

OpenLayers y Leaflet: Code-Combat

Moisés Arcos Santiago⁽¹⁾, María Arias de Reyna Domínguez⁽²⁾

⁽¹⁾ Ingeniero Informático, Emergya, Sevilla, marcos@emergya.es.

⁽²⁾ Ingeniera Informática, GeoCat bv, Bennekom, Países Bajos, maria.arias@geocat.net.

RESUMEN

A lo largo de este artículo se realizará una comparativa entre las dos librerías web de visualización de mapas más importantes: OpenLayers y Leaflet.

Palabras clave: *openlayers, leaflet*

ABSTRACT

On this article, we are going to overview the two most important web libraries for visualization of maps: OpenLayers and Leaflet.

Key words: *openlayers, leaflet*

INTRODUCCIÓN

Durante muchos años, la librería canónica para visualización de mapas en web ha sido, con gran diferencia, OpenLayers. Esta librería, que nació hace ya casi una década, nos ha acompañado a lo largo de multitud de aplicaciones y se hacía un imprescindible en proyectos de cierta envergadura.

Sin embargo, en los últimos dos años, una nueva librería javascript ha empezado a resonar fuerte como alternativa viable y sencilla de manejar a OpenLayers: Leaflet. En este artículo queremos comprobar si, realmente, ambas librerías ofrecen funcionalidades similares y hasta qué punto pueden ser intercambiables en un desarrollo.

Tipos de Capas

Un aspecto fundamental a la hora de utilizar una librería de mapas es conocer cuantos tipos de capas diferentes soporta y cómo de complejo es la utilización de estas capas.

En cuanto a capas raster, nos encontramos que Leaflet soporta todos los tipos básicos: WMS, Tile e Imagen. Sin embargo, OpenLayers además de tener estos mismos tipos de capas raster, ofrece algunos tipos más, como el WMTS.

Igualmente, OpenLayers nos ofrece también una serie de tipos de capas preconfigurados para poder utilizarlas capas directamente sin preocuparnos de cómo funcionan. Por ejemplo, para utilizar una capa OSM en los mapas, podemos comparar la diferencia entre los constructores de ambas librerías:

Leaflet:

```
var osmUrl='http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png';
var osmAttrib = 'Map data © <a href="http://openstreetmap.org">OpenStreetMap contributors';
var osm = new L.TileLayer(osmUrl, {attribution: osmAttrib});
map.addLayer(osm);
```

OpenLayers:

```
var osm = new OpenLayers.Layer.OSM();
map.addLayers([osm]);
```

En cuanto a las capas vectoriales, la diferencia es similar. Ambas librerías ofrecen un tipo básico de capa vectorial al que se le pueden añadir geometrías de diferentes tipos: polígono, línea, círculo. Ambas librerías también ofrecen WFS, WFS-t y parseadores GeoJSON para la construcción de capas vectoriales. La única diferencia fundamental, una vez más, es que OpenLayers ofrece una serie de capas preconfiguradas para simplificar los desarrollos, mientras que Leaflet, ofreciendo las mismas funcionalidades, obliga al usuario a introducir manualmente estas configuraciones.

En cuanto a características avanzadas, OpenLayers permite una flexibilización del estilo con el que se dibujan las geometrías vectoriales basado en los atributos del objeto que define la geometría. Leaflet también se basa en atributos, pero su capacidad de flexibilizar estilos es algo menor.

Habilidades de Interacción

Dentro de las características primordiales de un mapa, una de las más importantes es la capacidad que tiene de poder interactuar con él, por lo que resulta de utilidad a la hora de decidir qué librería usar, tener en cuenta qué habilidades de interacción nos ofrecen cada una de ellas. En este apartado analizamos la capacidad de cada librería para dar soporte a la interacción que se puede hacer con el mapa. No sólo en versiones web sino también en dispositivos móviles.

En OpenLayers nos encontramos con que este conjunto de actividades se implementan por defecto en el mapa o a través de controles, configurables y extendibles de manera personalizada, que hacen que sea modularizable a la par que configurable.

En Leaflet existen por el contrario un conjunto de funcionalidades también por defecto en el mapa, como el paneo, el zoom mediante la rueda del ratón o el zoom a través de los clics en el mapa, teniendo soporte además para versiones móviles con el multi-touch y el doble tap en el mapa. Además podemos configurar de manera personalizada la funcionalidad de cada uno de los eventos en el mapa, de forma que sea totalmente transparente para el usuario.

En este punto veo mucha similitud entre ambas librerías ya que delegan a la creación de controles, en el caso de Openlayers, o programación de manejadores de eventos, en el caso de Leaflet, para la interacción con el mapa.

Controles del Mapa

De poco sirve un mapa si no se puede interaccionar con él. Esta interacción se lleva a cabo a través de los controles del mapa. Esta puede que sea la característica más diferenciadora entre ambas librerías.

A lo largo de toda su evolución, OpenLayers ha desarrollado y mantiene casi cincuenta controles diferentes para el mapa. Sin embargo, Leaflet tan sólo posee cuatro controles específicamente implementados. Esto no significa en ningún momento que haya funcionalidades que no pueden realizarse en Leaflet, simplemente significa que el desarrollador tendrá que hacer uso de los eventos del mapa para construir sus propios controladores.

Los controles de los que dispone Leaflet son los más básicos: zoom, atribución (que ya hemos visto), visualización de la escala del mapa y selector de capas. Todos los controles que ya vienen desarrollados en OpenLayers, como puede ser la selección de geometrías o la edición de las mismas sobre el mapa, no están explícitamente desarrolladas en Leaflet, aunque sus correspondientes eventos de pulsar sobre unas coordenadas esté implementado.

Llegados a este punto, cabe destacar que aunque no formen parte de la librería, existen multitud de extensiones a Leaflet que suplen este inconveniente al proveer de estas funcionalidades a la librería base. Este punto se discutirá más adelante en la sección de módulos.

Compatibilidad entre navegadores

Al ser una librería más antigua, OpenLayers tiene un mayor soporte entre navegadores antiguos, siendo especialmente destacable su soporte para IE6. Respecto al resto de navegadores, el soporte es ciertamente similar entre ambas librerías.

En nuestra opinión, este punto no marca una diferencia realmente relevante entre ambas librerías, ya que los navegadores que OpenLayers soporta de más son navegadores que ya están fuera de mercado y que por tanto pronto estarán extintos.

Módulos (Extensiones) y Personalización

Una parte importante en cualquier librería es su capacidad de aceptar extensiones o plugins de terceras partes para ampliar su funcionalidad.

Tanto Leaflet como OpenLayers soportan la herencia y extensión de las funcionalidades de controles y tipos de capas de forma sencilla, tan sólo importando en la página en cuestión la librería javascript que la extienda.

Esto es especialmente importante para Leaflet porque, como ya hemos mencionado, OpenLayers tiene más controles y tipos de capas que Leaflet por defecto. El poder añadir extensiones de terceras partes hace que esta diferencia disminuya considerablemente. De hecho, Leaflet posee un repositorio de plugins para poder buscar y seleccionar de forma rápida e intuitiva qué capacidades necesitamos para nuestro mapa.

Esto también hace que aunque la utilización de OpenLayers sea más sencilla, ya que es probable que sólo con importar la librería base ya se tengan todas las funcionalidades que se buscan, también es bien cierto que Leaflet permite construir una librería personalizada sólo con las extensiones deseadas, permitiendo así que se importe una librería de menor tamaño.

En cuanto a extender el estilo de visualización del mapa (colores, iconos), Leaflet hace gala de las últimas capacidades de HTML y CSS3, lo que sin duda lo coloca por delante de OpenLayers en cuanto a simplicidad a la hora de personalizar el aspecto del mapa. Aunque OpenLayers también está basado en CSS y permite, hasta cierto punto, una personalización por sobreescritura de estilos, es cierto que su jerarquía de estilos no es tan sencilla como la de Leaflet.

CONCLUSIONES

Como ya podía imaginarse, el resultado de la comparativa nos lleva a la conclusión más lógica: ambas librerías son válidas y cada una será más útil en diferentes entornos.

Es indudable que OpenLayers ofrece muchos más controles y tipos de capas que Leaflet, por lo que simplifica en gran medida el desarrollo de aplicaciones complejas. Sin embargo, Leaflet ha demostrado ser muy eficiente en entornos donde no es necesaria una interacción compleja con el mapa.

Cabe destacar que en este artículo hemos utilizado la versión 2 de OpenLayers. La siguiente versión, reescrita de cero, ofrece no sólo características muy similares respecto a Leaflet, sino que ambas comunidades están trabajando conjuntamente para compatibilizar extensiones y parseadores de datos [1].

AGRADECIMIENTOS

Especial agradecimiento a la comunidad de software libre, que hacen posible un desarrollo sostenido y sostenible de las nuevas tecnologías.

REFERENCIAS

- ◆ Librería OpenLayers: <http://openlayers.org/>
- ◆ Librería LeafLet: <http://leafletjs.com/>
- ◆ [1] Why we are building OpenLayers 3:
<http://openlayers.org/blog/2012/11/14/why-are-we-building-openlayers-3/>