

Diego Coterá Gonzalo

71651642H

UO140155

PRÁCTICA FINAL

Sistemas de
Información
Geográfica y
Servicios Web

Máster de Ingeniería Web – Universidad de Oviedo

1. Definición del proyecto de sistema de información geográfica

Objetivos

Desarrollo de una aplicación web que permita a los organizadores de pruebas deportivas evaluar posibles recorridos de carrera, mediante un mapa obtener perfiles, situar puntos donde colocar banderines de balizamiento, comparar con otras rutas, conocer los nombres de los puntos y las zonas de paso, accidentes geográficos, etc.

Ámbito y límites del proyecto

Este prototipo pretende ser una herramienta modular, con un fin concreto, como se expone en los objetivos, servir como evaluador de recorridos en actividades al aire libre. En un futuro este módulo ha de formar parte de un sistema mayor, un gestor de eventos deportivos.

Tecnologías empleadas en el desarrollo

HTML5, CSS, JavaScript, JQuery, Semantic-UI, servidor Apache.

Sistema de coordenadas

La tierra tiene forma sensiblemente esférica, y la aproximamos mediante un elipsoide un poco achatado en los polos. WGS84 es un Datum, esto es, un tamaño de elipsoide concreto situando el centro en el tomado por convenio de la tierra. Google utiliza las coordenadas de longitud y latitud referidas a ese datum.

Servicios externos

- **Api de Google Maps**
- **Servicios WMS**

Para obtener las capas a georeferenciar con los recursos de GoogleMaps se utilizan principalmente los servicios del Instituto Geográfico Nacional, que se nutre de menor a mayor escala de: cartografía procedente del Instituto Geográfico Nacional y el Centro Geográfico del Ejército de Tierra (BTN100), Instituto Geográfico Nacional (BTN25, NGBE, SIGLIM), Sistema Cartográfico Nacional (IGR-RT, IGR-H y CartoCiudad), DG del Catastro.

Se descubren los recursos de los servicios con:

<http://www.ign.es/wms-inspire/ign-base?amp;request=GetCapabilities>

En este prototipo se utilizan 2 capas:

- Layer IGNBaseTodo-nofondo

Representación cartográfica desde escalas pequeñas hasta 1:4.000 de nombres geográficos, redes de transporte y sus infraestructuras, espacios naturales, culturales y arqueológicos, hidrografía, relieve, núcleos urbanos, islas, manzanas, edificios y direcciones. No contiene la representación del mar ni de los países.

WMS_layer1 = 'IGNBaseTodo-nofondo'

- Layer nombres geográficos

Siendo la capa: WMS_layer1 = 'IGNBaseTodo-nofondo'

2. Descripción de la funcionalidad que se implementará en el prototipo

La aplicación web facilita a los organizadores de pruebas deportivas evaluar posibles recorridos de carrera, mediante un mapa obtener perfiles, situar puntos donde colocar banderines de balizamiento, comparar con otras rutas, conocer los nombres de los puntos y las zonas de paso, accidentes geográficos, etc.

Requisitos básicos

API de Maps: Mapas, botones, toggle

Puntos geográficos: Rutas de montaña kml grabadas propias y balizamiento

Repositorio WMS: <http://www.ign.es/wms-inspire/ign-base?request=GetCapabilities>

Servicios de Maps: Mapas, perfil con altitud y charts

Interacción con y del usuario: Selectores, menús, checkboxes y otros elementos.

3. Diseño del prototipo y cuestiones de implementación

Desarrollo de un sitio web típico html5 y presentación visual con css y framework Semantic UI.

Lenguaje de programación JavaScript, empleando recursos y librerías (o bibliotecas) de JQuery, Api de Google...

Código autoexplicativo con comentarios

```
var map;

var infowindow;

var centro;

var markerActual = null;

var lineaElevacion;

var elevator;

var chart;

var marcas;

var kmls = ['http://156.35.98.19:8080/kml1.kml',
'http://156.35.98.19:8080/kml2.kml', 'http://156.35.98.19:8080/kml3.kml' ];

google.load('visualization', '1', {packages: ['columnchart']});

var bandera =
'https://developers.google.com/maps/documentation/javascript/examples/full/images/beachflag.png';

// Inits de WMS
var WMS_URL = 'http://www.ign.es/wms-inspire/ign-base?';
var WMS_layer1 = 'EL.ContourLine';
var WMS_layer2 = 'caminos';
var WMS_layer3 = 'PS.ProtectedSite';
var WMS_layer4 = 'GN.GeographicalNames';

function initMap() {
    centro = new google.maps.LatLng(43.370840, -5.842143);
    var misOpciones = {
        center: centro, // Obligatorio, objeto LatLng (grados decimales)
```

```
    zoom: 14, // Obligatorio, de 0/1 a 20
    mapTypeId: google.maps.MapTypeId.TERRAIN, // Opcional, ROADMAP,
SATELLITE, HYBRID o TERRAIN
    streetViewControl: false,
    fullscreenControl: false,
    mapTypeControl: false,
    disableDoubleClickZoom: false
};
elevator = new google.maps.ElevationService;

map = new google.maps.Map(document.getElementById("map_frame"),
misOpciones);

// Linea de elevaciones
lineaElevacion = new google.maps.Polyline({
    strokeColor: '#ff0000',
    strokeOpacity: 1.0,
    strokeWeight: 3
});
lineaElevacion.setMap(map);

google.maps.event.addListener(map, 'rightclick', function (event) {
    dibujarLinea(event.latLng);
});

google.maps.event.addListener(map, 'click', function (event) {
    borrarLinea();
});
// Fin linea de elevaciones

// KML
var kmlLayer = new google.maps.KmlLayer({
    url: kmls[0],
```

```
    suppressInfoWindows: true, // Permite mostrar nuestros iconos (fotos
incluidas)
```

```
    preserveViewport: false,
```

```
    map: map
```

```
});
```

```
google.maps.event.addListener(kmlLayer, 'click', function (event) {
```

```
    if (markerActual) markerActual.setMap(null);
```

```
    markerActual = situarMarcador(event.latLng);
```

```
});
```

```
$("#toggleKML").change( // Si se marca o desmarca un checkbox de WMS,
ponemos o quitamos el contenido y recargamos
```

```
function () {
```

```
    if ($(this).is(':checked')) {
```

```
        kmlLayer.setMap(map);
```

```
    } else kmlLayer.setMap(null);
```

```
});
```

```
$('#kmls').change(function() {
```

```
    kmlLayer.setUrl('http://156.35.98.19:8080/kml'+$(this).val()+'kml')
```

```
});
```

```
// Fin KML
```

```
// Marcas
```

```
if ( localStorage.getItem("marcas") === null) {
```

```
    marcas = [];
```

```
    localStorage.setItem("marcas",JSON.stringify(marcas));
```

```
} else marcas = JSON.parse(localStorage.getItem("marcas"));
```

```
ponerMarcas();
```

```
// WMS
```

```
var overlayOptions = {
```

```

    getTileUrl: TileWMS, // Invoca a la funcion que pide la capa WMS
    tileSize: new google.maps.Size(256, 256)
};

var overlayWMS = new google.maps.ImageMapType(overlayOptions); // Creamos
el overlay

map.overlayMapTypes.push(overlayWMS); // Ponemos el overlay en el mapa

$("input:checkbox").change( // Si se marca o desmarca un checkbox de WMS,
ponemos o quitamos el contenido y recargamos

function () {
    if ($(this).is(':checked')) {
        if ($(this).attr("id") === "toggleCurvas") {
            WMS_layer1 = 'EL.ContourLine';
        } else if ($(this).attr("id") === "toggleCaminos") {
            WMS_layer2 = 'caminos';
        } else if ($(this).attr("id") === "toggleProtected") {
            WMS_layer3 = 'PS.ProtectedSite';
        } else if ($(this).attr("id") === "toggleNombres") {
            WMS_layer4 = 'GN.GeographicalNames';
        }
    } else {
        if ($(this).attr("id") === "toggleCurvas") {
            WMS_layer1 = "";
        } else if ($(this).attr("id") === "toggleCaminos") {
            WMS_layer2 = "";
        } else if ($(this).attr("id") === "toggleProtected") {
            WMS_layer3 = "";
        } else if ($(this).attr("id") === "toggleNombres") {
            WMS_layer4 = "";
        }
    }
}

map.overlayMapTypes.removeAt(0); // Quitamos el overlay

```



```
        map.overlayMapTypes.push(overlayWMS); // Lo recargamos con las capas
actualizadas
    });
    // Fin WMS
}
```

```
function dibujarLinea(coords) {
    var pathOriginal = lineaElevacion.getPath();
    var path = [];
    pathOriginal.push(coords);
    for (var i = 0; i < pathOriginal.length; i++) {
        var latlng = new google.maps.LatLng(pathOriginal.getAt(i).lat(),
pathOriginal.getAt(i).lng());
        path[i] = latlng;
    }
}
```

```
if (path.length >= 2) {
    elevator.getElevationAlongPath({
        'path': path,
        'samples': 64
    }, plotElevation);
}
}
```

```
function borrarLinea() {
    lineaElevacion.getPath().clear();
}
```

```
/**
```

```
* @return {string}
```

```
*/
```

```
var TileWMS = function (coord, zoom) { // Crea la URL de llamada obteniendo datos
del mapa y juntando los que le seteamos
```

```

var proj = map.getProjection();
var zfactor = Math.pow(2, zoom);

var top = proj.fromPointToLatLng(new google.maps.Point(coord.x * 256 / zfactor,
coord.y * 256 / zfactor));

var bot = proj.fromPointToLatLng(new google.maps.Point((coord.x + 1) * 256 /
zfactor, (coord.y + 1) * 256 / zfactor));

var bbox = top.lng() + "," + bot.lat() + "," + bot.lng() + "," + top.lat();

var layers = $.grep([WMS_layer1, WMS_layer2, WMS_layer3, WMS_layer4],
Boolean).join(","); // Une las variables no vacias separandolas por comas

var myURL = WMS_URL +
"SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&SRS=EPSG
%3A4326&WIDTH=256&HEIGHT=256&FORMAT=image/png&TRANSPARENT=TRUE
";

myURL += "&LAYERS=" + layers;
myURL += "&BBOX=" + bbox;
return myURL;
};

```

```

function situarMarcador(localizacion) { // Crea un marcador y lo pone

var marker = new google.maps.Marker({
    position: localizacion,
    animation: google.maps.Animation.DROP,
    title: "Tu selección",
    icon: bandera,
    draggable: true,
    map: map
});

var html = "<div id='content'>\n" +
" <div id='siteNotice'></div>\n" +
" <h1 id='firstHeading' class='firstHeading'>Corredor</h1>\n" +
" <div id='bodyContent'>\n" +
"<div id='marca'>" +

```

```

"    <table><tbody>" +
"    <tr><td align='right'><label class='uk-form-label' for='crono'>Tiempo:
</label> </td>\n" +
"    <td><input class='uk-input' type='text' name='crono' id='crono' value='' +
document.getElementById('chronotime').textContent +"/></td>\n" +
"    </tr>" +
"    <tr><td align='right'><label class='uk-form-label' for='lugar'>Lat: </label>
</td>\n" +
"    <td><input class='uk-input' type='text' name='lugar' id='lugar' value='' +
localizacion +"/></td>\n" +
"    </tr>" +
"    <tr><td align='right'><label class='uk-form-label' for='name'>Nombre:
</label> </td>\n" +
"    <td><input class='uk-input' id='name' type='text' size='12' /></td>\n" +
"    </tr>" +
"    <tr><td align='right'>" +
"    <label class='uk-form-label' for='file'>Foto: </label> </td>\n" +
"    <td><input class='uk-input' type='file' id='inputFile' /> </td>\n" +
"    </tr><tr><td align='right'></td><td><input id='submit' value='Marcar'
type='submit' onclick='submitFoto()'></td></tr>" +
"    <tr><td align='right'></td><td><img id='image_upload_preview'
style='max-width: 150px; max-height:150px;' src='http://placeholder.it/100x100'
alt='your image' /></td></tr>" +
"    </tbody></table>" +
" </div>\n" +
" </div>\n" +
"</div>";
if (infowindow) infowindow.close();
infowindow = new google.maps.InfoWindow({
    content: html, maxWidth: 400
});
infowindow.open(map, marker);
$("#inputFile").change(function () {
    readURL(this);

```

```

});
marker.addListener('click', function () {
    infowindow.open(map, marker);
});
return marker;
}

function submitFoto() {
    if (document.getElementById('name').value === "") {
        document.getElementById('name').focus();
        return false;
    }
    if (document.getElementById('crono').value === "") {
        document.getElementById('crono').focus();
        return false;
    }
    if (document.getElementById('lugar').value === "") {
        document.getElementById('lugar').focus();
        return false;
    }
    var crono = document.getElementById('crono').value;
    var nombre = document.getElementById('name').value;
    var lugar = document.getElementById('lugar').value;
    lugar = lugar.replace(/\\(\\)/gi, "");
    lugar = lugar.split(",");
    guardarDatos(crono, nombre, lugar[0], lugar[1]);
    return false;
}

function guardarDatos(crono, nombre, lat, lon){
    marcas = JSON.parse(localStorage.getItem("marcas"));
    marcas.push([crono,nombre, lat, lon]);
}

```

```
localStorage.setItem("marcas", JSON.stringify(marcas));  
}
```

```
function ponerMarcas(){  
  var markersG;  
  marcas.forEach(function(element) {  
    markersG = new google.maps.Marker({  
      position: new google.maps.LatLng(element[2],element[3]),  
      animation: google.maps.Animation.DROP,  
      title: "Corredor: " + element[1] + "\nPunto:" + element[2] + "," + element[3] +  
      "\n Crono: " + element[0],  
      icon: bandera,  
      draggable: true,  
      map: map  
    });  
  });  
}
```

```
function readURL(input) {  
  if (input.files && input.files[0]) {  
    var reader = new FileReader();  
    reader.onload = function (e) {  
      $('#image_upload_preview').attr('src', e.target.result);  
    };  
    reader.readAsDataURL(input.files[0]);  
  }  
}
```

```
// Takes an array of ElevationResult objects, draws the path on the map  
// and plots the elevation profile on a Visualization API ColumnChart.  
function plotElevation(elevations, status) {
```

```
var chartDiv = document.getElementById('elevation_chart');
if (status !== 'OK') {
    // Show the error code inside the chartDiv.
    chartDiv.innerHTML = 'Cannot show elevation: request failed because ' +
        status + "The elevations are:" + elevations;
    return;
}
// Create a new chart in the elevation_chart DIV.
var chart = new google.visualization.ColumnChart(chartDiv);

// Extract the data from which to populate the chart.
// Because the samples are equidistant, the 'Sample'
// column here does double duty as distance along the
// X axis.
var data = new google.visualization.DataTable();
data.addColumn('string', 'Sample');
data.addColumn('number', 'Elevation');
for (var i = 0; i < elevations.length; i++) {
    data.addRow(["", elevations[i].elevation]);
}

// Draw the chart using the data within its DIV.
chart.draw(data, {
    height: 150,
    legend: 'none',
    titleY: 'Elevación (m)'
});
}
```

4. Uso y despliegue de la aplicación

Despliegue de la aplicación

Disponible en la url:

<http://elvertical.com/>

Despliegue en local

Descomprimir el archivo **"trailevaluator.zip"** en la ruta que se elija.

Con cualquier navegador moderno se puede utilizar la aplicación web, accediendo mediante el archivo "index.html".

Captura de pantalla de la aplicación. Muy intuitiva de uso y con cuadros explicativos de uso.

Inicio Info Contacto

RaceManager - TrailEvaluator

Opciones de capas WMS

- Activa y desactiva capas WMS
 - Utiliza los siguientes checkboxes
- IGN Base
- Nombres geográficos

Rutas

- Selecciona una ruta kml precargada
 - Utiliza este selector

Covadonga - Urdiellu

Mapa

Dibuja tu recorrido obteniendo dinámicamente el perfil :-)

- Click con el botón derecho del ratón en el mapa: Para ir dibujando la polilínea de tu recorrido (mínimo 2 puntos).
- Doble-click con el botón izquierdo del ratón en el mapa: Borrar la polilínea del recorrido dibujado.

Señala los puntos de la ruta con banderas

- Click con el botón izquierdo sobre la ruta kml cargada: Automáticamente caerá y quedará fijada en un punto una bandera.

Perfil

Elevación (m)

2,500
2,000
1,500
1,000
500
0

5. Referencias

- Apuntes de la asignatura Sistemas de Información Geográfica y Servicios Web del Máster en Ingeniería Web de la Universidad de Oviedo.
- Documentación del Instituto Geográfico Nacional.
- Tutoriales, manuales y otros materiales de la API de GoogleMaps