:3857), podemos reproyectarlo en el momento para poder hacer zoom del mapa en los coordinadas correctos, lo que en este caso es el centro de Cataluña.

Sistemas de coordinadas
Geodesia





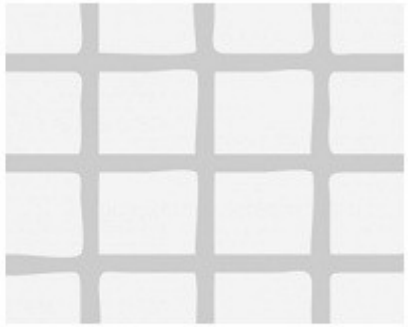
Victor Olaya: Sistemas de Información Geográfica
→ https://volaya.github.io/libro-sig/chapters/Fundamentos_cartograficos.html

Sistemas de coordinadas
Geográfico: Mercator

```
const Catalunya = ol.proj.fromLonLat([1.466667, 41.816667]);
```

Las coordinadas usadas en este ejemplo están en WGS84 (EPSG:4326), que es un sistema de coordinadas geográfico con longitud y latitud expresadas en grados decimales, como por ejemplo también utilizado en sistemas de GPS.

Geographic (3D)



Projected (2D)

Sistemas de coordinadas
Proyectado: Web Mercator

OpenLayers por definición usa coordinadas en proyección Web Mercator, o también llamado Pseudo Mercator (Spherical Mercator, EPSG:3857) con metros como unidades del mapa. Es un sistema de coordinadas proyectado.

Por lo tanto, otra forma de usar una coordinada sería:

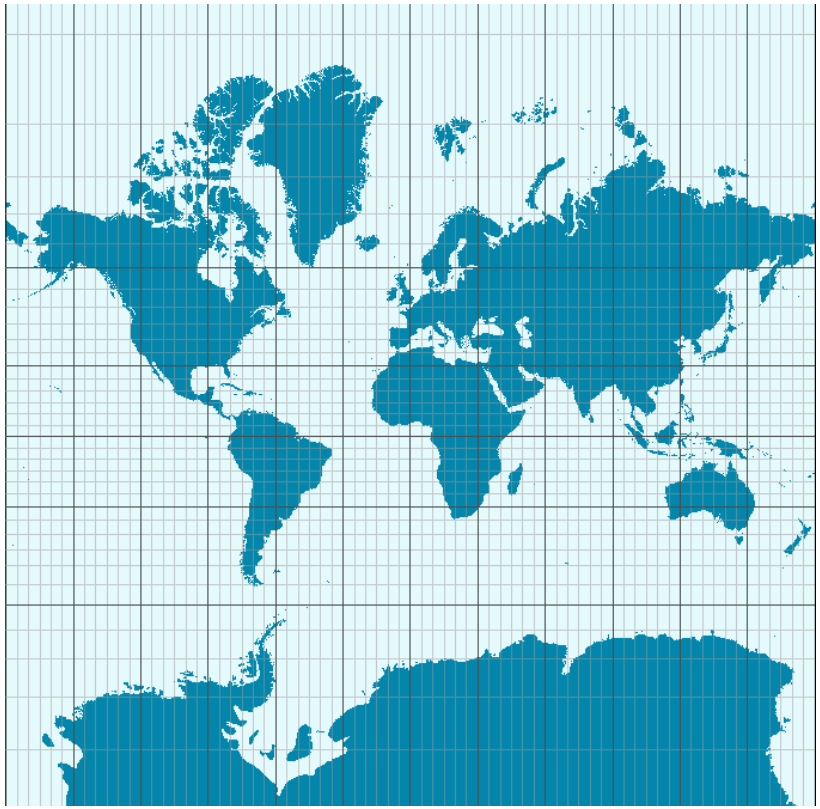
```
const Catalunya = [163268.62360329815, 5133556.487942288];
```

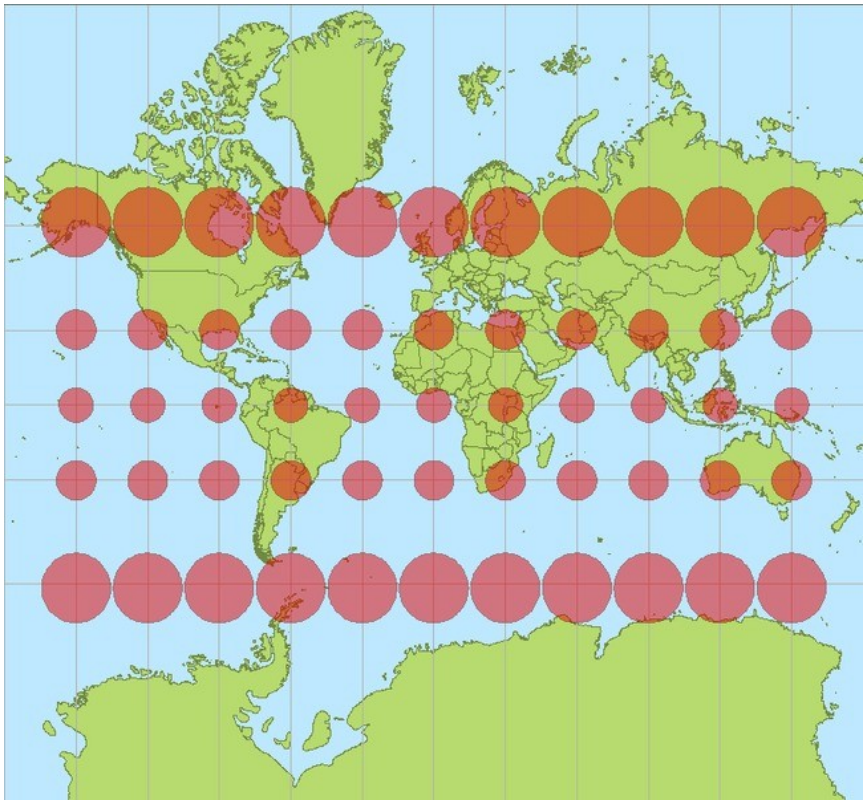
En todo caso hay que destacar que OpenLayers tiene un soporte muy sofisticado a todo tipo de proyecciones, más adelante profundizaremos en el tema.

→ Aquí un buen artículo para definir las diferencias entre EPSG 4326 y EPSG 3857.

→ Aquí un ejemplo que muestra las coordinadas en los 2 sistemas:
https://openlayers.org/en/latest/examples/mouse-position.html

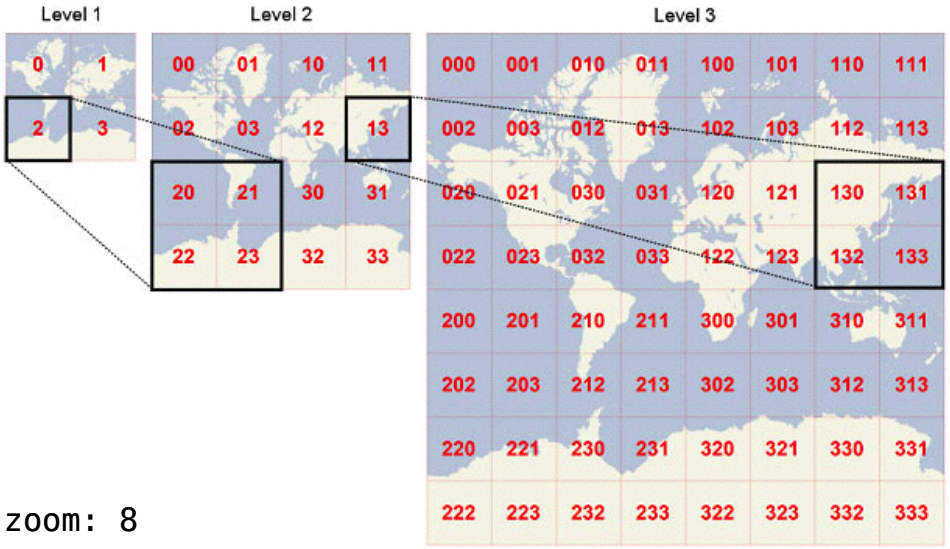
Sistemas de coordinadas
Proyectado: Web Mercator





→ https://thetruesize.com/

Zoom



`zoom: 8`

Se usa el parámetro zoom para definir de forma sencilla la resolución del mapa. Los niveles disponibles están definidos por *maxZoom* (por definición: 28), *zoomFactor* (por definición: 2) y *maxResolution* (por definición esta calculado de una forma que la extensión de la proyección quepa en una tesela de 256x256 pixeles). Empezando con el nivel de zoom 0 con la resolución *maxResolution* unidades por pixel los siguientes niveles de zoom están calculados dividiendo la resolución del nivel de zoom por *zoomFactor*, hasta llegar al *maxZoom*.

Ejercicios

Modificamos el mapa inicial con el siguiente propósito:

1. Mostrar Barcelona centrado sobre Plaza Catalunya
2. Mostrar Barcelona rotado por 44.5º y el mar abajo (consultar parámetro rotation en View API)
3. Mostrar Europa
4. Mostrar el mundo con el pacífico en el centro

Ejercicios solucionados

1. Mostrar Barcelona centrado sobre Plaza Catalunya

```
view: new ol.View({
  center: ol.proj.fromLonLat([2.183333, 41.383333]),
  zoom: 12
})
```

Ejercicios solucionados

2. Mostrar Barcelona rotado por 44.5º y el mar abajo

```
view: new ol.View({
  center: ol.proj.fromLonLat([2.183333, 41.383333]),
  zoom: 12,
  rotation: Math.PI/4/45*44.5
})
```

Ejercicios solucionados

3. Mostrar Europa

```
view: new ol.View({
  center: ol.proj.fromLonLat([21.033333, 52.216667]),
  zoom: 4
})
```

Ejercicios solucionados

4. Mostrar el mundo con el pacífico en el centro

```
view: new ol.View({
  center: ol.proj.fromLonLat([160, 0]),
  zoom: 0,
  rotation: Math.PI
})
```

Tipos de capas

OpenLayers admite diferentes tipos de capas:

- *Tile*: Capas de teselas para conjuntos de teselas ráster
- *Image*: Capas de imágenes para imágenes estáticas o imágenes que se proporcionan a pedido para la extensión del mapa
- *Vector*: Capas vectoriales para datos vectoriales de archivos estáticos para la extensión actual del mapa
- *Vector Tile*: Capas de teselas vectoriales

Capas ráster

OpenLayers viene preconfigurado con algunos proveedores de capas ráster, que sirven como mapas base:

- OpenStreetMap
- Stamen
- Bing Maps

Google Maps no esta soportado desde OpenLayers 4. Se puede usar con el plugin ol3-google-maps y una llave API de Google, pero por razones legales y técnicos no es recomendable.

Muchos otros se puede usar fácilmente invocando a la clase XYZ, por ejemplo los mapas base de Carto o del ICGC.

Capas ráster: Ejemplo capa_raster.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master/en/v6.5.0/css/ol.css"
type="text/css">
    <style>
      body { margin: 0; padding: 0; }
      .map { width: 100%; height: 100%; position: absolute; }
    </style>
    <script
src="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master/en/v6.5.0/build/ol.js"></script>
    <title>OpenLayers: Capas ráster</title>
  </head>
  <body>
    <div id="map" class="map"></div>
    <script type="text/javascript">
      var map = new ol.Map({
        target: 'map',
        layers: [
          new ol.layer.Tile({
            source: new ol.source.OSM()
          }),
        ],
        view: new ol.View({
          center: ol.proj.fromLonLat([2.183333, 41.383333]),
          zoom: 14
        })
      });
    </script>
  </body>
</html>
```

Capas ráster: Stamen

Probaremos diferentes proveedores de capas ráster. Primero probaremos una de Stamen, para hacerlo funcionar solamente hace falta substituir la capa del ejemplo base por el siguiente código:

```
new ol.layer.Tile({
  source: new ol.source.Stamen({
    layer: 'toner'
  }),
  opacity: 0.5,
  visible: true     // poner false para esconder capa
})
```

opacity y *visible* son propiedades que se puede aplicar a todas las capas ráster. *visible* se suele utilizar para mostrar y esconder ciertas capas.

Capas ráster: Bing Maps

Bing Maps ofrece hasta imágenes satelitales, tal como lo conocemos de Google, pero exige una llave, que hay que pedir en Bing Maps Dev Center:

```
new ol.layer.Tile({
  preload: Infinity,
  source: new ol.source.BingMaps({
  key: 'Ata7t8y4_jStXw5LscmH7HbH7oAkbKTGhmr5gvzHHBTETAGgUIJb4r_R3yHiZ3gJ',
    imagerySet: 'Aerial',
  })
})
```

Ofrece diferentes tipos de imágenes (imagerySet), como:

- *Aerial: Aerial imagery.*
- *AerialWithLabels: Aerial imagery with a road overlay.*
- *AerialWithLabelsOnDemand: Aerial imagery with on-demand road overlay.*
- *Streetside: Street-level imagery.*
- *Road: Roads without additional imagery.*
- *CanvasDark: A dark version of the road maps.*
- *CanvasGray: A grayscale version of the road maps.*

...

Capas ráster: XYZ

Además se puede utilizar cualquier fuente de tiles como URL en formato XYZ. Aquí un ejemplo usando una capa base de Carto:

```
new ol.layer.Tile({
  source: new ol.source.XYZ({
    url: 'https://{1-4}.basemaps.cartocdn.com/light_nolabels/{z}/{x}/{y}.png',
    //url: 'https://{1-4}.basemaps.cartocdn.com/rastertiles/voyager/{z}/{x}/
{y}.png',
  })
})
```

Capas ráster: XYZ

El ICGC ofrece muchas capas base de Cataluña libres, como ortofotos y topográficos. Aquí un ejemplo usando el topográfico:

```
new ol.layer.Tile({
  source: new ol.source.XYZ({
    url: 'https://tilemaps.icgc.cat/mapfactory/wmts/topo_suau/CAT3857/{z}/{x}/
{y}.png',
  })
})
```

Ejemplo explicado: Selector de capas

Repasamos paso por paso el ejemplo *capa_raster_switch.html*. Incluye las cuatro capas base nombrados anteriormente y añade un componente radio para activar las diferentes capas.

Primero definimos los elementos HTML de radio con <*input*>:

```
<input type="radio" id="osm" name="radio" checked /><label
for="osm">OSM</lable>
<input type="radio" id="stamen" name="radio" /><lable
for="stamen">Stamen</lable>
<input type="radio" id="bing" name="radio" /><lable for="bing">Bing</lable>
<input type="radio" id="carto" name="radio" /><lable for="carto">Carto</lable>
```

Ejemplo explicado: Selector de capas

Para cada capa definimos la visibilidad con *visible* y especificamos el nombre usando un parámetro propio *name*. Podemos definir los parámetros que queremos, luego lo podemos acceder al parámetro desde JavaScript con la función *layer.get();*

```
const osmLayer = new ol.layer.Tile({
  name: 'osm',
  source: new ol.source.OSM(),
  visible: true
});
```

Ejemplo explicado: Selector de capas

Como último tenemos que capturar el evento de cambio de los elementos *radio*, para eso asignamos un disparador de evento a cada <*input*>.

Una vez detectado el evento *change* iteramos sobre todas las capas y miramos si el nombre de la capa coincide con el atributo *id* del *input*. En caso afirmativo visibilizamos la capa, en caso contrario lo escondemos.

```
let radios = document.getElementsByName('radio');

for (let radio of radios) {
  radio.addEventListener('change', function() {

    for (let layer of layers) {
      layer.setVisible(layer.get('name') === this.getAttribute('id'));
    }
  });
}
```

Barcelonactiva

Ajuntament de Barcelona

bcn.cat/barcelonactiva
bcn.cat/cibernarium