Mapas vectoriales offline en apps multiplataforma

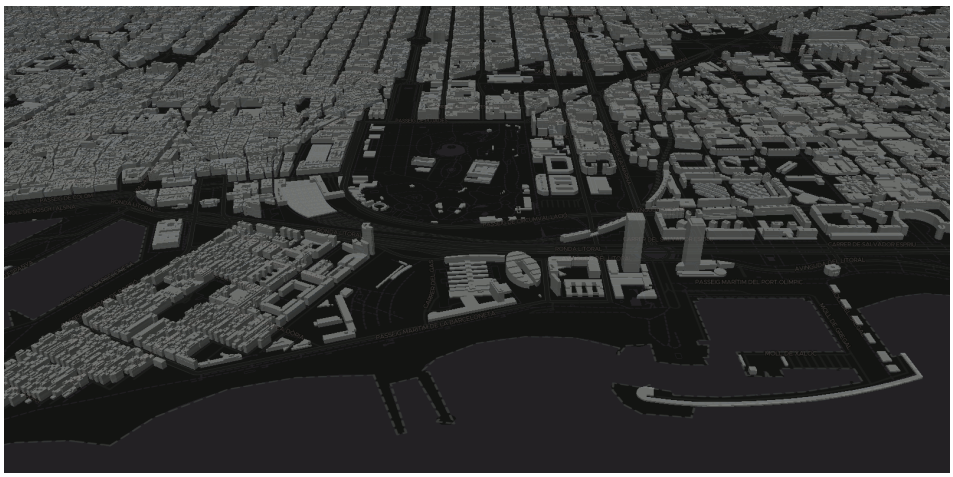
Oscar Fonts
Micho García
geomati.co

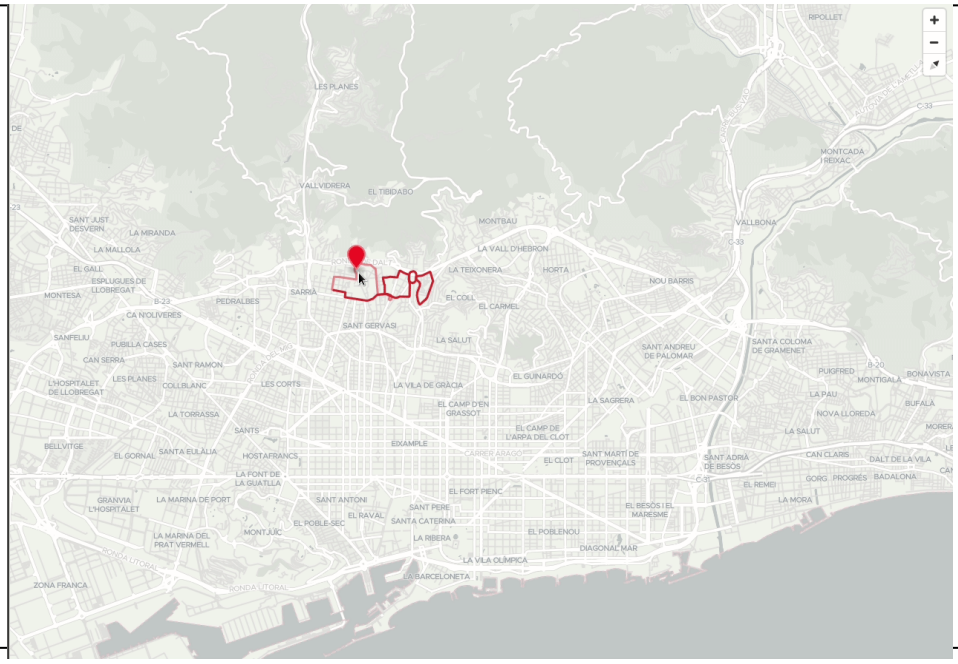
Os voy a contar la historia de un desarrollo

El subproducto resultante ha suscitado cierto interés

Nos dedicamos al web mapping, y hemos experimentado con las bondades de las teselas vectoriales

Hemos hecho algunos juguetes interesantes con la API de Mapbox GL JS





Nos piden una app de mapas offline y pensamos: Vector Tiles, ¡por supuesto!

¿Por qué?

Pesan poco

- **Todo Barcelona** en OpenMapTiles son unos 12 MB.
- **Medio millón de construcciones** del catastro son 37 MB.
- **Todo Catalunya** a escala 1:25 000 son 350 MB.

Te dan flexibilidad

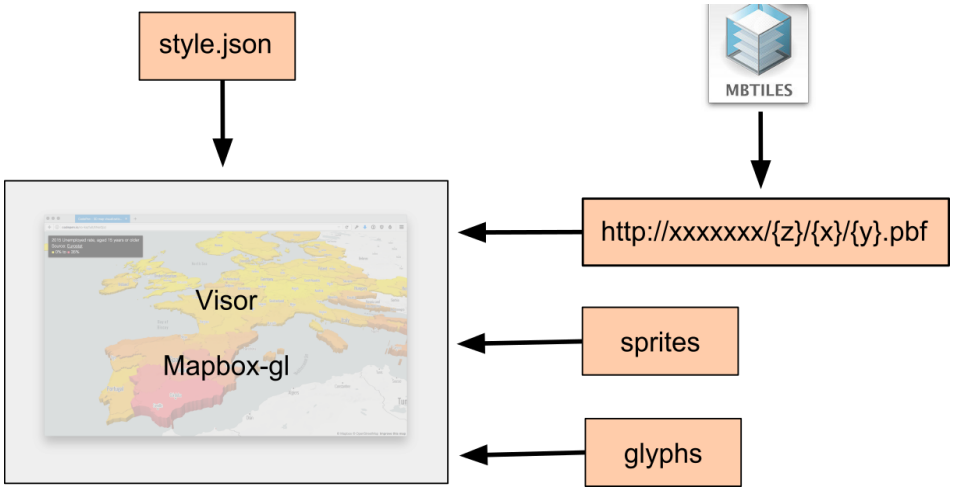
- Esos 12 MB del Barcelona los puedo pintar al nivel de zoom que quiera, y con el estilo que quiera.
- Esas 500.000 construcciones las puedo pintar en 3D en un movil, simbolizar al vuelo… ¡animarlas con música!

Disponemos de una API en JS brutal:

Se llama Mapbox GL JS

¿Hay mejor alternativa?

Decididos a aprovechar la API de MapboxGL JS.



Al googlear "Mapbox GL JS Offline"

La respuesta oficial es

- No, **JS y offline no**.
- Usa las **variantes nativas** de MapboxGL.
- Si quieres multiplatforma prueba **MapboxGL React Native**.

Pero no me gusta

- Yo quiero mi API, no un tinglado por encima.
- Los wrappers de APIs los carga el Diablo (tm).
- Yo a lo mío: Vanilla JS.

Primera iteración:

- Mapbox GL JS tal cual en Cordova.
- Todos los recursos on line.

Y funciona sin más. Check.

Segunda iteración:

- Movemos los recursos al movil (en `www`).
- Esto es: `sprite`, `glyphs` y `tiles`.

Hay pegas.

Pegas:

- Las rutas no pueden ser relativas:

  ```
  "glyphs": "fonts/{fontstack}/{range}.pbf"
  ```

  vs.

  ```
  "glyphs": "file:///www/fonts/{fontstack}/{range}.pbf"
  ```

- Los sprites que empiezan por `file:///` no se cargan por un bug. Solución: PR 4649

¿y las teselas?

A lo bruto.

Estructura de directorios:

```
0/0/0.pbf
1/0/0.pbf
1/0/1.pbf
1/1/0.pbf
1/1/1.pbf
```

Au, arreglado. Todos a casa.

No tan rápido

- `18/262144/262144.pbf`

Miles de directorios, con miles de teselas en cada uno, desparramados en `www`. Eso no hay quien lo gestione.

Tercera iteración: `.mbtiles`

Un único fichero que contiene todas las teselas.

El problema

ahora es:

Cómo leer un `.mbtiles` desde Mapbox GL JS

Que se divide en dos subproblemas:

- Primero, ¿cómo extraigo una tesela de un `.mbtiles`?
- Y luego, ¿cómo se la paso a MapboxGL JS?

¿cómo extraigo una tesela de un `.mbtiles`?

¿Qué es un

¿Qué es un `.mbtiles`?

Una BDD SQLite con una tabla: "z", "x", "y", "tesela".

Muy fácil:

- Googlear "Cordova SQLite Plugin".
- Existen plugins varios.
- Este nos sirve: `cordova-sqlite-ext`.

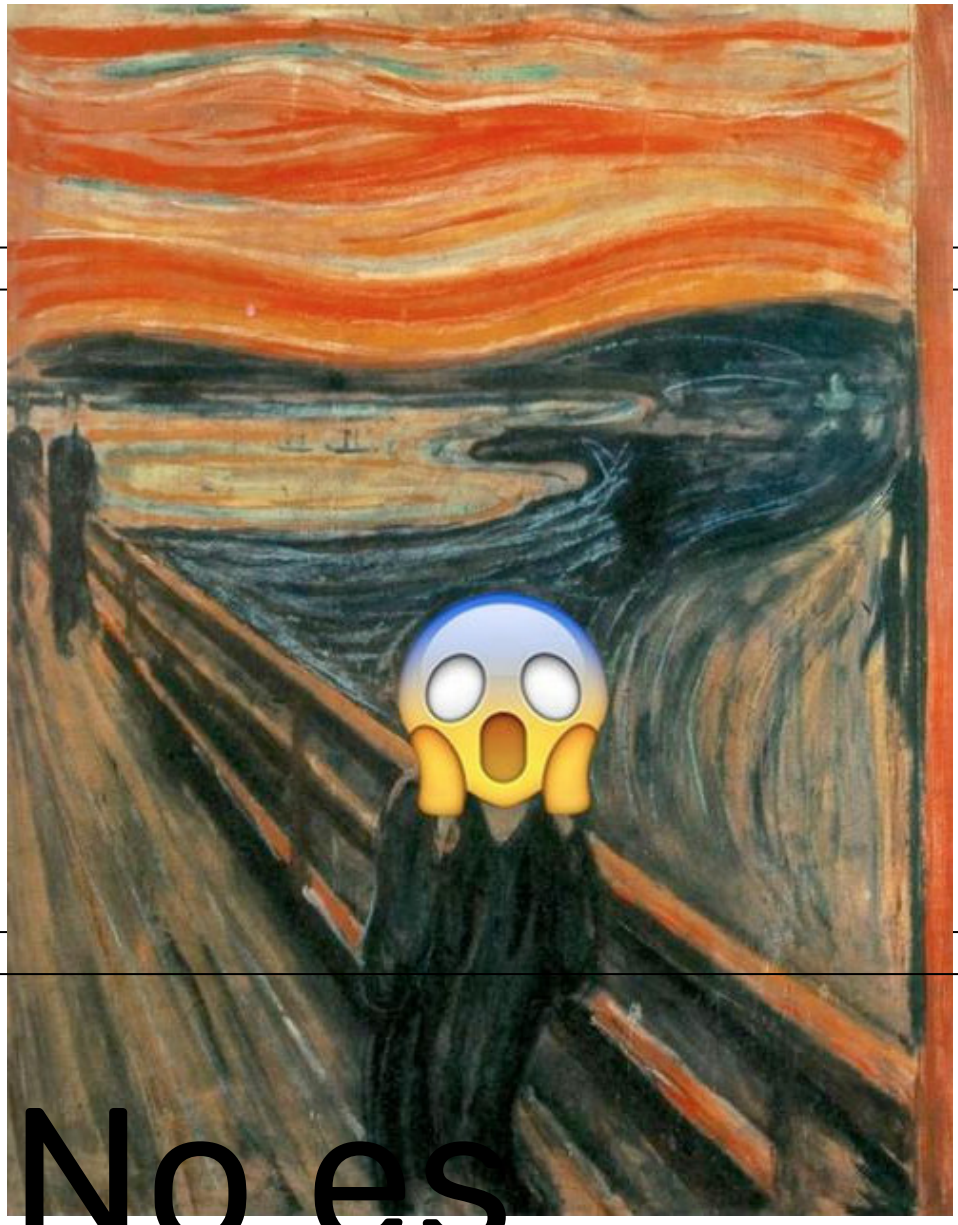
Ya podemos recuperar una tesela

pbf y la tenemos en un objeto JS.

La segunda parte del problema:
¿cómo se la pasamos a MapboxGL JS?

- La librería sólo sabe pedir teselas a URLs.
- Hay quienes han montado un servidor HTTP local embebido en la App.

Otros hemos modificado la ...



No es para tanto

Reescribir un método "`loadTile`"

para que donde dice

para que donde dice "AJAX" diga "SQL"

Y reempaquetar

(sí, el infierno está en los

detalles)

Aprovechamos para encapsular la ñapa de las URLs absolutas (véase segunda iteración)

Y convertimos todo eso en una librería reutilizable.

Apps en Play Store
(no son nuestras)

- OpenMapTiles
- Catalunya Offline

Fin.

¡No tan rápido amigo!

Show me the code

the code.

SHOW ME THE F*ING CODE!!!

que-esto-es-el-SIG-Libre

https://github.com/oscarfonts/mapbox-gl-cordova-offline

Vale.

¿Y cómo se usa?

Map

```
let map = mapboxgl.Map({
    container: 'map',
    style: 'styles/osm-bright/style.json'
});

map.addControl(new mapboxgl.NavigationControl());
```

vs. OfflineMap

```
new mapboxgl.OfflineMap({
    container: 'map',
    style: 'styles/osm-bright/style-offline.json'
}).then(function(map) {
    map.addControl(new mapboxgl.NavigationControl());
});
```

style.json

```
{
    "sprite": "http://example.com/style/sprite",
    "glyphs": "http://example.com/fonts/{fontstack}/{range}.pbf",
    "sources": {
        "online-source": {
            "type": "vector",
            "tiles": ["http://example.com/data/{z}/{x}/{y}.pbf"]
        }
    }
}
```

vs style-offline.json

```
{
    "sprite": "style/sprite",
    "glyphs": "fonts/{fontstack}/{range}.pbf",
    "sources": {
        "offline-source": {
            "type": "mbtiles",
            "tiles": "data/source.mbtiles"
        }
    }
}
```

Mejoras a la vista

- Soportar teselas raster (png, jpeg).
- Mejor gestión del fichero `.mbtiles`.
- Mantener compatibilidad con futuras versiones de Mapbox GL.

(contrátanos ♥)

Oscar Fonts <oscar.fonts@geomati.co>
Micho García
<micho.garcia@geomati.co>

http://fonts.cat/siglibre2018