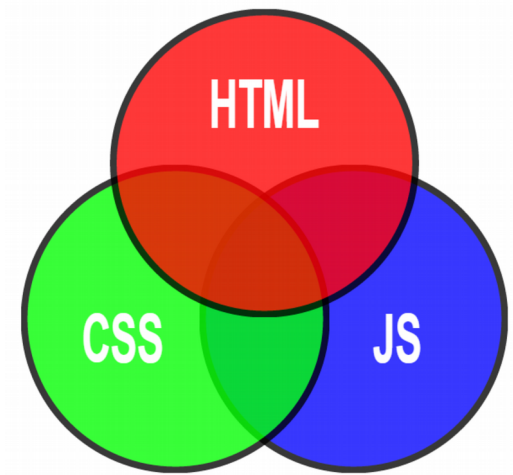


GIS en la Web



Características de una web app

- Herramientas accesibles desde internet
- Independiente del SO, facil de actualizar y mantener
 - Contenido dinamico (HTML, javascript, modo interactivo con el usuario)
- Estructurada en 3 capas:
 - Navegador
 - servidor web
 - BD



HTML

- Estándar del World Wide Web Consortium
 - Interpretado de la misma forma por cualquier navegador web actualizado
- Define una estructura básica y un código para la definición de contenido de una página web, como texto, imágenes, etc.
- Recae en el navegador web la tarea de unir todos los elementos (referencias externas, por ej.) y visualizar la página resultante.



JavaScript

- Lenguaje de programación interpretado, dialecto del estándar ECMAScript
- Orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico
 - Se utiliza principalmente del lado del cliente
- Ampliamente utilizado para enviar y recibir información del servidor



Mapa Web

- El término "mapa web" implica que no está simplemente en la web, sino más bien uno que es alimentado por la web.
- Depende de internet para acceder a los datos que muestra.
- Hoy vamos a presentar un "Demostrador funcional"



Demostrador funcional

- La idea es que puedan ver y usar un proyecto de código minimal.
 - Funcionalidades básicas.
 - No es una aplicación web.
- Faltan la capa intermedia, y la capa de persistencia.
- Vamos a acceder a archivos locales, y a archivos externos (con un Hack).
- ¡Mañana podemos usarlo en los talleres!



Librerías “clásicas” de cartografía

Bibliotecas de JavaScript de código abierto para mostrar mapas interactivos en los navegadores web



OpenLayers



Leaflet

- Es ligero, sencillo y flexible, y es probablemente la biblioteca de mapas de código abierto más popular en este momento.
- funciona bien en la mayoría de los tipos de dispositivos.
- Construido para ser utilizado con múltiples tipos de capas de base.
 - Documentación muy clara
 - Estilo estándar Javascript-plugin



Index.leaflet.html

- Enlaces a otros archivos .js (plugins),
ademas de iconos para los marcadores
- Codigo html con los input para el usuario
- Codigo javaScript para levantar un mapa
leaflet, cargar las capas bases y otras
capas de datos que se vayan agregando.



Ejemplos de funciones Leaflet

- Creacion del objeto mapa:

```
var map = L.map('map').setView([-34.60, -58.38], 4);
```

- Carga mapa base de OpenStreetMap

```
var openStreetMap = L.tileLayer('http://  
{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {  
  maxZoom: 19,  
  attribution: '&copy; <a  
href="http://www.openstreetmap.org/copyright">OpenSt  
reetMap</a>'  
});  
map.addLayer(openStreetMap);
```



Ejemplos de funciones Leaflet

- **Añadir un punto:**

```
var caba = L.marker([-34.60, -58.38]).addTo(map);  
var cord = L.marker([-31.41, -64.18]).addTo(map);
```

- **Añadir una linea**

```
var cabaCordDist = L.polyline([[-34.60, -58.38],  
[-31.41, -64.18]], {color: 'red', weight:  
10}).addTo(map);
```

- **Añadir un poligono**

```
var bariSalRosArea = L.polygon([[-41.14, -71.38],  
[-24.79, -65.40], [-32.94, -60.63]], {color:  
'blue', weight: 4}).addTo(map);
```



Ejemplos de funciones Leaflet

Otros tipo de datos:

- **Camino (path)**
- **MultiPolilinea (multipolyline)**
- **MultiPolygon (multipolygon)**
- **Rectángulo (Rectangle)**
- **Círculo (circle)**
- **Marcador de círculo (circleMarker)**



Ejemplos de funciones Leaflet

- **Añadir Interactividad con Pop-ups**

```
caba.bindPopup("Capital Federal de  
la Republica Argentina.");
```

- **Html como contenido**

```
cord.bindPopup("<img  
src='http://paradiseintheworld.com/w  
p-content/uploads/2012/04/Cathredal-  
of-Cordoba-Argentina.jpg'  
height='300' width='300'>");
```



Fuentes externas

- **addWmsLayer**
- **addKmlLayer**
- **addLocalKmlLayer**
- **addGeoRssLayer**
- **addLocalCsvLayer**



basics_cvs.js

- Levantamos en archivo local “datos.csv”.
 - Parseamos el contenido.
 - Suponemos mismo formato siempre: lat, long, valor en cada linea del archivo.
- Armamos un marcador, con la posicion y el valor lo insertamos dentro del contenido de un popup.
 - Insertamos todo en un group layer.
 - Agregamos al mapa.



basics_georss.js

- Similar comportamiento, pero accediendo a una fuente externa por url
- Parseo del feed (usamos una biblioteca)
 - Marcadores a un group layer
 - Agregamos al mapa el group layer
- En la actualidad, es mejor ir pensando en otros estandares como GeoJSON



basics_kml.js

- Mas facil que las dos variantes anteriores.
 - Leaflet es el responsable de hacer la lectura de la respuesta y del armado del layer.
- Podemos levantar tambien un kml local



basics_wms.js

- Misma logica que con un kml.
- Leaflet es el responsable de hacer la lectura de la respuesta y del armado del layer.



Usos de basicWebMap

Hay dos modos de uso:

- Agregando las capas desde la interfaz(?) de usuario.
 - Agregando el código “a mano”
(ambas terminan llamando a la misma función)

La última variante es mejor cuando ya eligieron las capas a usar, ya que como no hay BD, les queda “hardcodeado”.



¿Por qué usamos un Proxy?

- Cross-origin resource sharing (CORS) Error
- Ciertas solicitudes de "dominio cruzado", en particular las solicitudes Ajax, están prohibidas por defecto por política de seguridad.
 - Un atacante proporciona una URL incluyendo código malicioso
- Para evitarlo, pasamos por un servidor proxy



¿Consultas?

