# Wikilocs

Creación de las rutas para subir a la página. También uno puede descargar la aplicación Android y subir su ruta a la página web para luego ser descargada.

# Mapas Mapbox.js

Descargar el js de mapbox y de leaflet, este último es una librería que nos proporciona las herramientas necesarias para añadir un mapa a nuestra web.

También podemos utilizar el CDN ([Content delivery network](http://en.wikipedia.org/wiki/Content_delivery_network))

<!-- include method omnivore para gpx, kml-->

<script src='https://api.tiles.mapbox.com/mapbox.js/plugins/leaflet-omnivore/v0.2.0/leaflet-omnivore.min.js'></script>

Para poner la ruta gps en el mapa.

Al añadir los mapas a la web, por definición de la librería mapbox deber tener la posición absolute.

Instalar WAMP en Windows

Crear varios mapas para ir cambiando el diseño.

Problemas con TileMill, para hacer un mapa en el cual se puedan ver las calles se necesita un zoom de 15, y crear un fichero así supone más de 1GB de información. Además MapBox en la cuenta gratuita me deja subir solo hasta 100MB. EXPLICAR NIVELES DE ZOOM.

# Static Image de un mapa

<!-- include the geo-viewport plugin -->

<script src='https://api.tiles.mapbox.com/mapbox.js/plugins/geo-viewport/v0.1.1/geo-viewport.js'></script>

Para poner las imágenes estáticas se debe tener en cuenta que las coordenadas según el sistema estadounidense son al revés.

Obtener imagen estática del mapa editado, current. Se tendrá que mostrar la imagen del mapa con la ruta añadida como una línea poligonal.

<https://www.mapbox.com/mapbox.js/example/v1.0.0/polyline/>

Se debe abrir y leer la ruta gpx obteniedo la latitud y longitud y establecer la línea poligonal.

Leer fichero:

<http://www.mosync.com/files/imports/doxygen/latest/html5/filereader.md.html> (Desde fichero)

<http://www.html5rocks.com/es/tutorials/file/dndfiles/> (input)

Para obtener los datos del fichero, a modo de prueba usaré un fichero local y leere el fichero gpx(xml) con php. ~~La major opción es que cuando suba el fichero en gpx cambiarlo a txt.~~

Se ha cargado a modo de prueba el fichero ruta1.gpx mediante php para poder obtener la imagen con la ruta hecha a través de una línea poligonal. Se ha añadido la línea y coincide con la ruta gps, ahora obtener la imagen. (Se podría dibujar la línea como plyline).

Hay que codificar las coordenadas, cambiarlas a un formato reconocible para mapbox.

*https://api.tiles.mapbox.com/v4/gonzalito.78706231/path-4+026-0.75%28shjdGh|\_I\_Ej@KaCq@%7BAg@t@q@%60Ay@GiANo@%60@uAVqABw@LQ|@AhAAbAEz@iAV%7DBPkAEi@%5C%29/-1.6470424999999977,42.82496919182301,15/800x600.png?access\_token=pk.eyJ1IjoiZ29uemFsaXRvIiwiYSI6IlVJTGIweFUifQ.waoF7m8PZbBM6u8Tg\_rR7A*

Cambiar los paréntesis.

La codificación de google funciona bien para algunas coordenadas, puede ser por la cantidad de puntos o las coordenadas en si. El ejemplo de mapbox funciona correctamente pero un ejemplo creado con una ruta más larga y en pamplona funciona hasta cierto punto.

Trabajaremos con php para que el usuario pueda subir sus rutas. Se ha creado simple para poder subir una ruta.

La idea es luego de obtener la imagen abrir una nueva pestaña en la cual aparezca la imagen como textura de ciertos objetos 3D los cuales el usuario pueda imprimir en 3D.

# Manejo de eventos

L.mapbox.accessToken = 'pk.eyJ1IjoiZ29uemFsaXRvIiwiYSI6IlVJTGIweFUifQ.waoF7m8PZbBM6u8Tg\_rR7A';

var map = L.mapbox.map('map', 'gonzalito.78706231').setView([40, -74.50], 9);

map.on('ready', function() {

// the layer has been fully loaded now, and you can

// call .getTileJSON and investigate its properties

console.log("Mapa cargado correctamente");

});

map.on('zoomend', function() {

console.log("zoom:" + map.getZoom());

});

# Data en SideClient

Manejaremos la información en el lado del cliente, sino el servidor se nos llenaría de basura. Luego de usar toda la información sería preciso eliminar toda la información guardad en el lado del cliente. Otra opción que se puede manejar es usar PHP (get o post).

SessioStorage se elimina si se cierra el navegador, localStorage se mantiene.

# Cosas

* Color en la ruta.
* Subir ficheros gpx y cargarlos mediante drop.

# Implementación *three.js*

Añadir control con Detector, para saber si el navegador soporta webGL.g

Luego de tener el mapa ya en imagen, para la textura de nuestro objeto 3D. Se creará un interfaz para elegir el diseño base: poster, bolso, reloj de pared, entre otros.

Luego se cargará la textura en el objeto.

Error al cargar una textura: Uncaught TypeError: Failed to execute 'texImage2D' on 'WebGLRenderingContext': No function was found that matched the signature provided.

Nuevos objetivos: con datos de terreno se intentará crear un relieve correspondiente a la zona en formato hgt usando *three.js*.

# Terreno con elevación

Nuevos objetivos: con datos de terreno se intentará crear un relieve correspondiente a la zona en formato hgt.

GeoTIFF es un [estándar](http://es.wikipedia.org/wiki/Normalizaci%C3%B3n) de [metadatos](http://es.wikipedia.org/wiki/Metadato) de domino público que permite que información [georreferenciada](http://es.wikipedia.org/wiki/Georreferenciaci%C3%B3n) sea encajada en un archivo de imagen de formato [TIFF](http://es.wikipedia.org/wiki/TIFF). La información adicional incluye el tipo de [proyección](http://es.wikipedia.org/wiki/Proyecci%C3%B3n), [sistema de coordenadas](http://es.wikipedia.org/wiki/Sistema_de_coordenadas), [elipsoide](http://es.wikipedia.org/wiki/Elipsoide), [datum](http://es.wikipedia.org/wiki/Datum) y todo lo necesario para que la imagen pueda ser automáticamente posicionada en un sistema de referencia espacial. El formato GeoTIFF es completamente compatible con TIFF 6.0, por lo que un programa informático incapaz de leer e interpretar esa información podrá aun así abrir el archivo de imagen GeoTIFF y visualizarlo como si de un archivo TIFF normal se tratara.

Leer el fichero y ver como obtener los datos reales del terreno. Los documentos HGT son Archivos GIS asociados con Shuttle Radar Topography Mission Data (SRTM).

gdal para transformar de hgt to geoTiff

20-23 obtención tiff, para qué? Problemas en ordenador de la uni por las librerías c++, transformar con portátil personal y usando gdaldem:

<http://blenderartists.org/forum/showthread.php?294036-using-real-world-data-(DEM-files)-to-generate-3d-maps>

Editar imagen de elevación de terreno:

<http://www.mikejcorey.com/wordpress/2011/02/05/tutorial-create-beautiful-hillshade-maps-from-digital-elevation-models-with-gdal-and-mapnik/>

Obtener información del fichero:

* Gdallocationinfo
* Matlab (error dice que la función no existe “geotiffread” versión 2014)
* Usando Python <http://gis.stackexchange.com/questions/29632/raster-how-to-get-elevation-at-lat-long-using-python/29634#29634>
* ¿ Digital Elevation Model (DEM)?
* Javascript <http://jaanga.github.io/terrain-plus/cookbook/read-hgt-files/readme-reader.html>
* Completo ejemplo javascript leer elevación: <http://jaanga.github.io/terrain-plus/cookbook/hgt-viewer/r6/hgt-viewer-r6.html> (Más simple de lo esperado), no es necesario pasar la imagen a geotiff ya que el fichero hgt contiene la elevación del terreno. (4 días para nada!). Las elevaciones son todas de 1201 por 1201, hay que tener en cuenta que un array de javascript va de 0 a 1200 = 1201 elementos. Hay problemas ya que el plano que se crea debe tener muchos segmentos para la creación de un terreno óptimo y a mi navegador con 300 segmentos le cuesta alrededor de 5 segundos. Imagen muy grande, redimensionar.  
    
  Problemas  
  + Crear una función que disminuya el tamaño de los ficheros.
  + Si el plano tiene muchos segmentos provoca un terreno muy disperso, la idea sería disminuir el tamaño del fichero(array) y hacer un plano con pocos segmentos y cambiar la posición del vértice, como el *PFCGTerreno.html* .
  + Imágenes (*segmentos1.png* y *segmentos2.png*)
  + Cambiar escala porque salen valores muy altos. *segmentos3.png*
  + Dividí los valores entre 10 y se obtiene un plano aceptable donde se puede apreciar bien la variación del relieve. *segmentos4.png*

Ya se tiene el terreno con la elevación, para ir probando se añadirá una textura, de las que tiene por defecto la librería three.js