

# TREBALL PRÀCTIC FINAL: ITINERARI GEOINFORMÀTICA I PROGRAMACIÓ SIG

Programació d'aplicacions per l'obtenció, anàlisis i visualització de dades  
geogràfiques

*Máster profesional en SIG*

*UNIGIS*

Magí Pàmies Sans

10 de setembre de 2019

# Índex

1	Descripció . . . . .	1
2	Objectius . . . . .	2
3	Processos duts a terme . . . . .	4
4	Dificultats . . . . .	8
5	Conclusions . . . . .	9

# 1 Descripció

Aquest treball és una informe del mapa web interactiu creat com a Treball Final de Màster del Màster professional en SIG de la UNIGIS.

L'informe s'estructura en cinc apartats: el primer(l'actual), anomenat *Descripció*, per fer una breu descripció de d'aquest document. El segon, anomenat *Objectius*, explica quins eren els objectius plantejats abans de realitzar el mapa, tant a nivell general com a un nivell més detallat i tècnic. El tercer, *Processos duts a terme*, desgrana pas a pas tot el procés seguit per a desenvolupar el mapa. El quart, *Dificultats*, mostra quins problemes m'he trobat a l'hora de realitzar el treball i com els he pogut solucionar. Per últim el cinquè, *Conclusions*, hi trobareu una petita reflexió sobre el que m'ha comportat realitzar el treball i que n'he tret de tot plegat.

## 2 Objectius

L'objectiu final del treball és obtenir un mapa web interactiu de la ciutat de Reus a on s'hi pugui veure quines són les zones més visitades pels turistes.

Per poder-ho mostrar he utilitzat dos tipus de capes. Una vectorial de punts a on s'hi representen tots els restaurants i una altra de tipus *HeatMap* a on s'hi mostren la densitat de les fotografies publicades a Flickr.

La capa dels restaurants ens permet fer-nos una idea de quines zones són les preferides per als turistes per anar-hi a dinar, òbviament que estaran en estreta relació en la zones que més freqüenten.

Per altra banda la capa *HeatMap* ens permet veure la distribució de les zones a on els turistes hi fan més fotografies.

Si ho haguéssim de dir amb un vocabulari més tècnic, podríem dir que *l'objectiu principal d'aquest treball és desenvolupar un mapa web interactiu a partir de dades extretes de fons diverses (Flickr, WMS, opendata, etc) que necessiten processos d'extracció, tractament i interacció que he après en aquest màster.*

Un cop tenim clars quin és l'objectiu principal, convé detallar quins són els objectius situats a nivells inferiors que ens han de permetre arribar a l'objectiu final. Els podríem esquematitzar de la següent manera:

- Saber utilitzar una base de dades per alimentar una aplicació web.
- Ser capaç de publicar un servei OGC (GeoServer) i utilitzar-lo en un mapa web.
- Ser capaç de utilitzar una API de codi obert (Openlayer) per la visualització de dades geoespecials a la web.
- Saber desenvolupar, mitjançant el codi de programació Python, un script que ens permeti obtenir dades online a través de la API de Flickr.
- Fer servir el codi de programació Python per importar dades a una base de dades espacial.

El primer objectiu fa referència al fet de poder incorporar al mapa web una capa creada a partir de les dades d'una base de dades. En aquest cas crearem una taula amb les dades de les

fotos de la API de Flickr.

El segon objectiu en el nostre cas consistirà en crear la capa *Heatmap* agafant les dades de la base de dades i publicar-la mitjançant un WMS. El fet d'utilitzar un servei OGC ens permet donar-li dinamisme a la capa, ja que modificant els valors de la base de dades la capa s'actualitzarà automàticament.

El tercer objectiu fa referència a saber utilitzar una API de codi obert, en el nostre cas Openlayer, per poder incorporar les capes al mapa i dissenyar el format del mapa que s'adjuntarà al codi html que crea la pàgina web.

El quart objectiu fa referència a obtenir les dades de la API de Flickr mitjançant un script de Python. És a dir que al executar aquest script es descarreguin les dades.

Per últim, el cinquè objectiu està relacionat amb l'anterior, ja que fa referència a incorporar aquestes dades a una base de dades. Per fer-ho, s'ha de codificar un script en Python.

### 3 Processos duts a terme

En relació a quins han sigut els processos duts a terme per a poder realitzar el mapa web, els podem diferenciar entre:

- **CREACIÓ DE LA CAPA HEATMAP**

- *Descargar les dades de la API Flickr*
- *Disseny de la base de dades*
- *Pujar les dades a la base de dades*
- *Crear la capa HeatMap*
  - \* *Obtenir les dades de la base de dades*
  - \* *Crear la capa*
  - \* *Dissenyar la HeatMap*

- **DISSENY DEL MAPA WEB**

- *Estructurar la pàgina web(html)*
- *Dissenyar i configurar el mapa*
- *Incorporar les capes al mapa*
- *Disenyar la pàgina web (css)*

Per aconseguir les dades de la API de Flickr i per pujar-les a la base de dades he creat un script de Python conjunt. És a dir que al executar-lo es produïen els dos processos. Ho he fet simplement per temes de comoditat.

Les dades les he baixat mitjançant el mòdul *flickrapi* que simplifica molt el procés. Li he introduït les coordenades d'on havia d'extreure les dades, també la data a partir de quan les havia d'agafar (01-01-2016). Les he descarregat en format *json*. Per poder tenir aquestes dades emmagatzemades, el que he fet ha sigut primer fer una primera consulta de les dades per saber quantes pàgines tenia. Així he pogut crear una variable amb aquest valor i posteriorment crear una estructura de control tipus *while* per anar descarregant les dades pagina per pàgina. Les dades es guardaven en una llista en forma de tupla. Aquesta tupla contenia tres valors:

*id*, *longitude*, *latitude*. El primer serveix per poder identificar a cada element i els altres dos serviran per georeferenciar el punt.

Un cop la llista ja està plena amb els dades descarregades de la API Flickr, cal pujar aquestes dades en una base de dades. Per dissenyar la base de dades he fet servir el llenguatge SQL i el software *PgAdmin*. He creat una base de dades amb el nom de *tfm* i li he instal·lat la extensió espacial *PostGIS* per tal de poder georeferenciació les dades. Posteriorment he creat una taula, que serà a on s'hi carregaran les dades. Aquesta taula té quatre columnes, les tres primeres són els tres valors de la tupla i la quarta és la que servirà per crear el punt geomètric de cada valor. Per fer aquesta última, cal indicar-li que és un objecte geomètric i que està referenciada en un sistema de referència 4326. També he indicat que la columna *id* és la que serà la clau.

Creada la taula, cal tornar al scrip de Python per codificar el carregament de les dades que tenim en una llista a la base de dades. Per fer-ho he fet servir el mòdul *psycopg2*. Mitjançant aquest, és crea una connexió amb la base de dades (cal introduir el nom de la base de dades, el host i les dades de l'usuari). Després instauem una variable que serà un cursor que ens servirà per poder executar les funcions SQL dins del script de Python. Aquestes són primer esborrar tots els registres que té la taula (per poder-ne introduir de nous), després introduir les dades que conté la llista (fins a 1250 elements) i per ultim generar l'objecte geomètric, en aquest cas un punt, a la quarta columna. Per fer aquest pas he utilitzat la funció *ST\_MakePoint* dins de la funció *ST\_SetSRID* configurada en un sistema de referencia 4326.

Per acabar l'script, només cal tancar tant el cursor com la connexió.

Un cop les dades carregades a la base de dades, el següent pas era utilitzar un servei OGC per crear una capa que posteriorment utilitzarem amb el mapa. Per fer-ho he utilitzat GeoServer. Allí he creat un espai de treball i en aquest hi he creat un magatzem de dades de tipus PostGis. Aquest l'he connectat a la base de dades a on hi havia penjat les dades. Per últim hi he creat una capa amb les dades de la taula i assignat un disseny que prèviament havia creat, al mateix GeoServer, per tal que la capa tingués format de HeatMap.

Un cop creada la capa, ja estava disponible per a ser inserida al mapa. Per tant ara cal passar a l'altre part del procés, que és la creació de propi mapa.

Com que volia utilitzar la llibreria i a la vegada entorn d'execució *Node.js*, l'hi he hagut de crear un projecte nou. Un cop creat i configurat, ja he pogut crear el fitxer *tmf.html* que serveix

per configurar la pròpia pàgina web. A part d'aquest arxiu, també n'he creat un de Java per configura el mapa, tres de CSS per dissenyar la web (*tfm.css*), la finestra emergent que apareix quan cliques la icona d'un restaurant (*popup.css*) i l'intercanviador de capes (*layerswitcher.css*). Per tal de tenir ordenats els fitxers, els he ordenat en carpetes. Una per l'arxiu Java, una altre pels arxius CSS i una altre per la icona dels restaurants (*imatges*).

El fitxer html hi he relacionat els tres fitxers CSS de disseny i el disseny de l'Openlayers. A nivell de cos, hi he introduït una capçalera amb el títol del mapa i el meu nom. Després un element a on hi anirà el mapa (que ocuparà la major part de la pantalla). Un altre element que serà el requadre emergent que sorgirà quan cliquis un restaurant i per últim un requadre a la part inferior de la pantalla a on hi aniran les coordenades del punt del ratolí. Per acabar també hi he introduït un link a la API *jQuery*, que ens servirà per poder fer servir la tecnologia AJAX, i un altre link al codi Java que he codificat.

Pel que fa al codi Java és a on hi ha la configuració del mapa. El mapa té dues capes base (una ortofoto i una altre topogràfica) i dues capes sobreposades (els restaurants i la HeatMap). Per poder-les gestionar he utilitzat el *LayerSwitcher*, ajuntant primer totes les capes en una sola variable que conté dos *LayerGroup*, una per a cada tipus de capa. En el cas de les capes base, el mapa només te'n deixarà seleccionar una i a la vegada sempre n'hi haurà d'haver un d'activada. Per altra banda les capes sobreposades, es podran activar les dues a la vegada o només una o cap.

A més a més del *LayerSwitcher*, el mapa incorpora altres elements com són un minimapa (*OverviewMap*), l'opció de veure'l a pantalla completa (*FullScreen*), el fet que ens digui les coordenades del ratolí (*Mouse Position*) i una escala gràfica (*ScaleLine*). Tots aquests elements estan definits dins del fitxer JAVA.

Per altra també he hagut de definir les configuracions bàsiques del mapa com són el centre del mapa per a la visualització inicial, el zoom inicial i el seu màxim i mínim i els límits fins a on és pot veure el mapa.

Per introduir les capes base, les dos de tipus *TileLayer*, utilitzant servidors online. La topogràfica utilitza el OpenStreetMap i la ortofoto l'ArcGIS. En el cas de la primera utilitza un fon *OSM*, mentre que la segona una genèrica *XYZ*.

Pel que fa a les capes sobreposades, cada una és d'un tipus diferent. Pel que fa a la capa



HeatMap és del tipus *ImageLayer* i utilitza una font *ImageWMS*. Per poder-la connectar amb el servei WMS del GeoServer, hi he posat per una banda la seva url, i després li he configurat els paràmetres. Així buscava la capa en qüestió dins del servei WMS i amb el format *image/png*. Per acabar amb la configuració de la font, també li he dit que era un servei de tipus *geoserver*.

Pel que fa a la capa dels restaurants, havia de codificar-la de tal manera que es descarregués les dades directament del OpenStreetMap. Per fer-ho creem una variable buida de tipus *VectorSource*, que ens servirà com a font per a la capa. Després omplim de dades aquesta variable, en aquest cas li demanem que ens retorni els restaurants *amenity=restaurant* dins de la zona que li delimitem.

Un cop tenim les dades, creem la capa, que serà de tipus *VectorLayer*. Li indicarem que la font és la variable amb les dades que acabem de definir. Per altre banda també definim el disseny de cada punt, perquè ens mostri la icona que tenim guardat a la carpeta *images*.

Aquesta capa tindrà un element *PopUp*, és a dir una finestra emergent que s'activarà quan cliquem damunt d'una icona amb el ratolí. Per fer-ho hem de definir la interacció mitjançant una variable de tipus *Select*. Amb aquesta variable li diem el que ha de fer un cliquem un element d'aquesta capa. Li definim un disseny nou del punt seleccionat. Després cal indicar-li que quan es seleccioni un element, és a dir que s'activi la variable que defineix la interacció anterior, s'obri la finestra emergent amb la informació de l'element seleccionat. Per tant cal relacionar-ho amb l'element *PopUp* definit i dissenyat al codi html. Per últim, cal indicar-ho en la definició de la variable *map*, és a dir del propi mapa.

## 4 Dificultats

Basicament m'he trobat amb dos grans dificultats a l'hora de desenvolupar el treball. Una relacionada amb la programació Python dins del sistema operatiu Windows 10, sobretot el mòdul *psycpg2* i una altre relacionat amb com havia de descarregar les dades de la API Flickr.

Pel que fa a la primera, tot i que el mòdul de Python l'haviem fet amb linux, vaig voler provar de fer-ho amb Windows. Com que ja tenia instal·lat Python 2.7 a l'ordinador ja el vaig utilitzar i no em vaig preocupar de fer una instal·lació neta des de zero. Això va fer que ja tingué problemes per executar un script senzill o per instal·lar-hi qualsevol mòdul. Problemes que mica en mica vaig poder anar solucionant. Fins que vaig arribar a la fase de pujar les dades a la base de dades. Ja que havia d'instal·lar el mòdul *psycpg2* i aquí vaig tenir molts problemes i òbviament vaig perdre molt de temps. Tot i això finalment ho vaig poder resoldre.

També vaig perdre molt de temps intentant pujar més de 1250 elements. Finalment me'n vaig adonar que el problema no era del mòdul *psycpg2*, sinó de la API de Flickr que tot i que et deixava baixar tants elements com volguessis, a partir dels 1250 el que feia era repetir-los. Per tant el que passava era que volia introduir un element amb un *id* que ja tenia un altre element. Com que columna *id* era la clau, no es podia repetir.

Amb la API Flickr també vaig perdre força temps intentant baixar les dades. Vaig provar diversos mòduls, fins que finalment vaig trobar el *flickrapi*. A més a més també em va costar trobar la manera de treballar amb les dades que estaven en format JSON per poder-les introduir a la base de dades.

## 5 Conclusions

A nivell de conclusions, em costa treure'n pel simple fet que el treball en sí no tenia com a objectiu final l'anàlisi en si d'un element o d'una situació o regió, sinó més aviat ser un compendi de totes eines que em aprés en aquest màster. Tot i això entenc tal i com està plantejat l'anunciat, cal treure'n conclusions de les zones més turístiques en funció de les dues capes sobreposades que he introduït al mapa, tot i el poc rigor científic que tindrà.

Pel que es pot veure al mapa podem afirmar que el casc antic de reus és la zona amb una major aflluència turística, tant pel que va al número de fotografies que fan, com per la concentració de restaurants. En el cas de la distribució de fotografies, que ens pot donar una imatge una mica més nítida, podem veure com dins del casc antic, la zona que queda delimitada pel tomb de raval és la més densa i dins d'aquí la Plaça del Mercadal és la zona més fotografiada. Seguit de lluny per la plaça del Prim i la Prioral de Sant Pere.

Com a altres zona a on s'hi concentra un element número de fotografies, trobem el camp de futbol de la Reddis, el Parc de Mas Iglésies, la plaça de la Llibertat i una zona pròxima a les piscines municipals.

En el cas dels restaurants, també es pot apreciar una important concentració a la zona delimitada pel tomb de raval, però a la vegada també s'observa una major difuminació en la seva distribució a la resta de la ciutat.