

# VISOR DE RESTAURANTES EN CATALUÑA

RESUMEN - TESINA MÁSTER SIG

Máster en Sistemas de Información Geográficos  
Tech Talent Center - UPC School



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
School of Professional & Executive Development

Autora: Eliana Rubiés Roldán  
Tutor: Víctor Pascual Ayats  
Promoción: 2018-2019

## 1. INTRODUCCIÓN

El uso de mapas para visualizar información está cada vez más extendido en la vida cotidiana, por este motivo los recursos ofrecidos cada vez son mayores. Si echamos la vista atrás, no hace tantos años del uso del mapa en formato papel. Gracias a la evolución tecnológica de los móviles y a la aparición de Google Maps, la gente empezó a usar mapas digitales para ubicarse, mostrar las mejores rutas o simplemente localizar lugares de interés.

Hace un año estaba preparando mi boda y me surgió la necesidad de buscar lugares donde se celebren bodas, pero me fue imposible encontrar un visor simple sobre el que mapificarlos. Seguramente mucha gente se ha encontrado en esta misma situación y por ese motivo decidí investigar en el asunto.

La presente tesina titulada VISOR DE RESTAURANTES EN CATALUÑA pretende hacer una reflexión a esta cuestión, pero trabajando en un tema más genérico como la búsqueda de restaurantes.

Para realizar dicho estudio, se ha escogido la zona de Cataluña, centrándose en las 4 capitales de provincia: Barcelona, Tarragona, Girona y Lleida.

## 2. OBJETIVOS Y ALCANCE

La presente tesina tiene como principal objetivo generar un visor, en el que se muestren lugares según la selección de ubicación del usuario.

Para ello se construirá un selector a través del cual se podrá elegir la ciudad en la que se quiere realizar la búsqueda. Tal y como se ha comentado en el punto anterior, habrá 4 posibles opciones. Una vez decidido el lugar, el visor centrará la vista del mapa según la selección del usuario para ver el mapa con más detalle. En este punto aparecerán los marcadores sobre el mapa indicando la situación de los restaurantes, con la particularidad que algunos se agruparán por clúster.

El motivo principal para el uso de estas agrupaciones es ofrecer una visión más limpia del mapa. En algunos casos la concentración de puntos de interés sobre el mapa era muy densa, impidiendo una correcta visualización de los restaurantes.

Además, otro punto interesante es el hecho de proporcionar más detalles del restaurante seleccionado. La primera idea al respecto era que la propia API de OSM devolviera la URL correspondiente y a partir de ella redirigir al usuario a la web, pero resultó que muchos de los lugares no tenían la web informada. Por este motivo, se decidió buscar una alternativa. Esta solución consistía en clicar sobre el punto de interés y redirige al usuario a Google, rellenando la propia barra de búsqueda del buscador con el nombre del lugar elegido.

## 3. METODOLOGÍA

El desarrollo de la presente tesina se ha dividido en tres grandes bloques: *Leaflet*, *API Overpass* y *JavaScript*, *HTML* y *CSS*.

### 3.1. Leaflet

*Leaflet* es una librería JavaScript *open source*, compatible con la mayoría de las plataformas móviles y escritorio, que se usa para la publicación de mapas. Gracias a esta librería podremos diseñar el mapa e incluir lo que deseemos en él.

Se indica a continuación lo más destacado de *Leaflet* usado para este trabajo:

Para poder visualizar un mapa, se debe definir el marco donde queremos que aparezca el mapa. Para ello usaremos el **objeto map**.

Además, sobre el mapa se pueden configurar una serie de parámetros, como son:

El control de *zoom*:



IMAGEN 1 ZOOM

O el control de capas:

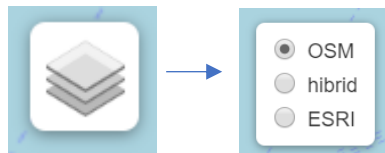


IMAGEN 2 CONTROL CAPAS

Para visualizar el mapa, es necesario indicar las capas que se quieren mostrar. Esto lo haremos con la **clase TileLayer** y una vez definidas, se añadirán al mapa con el **método addTo(map)**. En este proyecto definiremos 3 capas, como se muestra en la figura anterior: *OpenStreetMap*, *ESRI* (ortofoto) e híbrida (combinación de la ortofoto y un diseño de *Stamen*, en concreto el diseño *toner*: <http://maps.stamen.com/#terrain/12/37.7706/-122.3782>)

Para marcar los puntos de interés se ha querido incluir un diseño específico de *marker*. Por ello se usa el **objeto L.Icon**. El icono usado es una imagen .svg que originariamente era de color negro, pero para visualizarla correctamente sobre el mapa, se ha modificado su color en el propio archivo .svg añadiendo **fill="#c82a54"**:

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="Capa_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
width="263.335px" height="263.335px" fill="#c82a54" viewBox="0 0 263.335 263.335" style="enable-background:new 0 0 263.335 263.335;"
xml:space="preserve">
```

Además, se incluye también la posibilidad de agrupar los *markers* en caso de mucha densidad de puntos sobre el mapa, visualizándose de la siguiente manera:

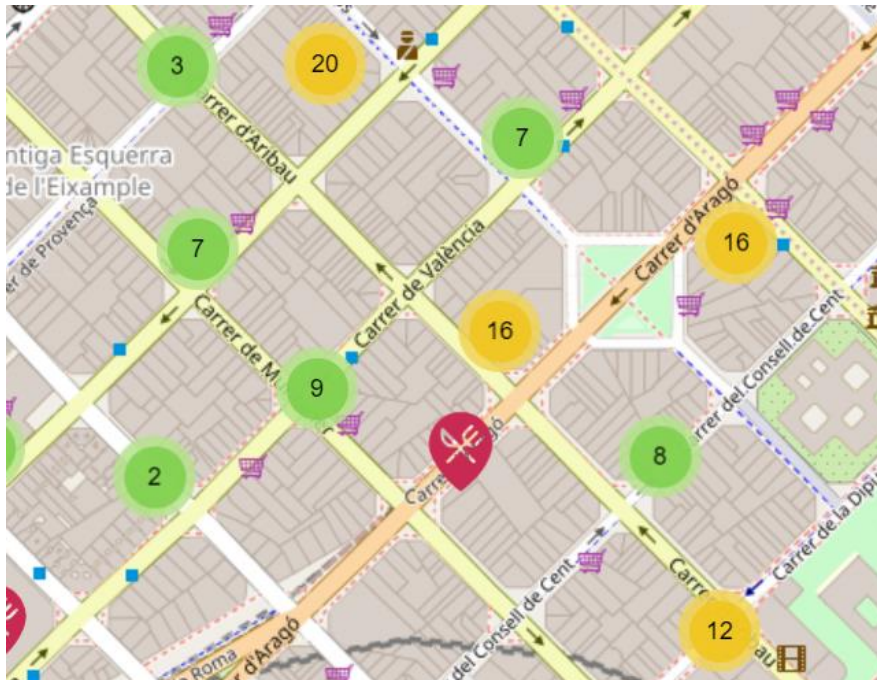


IMAGEN 3 MARKERS Y CLÚSTER

### 3.2. API Overpass – Open Street Maps (OSM)

*OpenStreetMap* es la mayor base de datos geográfica de libre acceso que existe. La información es introducida y actualizada por usuarios voluntarios de todo el mundo y toda la información de la base de datos es de libre acceso y uso.

La información que proporciona OSM es de carácter espacial y temática. La información espacial se puede representar como un nodo, una vía o una relación; mientras que la información temática se representa mediante etiquetas.

El elemento más simple es el nodo y se usa para representar puntos. Debido a la naturaleza de los datos que queremos incorporar sobre el mapa, este será el elemento que usaremos. De los atributos que pueden contener los nodos, los que nos interesarán para este trabajo son la latitud y la longitud.

La estructura de una petición tiene un formato muy concreto: se divide en declaraciones, terminando cada una de ellas en un punto y coma. La última declaración será "out". La petición indicará que tipo de elemento se desea mostrar, en qué área y la etiqueta de ese elemento (si se desea una en concreto).

Para indicar el área hay distintas maneras, la usada en este caso es **around**, ya que se busca encontrar elementos cerca de un punto determinado. Este punto concreto será la ciudad elegida por el usuario mediante el selector.

### 3.3. JavaScript, HTML y CSS

**JavaScript (JS)** es un lenguaje de programación interpretado orientado a objetos. Se usa principalmente en lado cliente, permitiendo crear páginas dinámicas.

**HTML** es el lenguaje usado para la elaboración de páginas web. Se basa en XML, lenguaje de marcado de etiquetas.

**CSS** es un lenguaje de estilo en cascada que define como se ve el HTML. El término cascada hace referencia a que el estilo se aplicará a un elemento y a todos los que estén por debajo de él.

La combinación de estos tres lenguajes nos ayudará a generar el resto de los elementos que no son específicos del mapa, así como conseguir el comportamiento que queremos que tenga la interfaz según interactúe el usuario.

## 4. RESULTADOS: ENLACE AL VISOR

Se puede acceder al visor desde:

[https://elianarr.github.io/Tesina\\_Visor-Restaurants/](https://elianarr.github.io/Tesina_Visor-Restaurants/)

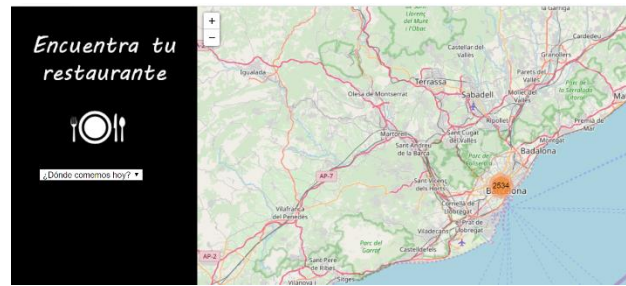


IMAGEN 4 ENLACE AL VISOR

## 5. CONCLUSIONES

Como resultado del trabajo realizado, puedo concluir que el uso de librerías y datos libres puede proporcionar al mercado soluciones de gran calidad, que nada tienen que envidiar a las APIs de pago, como por ejemplo Google.

Después de haber analizado el potencial de la API de *Leaflet*, se concluye que con ella y con el uso de los *plugins* existentes es posible desarrollar un visor completo y sin limitaciones si se compara con otras soluciones de pago.

Asimismo, se ha podido comprobar la calidad de los datos libres y abiertos que aporta OSM. Recordemos que OSM es un proyecto colaborativo que se alimenta de datos proporcionados por los usuarios registrados. Gracias a todos los datos aportados, se ha podido generar la API *Overpass* usada en este proyecto y que, como ya se ha mencionado, actúa como base de datos.

La existencia de esta API ha facilitado el desarrollo del visor, pues ha evitado tener que generar una base de datos amplia, cuya elaboración hubiera supuesto mucho esfuerzo. Gracias a ello, ese tiempo se ha invertido en profundizar en el resto de los objetivos, como el lenguaje *Javascript*, *Leaflet* y la propia *API Overpass*.

No obstante, es cierto que se ha detectado que en algunos casos los datos estaban incompletos. Es el caso de las páginas web. La idea inicial era poder navegar a la propia página web del lugar cuando se clicase sobre el *marker*. Sin embargo, esto no ha sido posible por falta de datos en algunos casos. Por este motivo, se ha decidido solucionar este pequeño inconveniente creando un enlace a Google y rellenando automáticamente la barra de búsqueda con el nombre del lugar.

En última instancia, la realización de esta tesina también ha supuesto adentrarse en el mundo de la programación en *Javascript*. Este era uno de los objetivos de este trabajo, pues hoy en día es un lenguaje muy usado por los programadores de *front-end*. Se ha podido comprobar que, con esfuerzo, se puede llegar a desarrollar un visor de forma simple.

Por consiguiente, se puede afirmar que hoy en día es posible crear visores muy completos mediante el uso de código abierto; por lo que animo a todo aquel que esté interesado a adentrarse en este mundo y a potenciar su uso.