



Ajuntament
de Barcelona

Diseña mapas interactivos con OpenLayers.

Lunes 01/02/2021, 16:00-18:00

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.

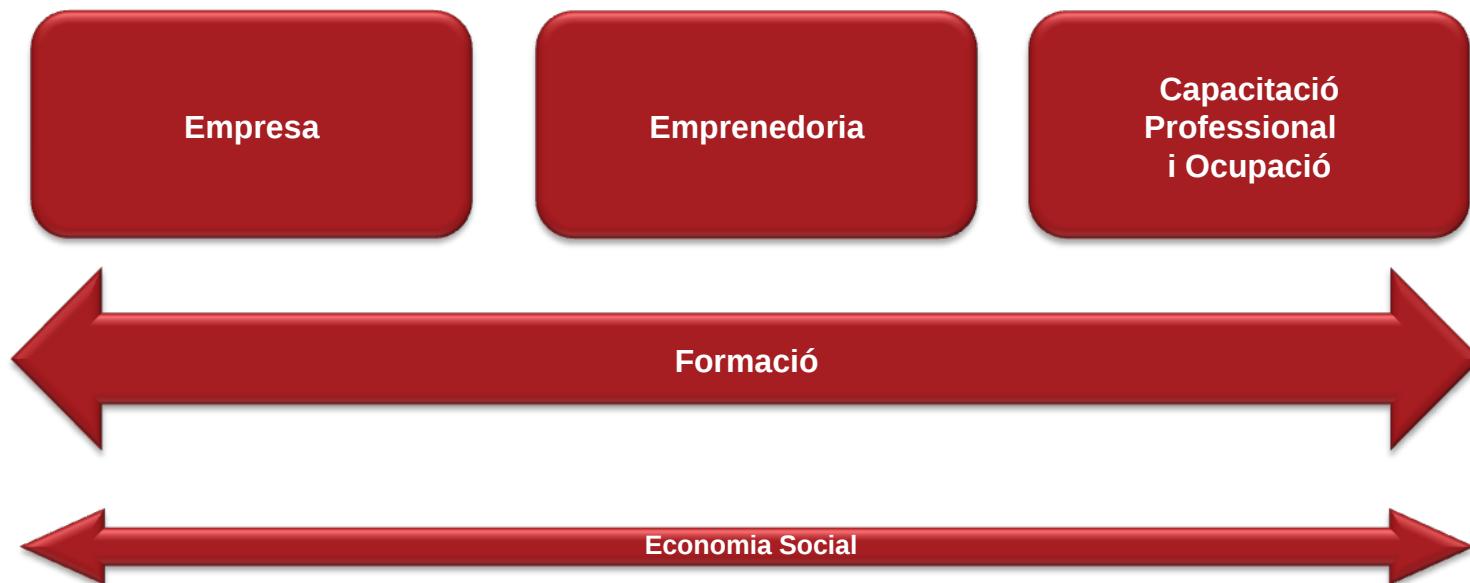
Des de fa 25 anys impulsa el creixement econòmic de Barcelona i el seu àmbit d'influència donant suport a les empreses, la iniciativa emprenedora i l'ocupació, alhora que promociona la ciutat internacionalment i els seus sectors estratègics; en clau de proximitat al territori.



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



Seu Central



Centre
Iniciativa Emprenedora



Incubadora
Glòries



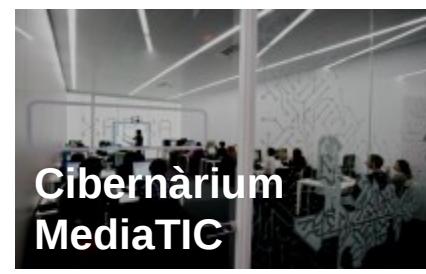
Almogàvers
Business Factory



Parc Tecnològic
BCN Nord



Centre
Desenvolupament
Professional Porta22



Cibernàrium
MediaTIC



Convent
de Sant Agustí



Can Jaumandreu



Ca n'Andalet

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 y su diseño, overlays y interacciones**
- Día 3: Estilos dinámicos, capas WMS, proyecciones y controles
- Día 4: build para entornos de producción, animaciones y plugins



Índice día 2

- Capas vectoriales de datos
- GeoJSON
- Cluster
- Interacción
- Overlay
- Estilos estáticos para capas
- Fuentes de datos



Capas vectoriales de datos

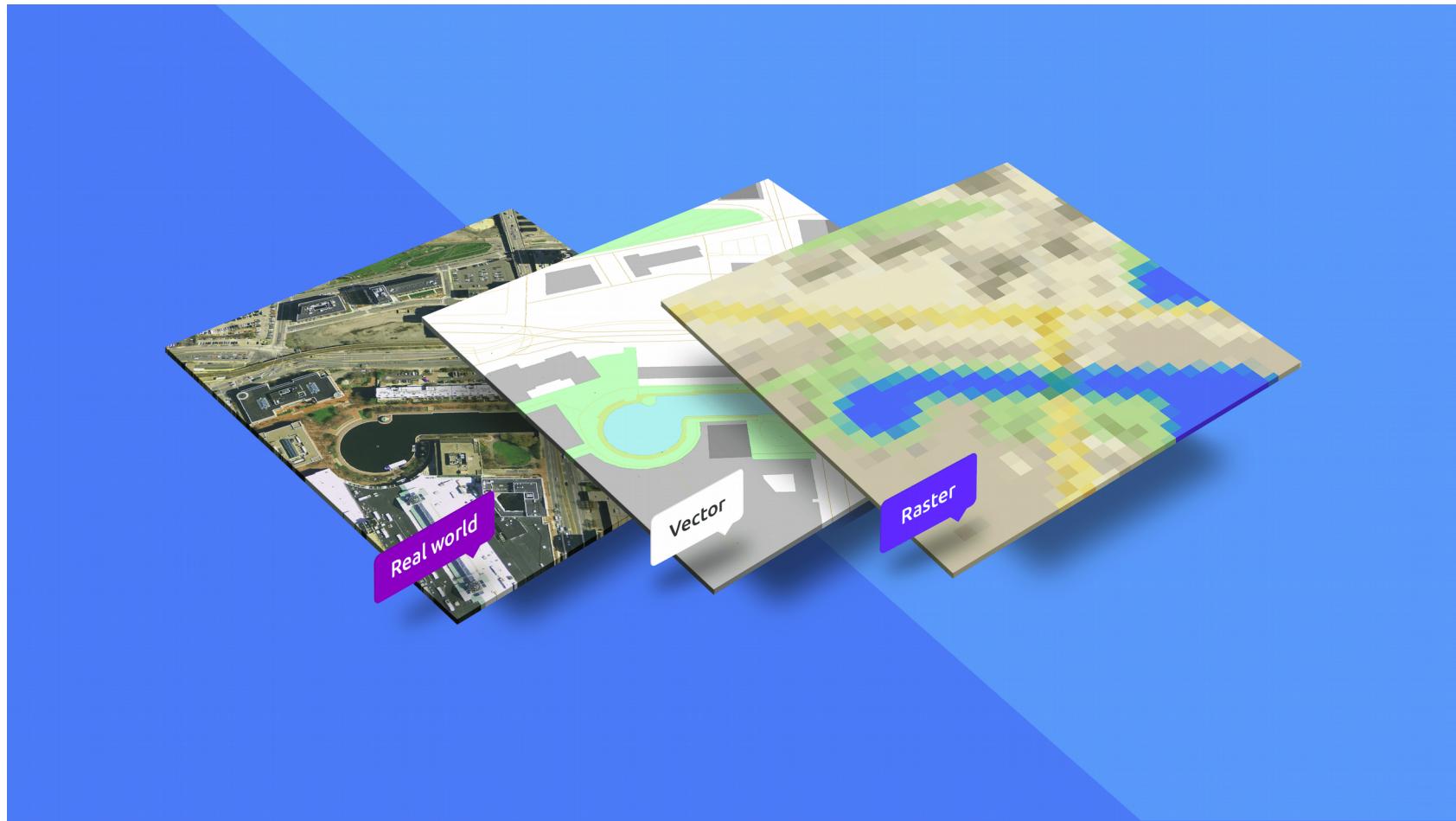
En este apartado miraremos como usar datos vectoriales. Al contrario de capas vectoriales teseladas se cargan a través de un fichero para toda la extensión del mapa, y todos los niveles de zoom.

OpenLayers da soporte a los siguientes formatos:

- GeoJSON
- TopoJSON
- KML
- GML



Capas vectoriales de datos





Capas vectoriales de datos: GeoJSON

Utilizaremos dos archivos en formato GeoJSON, *countries.json* contiene todos los países del mundo y *capitals.json* las capitales. Tengan en cuenta que todos los archivos GeoJSON tienen que estar con coordenadas en EPSG:4326.

QGIS es una excelente herramienta para convertir cualquier formato geográfico en GeoJSON. En este ejemplo se usó de base unos shapefiles de [Natural Earth](#).

Conviene comprobar que el archivo exportado cumple con el estandard de GeoJSON usando un programa Lint, o una de estos servicios online:

- <http://geojson.io/>
- <https://geojsonlint.com/>



Capas vectoriales de datos: GeoJSON

Geometrías:

Type		Examples
Point		{ "type": "Point", "coordinates": [30, 10]}
LineString		{ "type": "LineString", "coordinates": [[30, 10], [10, 30], [40, 40]]}
Polygon		{ "type": "Polygon", "coordinates": [[[30, 10], [40, 40], [20, 40], [10, 20], [30, 10]]]}
		{ "type": "Polygon", "coordinates": [[[35, 10], [45, 45], [15, 40], [10, 20], [35, 10]], [[20, 30], [35, 35], [30, 20], [20, 30]]]}



Capas vectoriales de datos: GeoJSON

Open Save New Share Meta unsaved

</> JSON Table ? Help anon | login

Mapbox Satellite OSM

1 "type": "FeatureCollection",
2 "features": [
3 {
4 "type": "Feature",
5 "properties": {
6 "country": "Bangladesh",
7 "city": "Dhaka",
8 "tld": "bd",
9 "iso3": "BGD",
10 "iso2": "BD"
11 },
12 "geometry": {
13 "coordinates": [90.24, 23.43],
14 "type": "Point"
15 },
16 "id": "BD"
17 }, {
18 "type": "Feature",
19 "properties": {
20 "country": "Belgium",
21 "city": "Brussels",
22 "tld": "be",
23 "iso3": "BEL",
24 "iso2": "BE"
25 },
26]



Capas vectoriales de datos

Cargamos un archivo GeoJSON utilizando una capa vector con una fuente vector. Es importante indicar el formato GeoJSON para la fuente y la ruta URL del archivo. En caso que el archivo venga de otro servidor se debe utilizar una carga asíncrono (Ajax) para no bloquear el programa mientras que se espera a la carga.

```
new ol.layer.Vector({
  source: new ol.source.Vector({
    format: new ol.format.GeoJSON(),
    url: 'countries.json'
  })
})
```



Capas vectoriales de datos

CORS (Cross-origin resource sharing), o intercambio de recursos de origen cruzado quiere decir que no podemos mezclar protocolos. Sale este fallo si abrimos el archivo HTML poniendo la ruta al archivo y así usando el protocolo FILE ya que cuando cargamos el archivo GeoJSON se usa el protocolo HTTP:

Firefox:

Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at file:///Desktop/curso-basico-openlayers/dia2/capitals.json. (Reason: CORS request not http)

Chrome:

Access to XMLHttpRequest at 'file:///Desktop/curso-basico-openlayers/dia2/capitals.json' from origin 'null' has been blocked by CORS policy: Cross origin requests are only supported for protocol schemes: http, data, chrome, chrome-extension, chrome-untrusted, https.

Lo mejor es cargar las páginas web en local usando un servidor web:

- como Apache o Nginx en localhost
- incluido en editores de texto como [Brackets](#), [Atom](#) (con atom-live-server) o Sublime (con LiveServer)
- lanzando manualmente, por ejemplo usando python: `python -m http.server`



Ejercicio

Añade las capitales al mapa mundi de los países. Para eso hay que cargar el fichero *capitals.json* y añadir otra capa vectorial.

Nota: En el caso que no se visualizan las capitales en el mapa, entonces hace falta investigar el fallo con las herramientas del desarrollador web, tanto en Firefox como en Chrome. Revisaremos las siguientes herramientas:

- *Consola*
- *Red/Network*
- *Inspector*



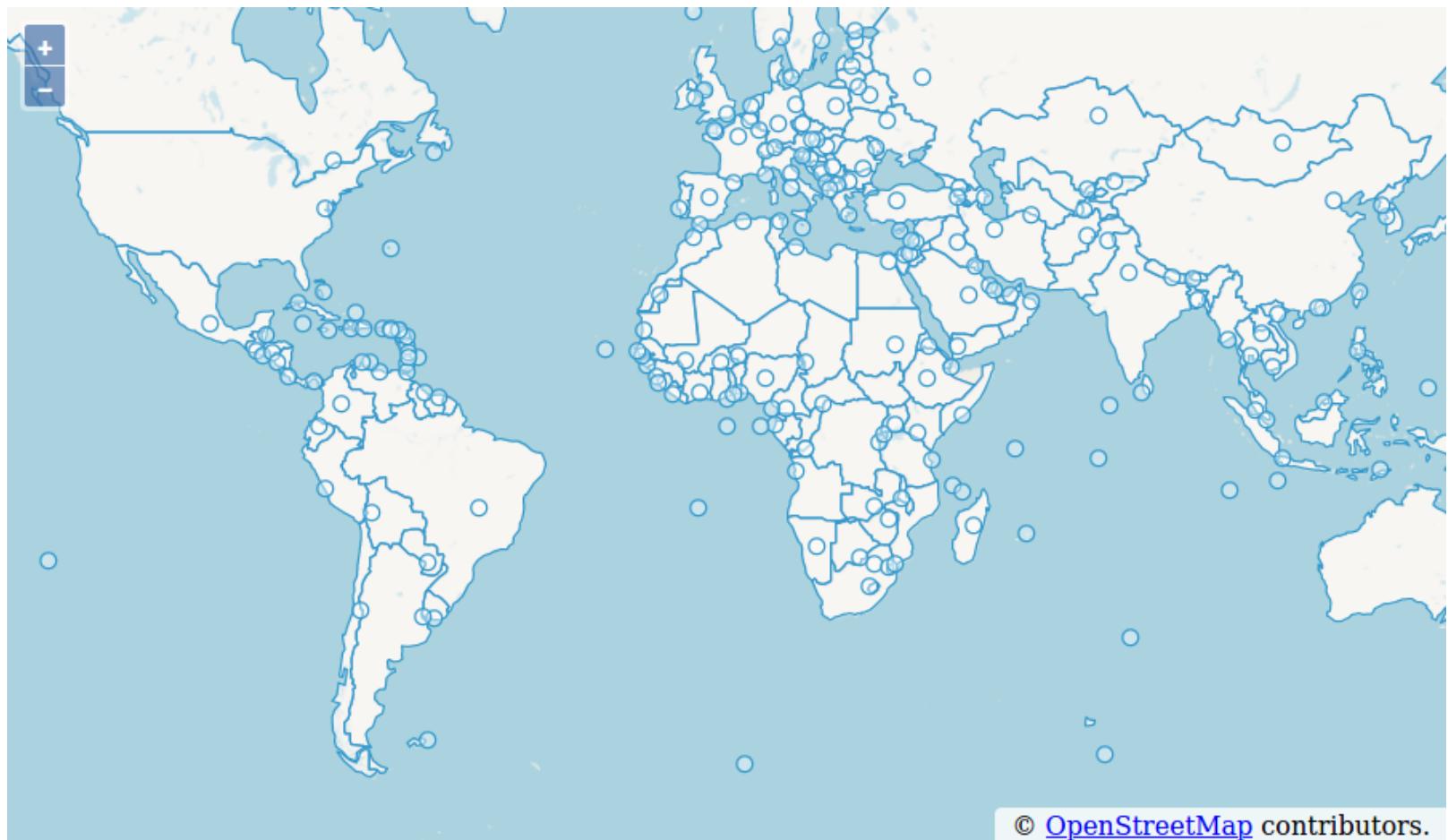
Capas vectoriales de datos: capa_vector.html

```
new ol.layer.Vector({
  source: new ol.source.Vector({
    format: new ol.format.GeoJSON(),
    url: 'countries.json'
  })
}),

new ol.layer.Vector({
  source: new ol.source.Vector({
    format: new ol.format.GeoJSON(),
    url: 'capitals.json'
  })
})
```



Capas vectoriales de datos: capa_vector.html





Ejercicio: Capas vectoriales de datos

Crea un archivo GeoJSON con por lo menos un *feature* del tipo *Point*, por ejemplo para marcar la posición de una ciudad, utilizando el servicio <http://geojson.io/> y cargalo en el mapa.

Nota: Para saber la coordinada de una posición concreta en un mapa también se puede utilizar una pequeña función tipo:

```
map.on('click', function(evt) {  
    console.log('Coordinadas 3857:', evt.coordinate);  
    console.log('Coordinadas 4326:', ol.proj.toLonLat(evt.coordinate));  
});
```



Capas vectoriales de datos: Cluster

Cluster son agrupaciones de features, POIs o íconos en un mapa para reducir la cantidad de información a mostrar. Con un zoom in se va adaptando la cantidad de features a mostrar, como se ve en el ejemplo *capa_vector_cluster.html*:

```
new ol.layer.Vector({
  source: new ol.source.Cluster({
    source: new ol.source.Vector({
      format: new ol.format.GeoJSON(),
      url: 'capitals.json'
    })
  })
})
```

Proyecto OpenLayers que usa un cluster:
<https://www.solivid.org/construim-un-mapa/>



Capas vectoriales de datos: Interacción

Para empezar utilizamos un `<div>` para mostrar la información:

```
<div id="name" style="font-size: 24px;"></div>
```

Definimos un evento `pointermove` que reacciona a cualquier movimiento del ratón y lo muestra en el `<div>` abajo del mapa:

```
map.on('pointermove', function(evt) {
    let el = document.getElementById('name');
    el.innerHTML = '';

    let coordinate = evt.coordinate;
    el.innerHTML += 'Coordinates: ' + ol.proj.toLonLat(coordinate) + '<br>';

    let pixel = map.getPixelFromCoordinate(coordinate);
    map.forEachFeatureAtPixel(pixel, function(feature) {
        if (feature.get('name') === undefined) {
            el.innerHTML += 'Capital: ' + feature.get('city') + '<br>';
        }
        else {
            el.innerHTML += feature.getProperties().name + '<br>';
        }
    });
});
```



Capas vectoriales de datos: Popup

Hay diferentes técnicas para mostrar popups encima de un mapa OpenLayers, la que me resulta más cómoda y flexible es utilizando este elemento HTML con estilos definidos en el fichero *popup.css*:

```
<div id="popup" class="ol-popup">
  <a href="#" id="popup-closer" class="ol-popup-closer"></a>
  <div id="popup-content"></div>
</div>
```

Overlay es una clase de OpenLayers que muestra un contenido encima del mapa ligado a una coordinada:

```
var overlayPopup = new ol.Overlay({
  element: document.getElementById('popup'),
  autoPan: true,
  autoPanAnimation: {
    duration: 250
  }
});
map.addOverlay(overlayPopup);
```



Capas vectoriales de datos: Popup

Asignamos el contenido al popup y usamos la función *setPosition* de *Overlay* para posicionar el popup sobre la coordinada clickada:

```
map.on('click', function(evt) {
  let coordinate = evt.coordinate;
  let pixel = map.getPixelFromCoordinate(coordinate);
  let output = '';

  overlayPopup.setPosition(undefined);

  map.forEachFeatureAtPixel(pixel, function(feature) {

    if (feature.get('name') === undefined) {
      output += 'Capital: ' + feature.get('city') + '<br>';
    }
    else {
      output += feature.getProperties().name + '<br>';
    }

    document.getElementById('popup-content').innerHTML = output;
    overlayPopup.setPosition(coordinate);
  });
});
```



Estilos estáticos para capas

style es un parámetro de las capas vectoriales de datos y le podemos dar un color de relleno y de trazado.

Vamos a substituir el círculo por definición de los capitales por un estilo propio en el ejemplo del día 1 llamado *mundo.html*:

```
style: new ol.style.Style({
  image: new ol.style.Circle({
    stroke: new ol.style.Stroke({
      width: 2,
      color: "#ff0000"
    }),
    fill: new ol.style.Fill({
      color: "#000"
    }),
    radius: 3
  })
})
```



Estilos estáticos para capas

Para utilizar un ícono de bitmap (preferiblemente png o svg), utilizamos la clase *Icon*:

```
style: new ol.style.Style({  
    image: new ol.style.Icon({  
        src: 'marker.png'  
    })  
})
```

Tiene parámetros para especificar el tamaño, la posición y mucho más, mira la API:

→ https://openlayers.org/en/latest/apidoc/module-ol_style_Icon-Icon.html



Estilos estáticos para capas

También se puede aplicar múltiples estilos a una misma capa:

```
var shadowStyle = new ol.style.Style({
  stroke: new ol.style.Stroke({
    color: [0, 0, 127, 0.15],
    width: 8
  }),
  zIndex: 1
});
var countryStyle = new ol.style.Style({
  fill: new ol.style.Fill({
    color: [203, 194, 185, 1]
  }),
  stroke: new ol.style.Stroke({
    color: [101, 95, 90, 1],
    width: 1,
    lineCap: 'round'
  }),
  zIndex: 2
});

style: [shadowStyle, countryStyle]
```



Ejercicio

Crea un mapa de Barcelona, Cataluña o el mundo y carga uno o varios de los siguientes GeoJSON de la carpeta `/data`:

- Barrios y distritos de Barcelona en GeoJSON (`barris.geo.json`, `districtes.geo.json`) o TopoJson (`barris.topo.json`, `districtes.topo.json`)
- Ayuntamientos de Cataluña desde [dades obertes catalunya](#) o archivo adjunto *Ajuntaments de Catalunya.geojson* (puntos y no polígonos)
PD: Esta online en formato CSV, se puede utilizar QGIS para exportarlo en formato GeoJSON, o como alternativa servicios online como por ejemplo <http://convertcsv.com/csv-to-geojson.htm>
- Carta arqueológica de la ciudad de Barcelona: Descarga desde [Open Data Barcelona](#) o archivo adjunto *2019_Carta_Arqueologica_de_Barcelona.geojson*

Una vez visualizados los features del GeoJSON con éxito, aplicamos estilos propios a las capas de vectores.



Open Data

“El conocimiento es abierto si cualquiera es libre de acceder a él, usarlo, modificarlo y compartirlo, estando sujeto a lo sumo a medidas que preserven su autoría y su apertura.”

Definición Open Data de la [Open Knowledge Foundation](#)

Algunos catálogos de datos abiertos:

- [Dades obertes catalunya](#)
- [Open Data Barcelona](#)
- [datos.gob.es](#)
- [EU Open Data Portal](#)
- [OECD Open Data](#)
- [World Bank Open Data Catalogue](#)
- [Gapminder Data](#)
- [Global Open Data Index](#)



Fuentes de datos geográficos abiertos

- [Natural Earth](#)
- [INSPIRE geoportal](#)
- [GeoSeer \(WMS\)](#)
- [IGN \(Instituto de Geografía\)](#)
- [ICGC \(Institut Cartogràfic i Geològic de Catalunya\)](#)
- [Carto BCN](#)
- [Geoportal BCN](#)
- [Recursos ICGC en obert](#)



OpenStreetMap como fuente de datos

Hay diferentes formas de acceder los datos de OpenStreetMap (OSM), una muy potente pero no muy fácil es utilizando su [API](#) Overpass.

Hay una herramienta online que permite utilizar esta API simplificada, se llama [Overpass Turbo](#) y hay un buen [Wizard](#). Una vez seleccionado se puede exportar los datos en formato GeoJSON.

Nota: A veces el archivo GeoJSON exportado no cumple el estandard y es mejor comprobarlo con [geojson.io](#) o [geojsonlint.com](#).

Un primer ejemplo sería conseguir todos las escuelas de Barcelona escribiendo la siguiente orden en el Wizard:

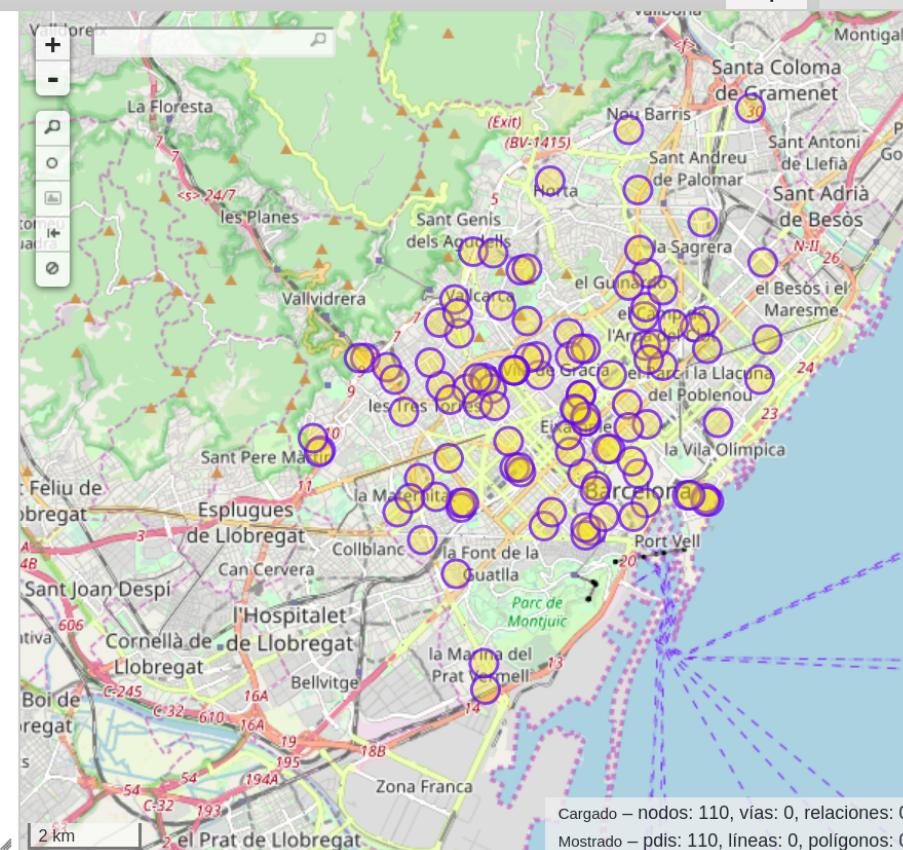
```
amenity=school in Barcelona
```



OpenStreetMap como fuente de datos

Ejecutar Compartir Exportar Asistente Guardar Cargar Configuraciones Ayuda overpass turbo ⚙Mapa Datos

```
/*
This has been generated by the overpass-turbo wizard.
The original search was:
"amenity=school in Barcelona"
*/
[out:json][timeout:25];
// fetch area "Barcelona" to search in
{{geocodeArea:Barcelona}}->.searchArea;
// gather results
(
    // query part for: "amenity=school"
    node["amenity"="school"](.area.searchArea);
);
// print results
out body;
>;
out skel qt;
```



Cargado – nodos: 110, vías: 0, relaciones: 0
Mostrado – pdis: 110, líneas: 0, polígonos: 0



OpenStreetMap como fuente de datos

Ejemplos Overpass Turbo Wizard:

- *bar* (lo mismo que *amenity=bar*)
- *amenity=bar in Barcelona* (¡limitar área!)
- *amenity=school in Barceloneta*
- *name="Edifici Media-TIC"* (¡cuidado a mayúsculas y menúsulas!)
- *highway=* in Barcelona* (¡muchos datos!)
- *public_transport=* in "Ciutat Vella"* (¡comillas!)
- *public_transport=* in bbox* (*bbox* es el área visible en el mapa)
- *public_transport=* around "Edifici Media-TIC"*
- *amenity=library and operator="Diputació de Barcelona" in Barcelona*



Ejercicio

Practicamos con el Overpass Turbo Wizard exportando los siguientes datos o inventando sus propios pedidos:

- Todos los arboles de Barcelona
 - Todas las ciclovías de Girona
 - Todas las restaurantes de Tarragona
 - Todos los aparcamientos de Lleida
- ...
- Exportar datos en formato GeoJSON y visualizarlo con OpenLayers

Barcelonactiva



Ajuntament
de Barcelona

bcn.cat/barcelonactiva
bcn.cat/cibernarium