

ACTUALITZACIÓ

SERVEI DE RUTES BASAT EN

OSRM

OPEN SOURCE ROUTING MACHINE



MÀSTER EN TECNOLOGIES DE LA INFORMACIÓ GEOGRÀFICA

PROJECTE FINAL



Autor **ARROYO SOTO, Luis Alberto**

Tutors **SALA, Laura (LIGIT) | CLOS, Manel (UMAT)**

INTRODUCCIÓ

OBJECTIUS

REQUERIMENTS

TECNOLOGIES

IMPLEMENTACIÓ

RESULTAT

CONCLUSIONS

INTRODUCCIÓ

UMAT

La Unitat Municipal d'Anàlisi Territorial (UMAT), va ser creada l'any 1995 per l'Ajuntament de Girona. És l'encarregada de gestionar:

- Servei Municipal de Cartografia
- Gestió del Sistema d'Informació Territorial
- Estudis d'anàlisi territorial



INTRODUCCIÓ

OBJECTIUS

REQUERIMENTS

TECNOLOGIES

IMPLEMENTACIÓ

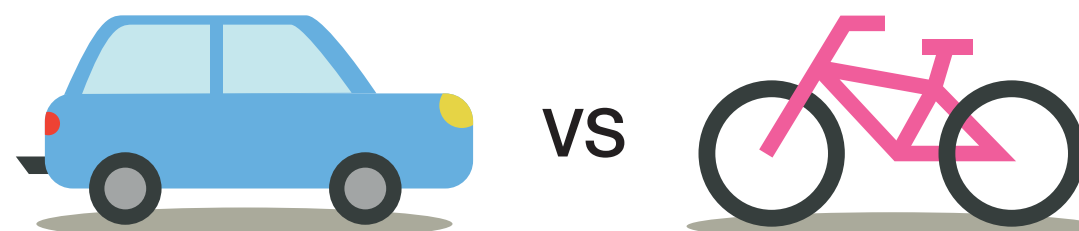
RESULTAT

CONCLUSIONS

INTRODUCCIÓ

Context i necessitats

1. Problemàtica entre ciclistes i vehicles a motor a la ciutat de Girona.
2. Inseguretat per part dels ciclistes.
3. Necessitat per millorar la fluïdesa entre ambdós medis de transport.



Antecedents

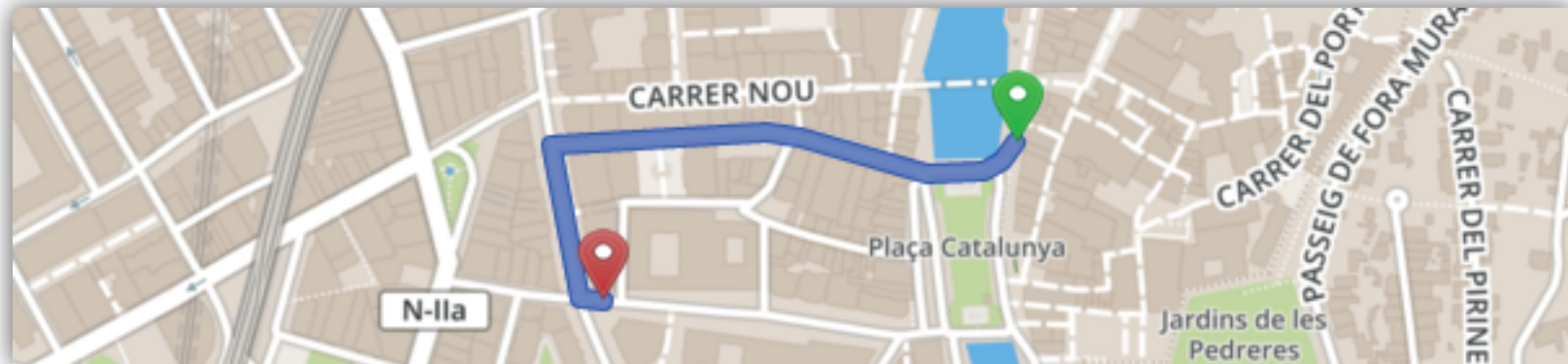
Edició 15 del MTIG es va dur a terme la primera implementació d'OSRM



Objectius

L'objectiu principal

Implementar una aplicació web de routing per a la ciutat de Girona adaptat a la via ciclista.



Objectius específics

- Tractar les dades per tal de disposar d'una xarxa complexa de carrers i carrils bici.
- Implementar un servei de routing amb un perfil específic de bicicletes.
- Generar les rutes a partir de dades pròpies per tal de tenir el sistema actualitzat.
- Realitzar el procés en una màquina virtual per treballar amb requisits mínims.
- Automatitzar el procés per a que sigui fàcilment reproducible.



Requeriments

Funcionals

- Visualitzar distància / temps
- Ruta en destacat
- Invertir sentit de la ruta
- Permet canviar origen o final
- Prioritat carril bici
- Afegir punts intermitjos
- Entrada de direccions origen / final
- Només rutes de bicicleta

No-Funcionals

- Sistema operatiu: Mac OS X El Capitan
- Processador: 2,5 Ghz Intel Core I5

Per a la màquina virtual:

- Sistema operatiu: Ubuntu 14.04+ (64bits)
- Memòria Disc dur: 2Gb i 4 processadors



Tecnologies

INTRODUCCIÓ

OBJECTIUS

REQUERIMENTS

TECNOLOGIES

IMPLEMENTACIÓ

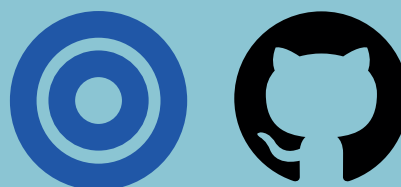
RESULTAT

CONCLUSIONS

1 Tractament de dades
JOSM, ArcGis i QGis



2 Software
OSRM i Git



3 Sistema operatiu
Ubuntu 14.04

5 Servidor Web
Apache2.4



4 Editor de codi
Atom



6 Creador d'entorn
Vagrant



7 Llenguatges
JavaScript i Python



8 Llibreries
GDAL, nodejs, npm i Leaflet



Tecnologies

OSRM

- Motor d'enrutament en C++, de codi lliure.
- Combina algoritmes complexos amb dades d'OSM.
- Flexible en la importació de dades.
- Manega rutes continentals en poc temps.
- Perfils editables (cotxe, peu, bicicleta).



OpenStreetMap (OSM)

- Projecte de mapamundi col·laboratiu.
- Inspirat en l'èxit de la Wikipèdia.
- Exemple d'informació geogràfica voluntària.
- Desenvolupament de multitud d'aplicacions donat el seu gran creixement, OSRM és un bon exemple.



Implementació

Arquitectura del Sistema

INTRODUCCIÓ

OBJECTIUS

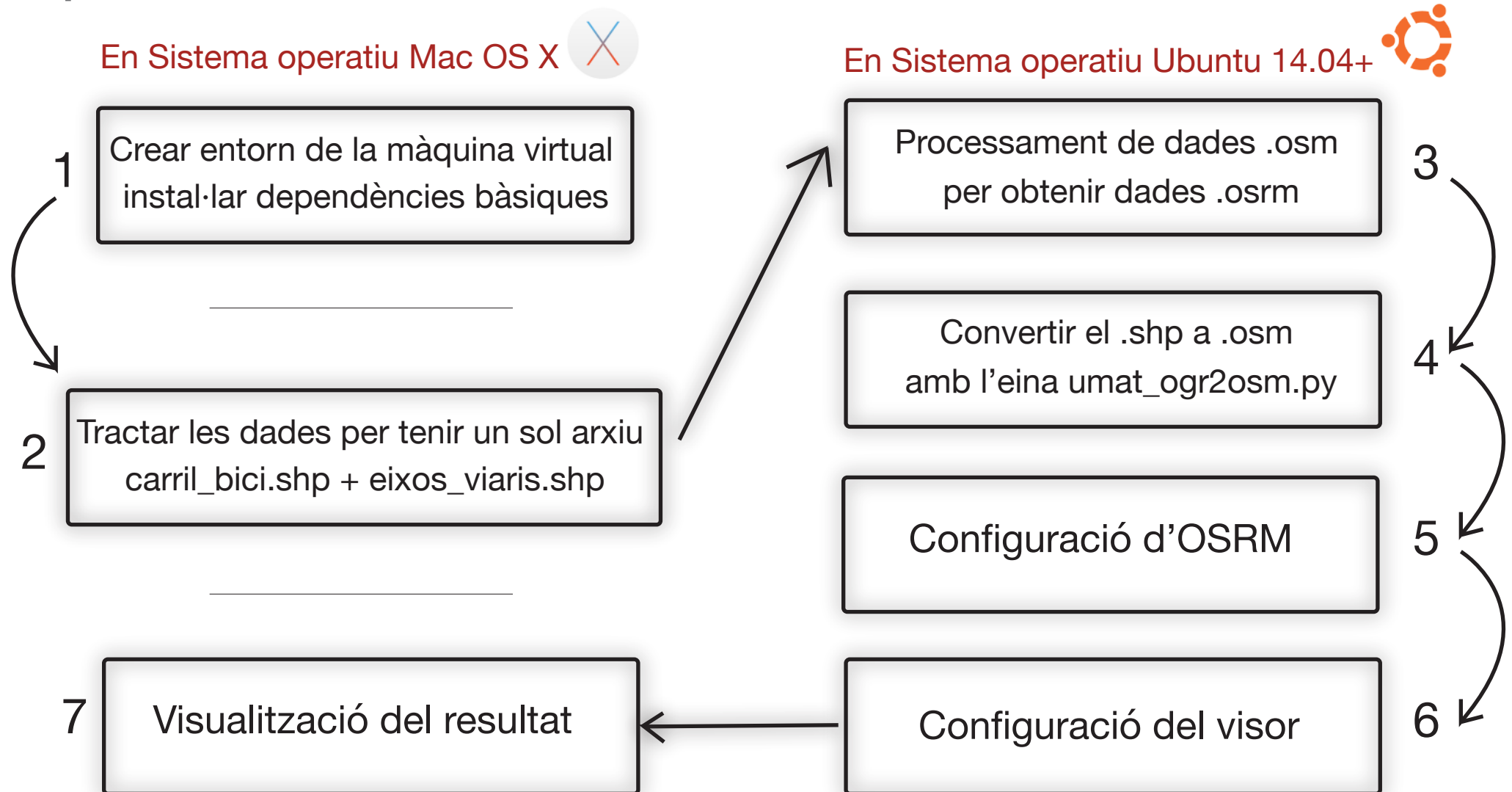
REQUERIMENTS

TECNOLOGIES

IMPLEMENTACIÓ

RESULTAT

CONCLUSIONS



Implementació

Creació de l'entorn

INTRODUCCIÓ

OBJECTIUS

REQUERIMENTS

TECNOLOGIES

IMPLEMENTACIÓ

CREACIÓ ENTORN

TRACTAR DADES

CONVERSIÓ ARXIUS

CONFIGURAR OSRM

PROCESSAR OSRM

CONFIGURAR VISOR

RESULTAT

CONCLUSIONS

1. Instal·lar VAGRANT

2. Editar el VAGRANTFILE (arxiu de configuració inicial)

```
config.vm.network "forwarded_port", guest: 80, host: 8080
config.vm.network "forwarded_port", guest: 5000, host: 5000
```

```
config.vm.provider "virtualbox" do |v|
  v.memory = 2048
  v.cpus = 4
```

3. Amb la comanda **vagrant up** inicialitzem, amb **apt-get update** s'actualitzen els paquets essencials d'Ubuntu 14.04+.-

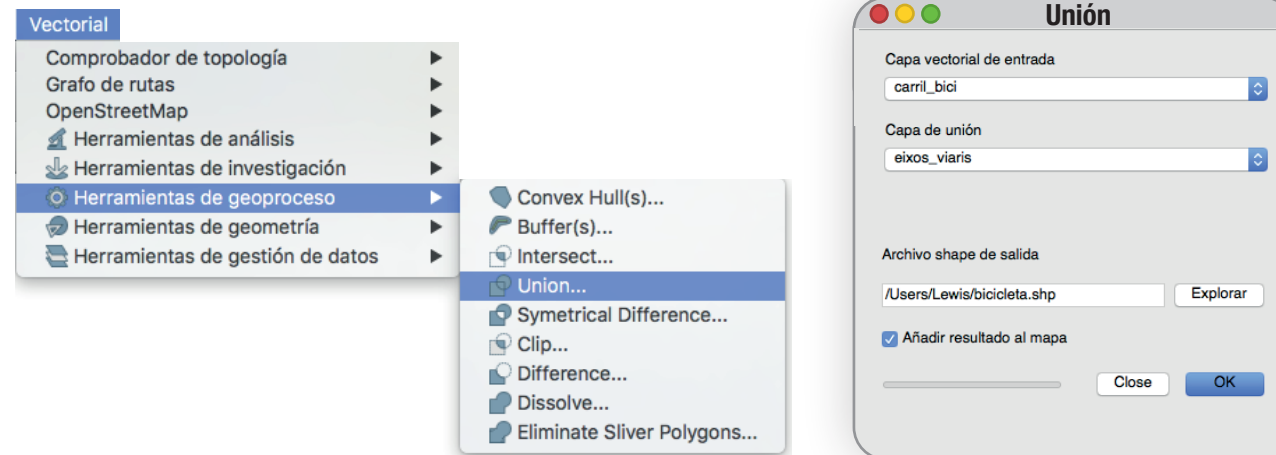
```
Last login: Mon Feb  8 12:24:09 on console
[MacBook-Pro-de-Luis:~ Lewis$ cd mtig2
[MacBook-Pro-de-Luis:mtig2 Lewis$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'ubuntu/trusty64' is up to date...
==> default: A newer version of the box 'ubuntu/trusty64' is available! You currentl
y
==> default: have version '20150909.1.0'. The latest is version '20160206.0.0'. Run
==> default: `vagrant box update` to update.
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
==> default: Forwarding ports...
default: 80 => 8080 (adapter 1)
default: 5000 => 5000 (adapter 1)
default: 22 => 2222 (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
default: SSH address: 127.0.0.1:2222
default: SSH username: vagrant
default: SSH auth method: private key
default: Warning: Connection timeout. Retrying...
==> default: Machine booted and ready!
```



Implementació

Tractament de dades

1. Unió dels dos shapefiles inicials (eixos_viaris.shp i carril_bici.shp) amb l'eina UNION de QGis.



2. Buidatge de columnes irrelevantants.
3. Instal·lar la llibreria OGR (part de GDAL) per deprés fer la conversió del shapefile a format .osm.

```
apt-get install -y python-gdal
```

- INTRODUCCIÓ
- OBJECTIUS
- REQUERIMENTS
- TECNOLOGIES
- IMPLEMENTACIÓ**
- CREACIÓ ENTORN
- TRACTAR DADES**
- CONVERSIÓ ARXIUS
- CONFIGURAR OSRM
- PROCESSAR OSRM
- CONFIGURAR VISOR
- RESULTAT
- CONCLUSIONS



Implementació

Conversió .shp en .osm

1. Es farà servir l'arxiu bicicletes.py, que és un arxiu de translació d'atributs específics iniciats per la funció TranslationMethod, dins de l'arxiu umat_ogr2osm.py.

```
if translationMethod:
    try:
        sys.path.append(os.getcwd() + "/translations")
        module = __import__(translationMethod)
        translateAttributes = module.translateAttributes
        translateAttributes([])
    except Exception, e:
        print e
        print "Could not load translation method " + translationMethod + ". Check
        sys.exit(-1)
    print "Successfully loaded " + translationMethod + " translation method."
else:
    # If no function has been defined, perform no translation: just copy everything
    translateAttributes = lambda(attrs): attrs

# Manel: generar IDs positius
# elementIdCounter = -1
elementIdCounter = 1
nodeCount = 0
```

umat_ogr2osm.py (funció translationMethod)

```
def translateAttributes(attrs):
    if not attrs: return

    tags = {}

    if attrs['OBJECTID_1']:
        tags.update({'objectid': attrs['OBJECTID_1']})
    #carril_bici
    if attrs['OBJECTID']:
        tags.update({'objectid': attrs['OBJECTID']})

    tags.update({'highway': 'unclassified'})
    tags.update({'cycleway': 'lane'})

    # tags.update({'highway': 'unclassified'})

    # Street name
    if attrs['NOM_COMPLE']:
        nom_carrer = attrs['NOM_COMPLE']
        if nom_carrer:
            nom_carrer = nom_carrer.decode('utf-8')
        tags.update({'name': nom_carrer})
        tags.update({'name:ca': nom_carrer})
```

bicicleta.py (translació dels atributs)

**destacat en vermell, correcció de la tipografia*

2. Amb la següent comanda s'arrenca el procés de conversió per obtenir bicicleta.osm

```
python umat_ogr2osm.py -t bicicleta -p "+init=epsg:25831 +wktext" bicicleta.shp
```



Implementació

Configuració d'OSRM

1. Per implementar OSRM es treballa desde el lloc oficial: <https://github.com/Project-OSRM>
 2. Es descarreguen les dependències necessàries.
 3. S'instal·la el Git i es descarrega el projecte, abans però cal crear els directoris on s'emmatzemanan els arxius.
 4. Per últim, una vegada descarregat es crea el directori build per tal de realitzar el cmake i el make.
- En acabar el procés, OSRM resta configurat en la màquina virtual.

```
cd /data/src/osrm-backend  
mkdir build  
cd build  
cmake ..  
make
```

```
root@vagrant-ubuntu-trusty-64:/data/src/osrm-backend/build# make  
[ 2%] Built target ANGLE  
[ 4%] Built target COMPRESSEDGE  
[ 9%] Built target COORDINATE  
[ 11%] Built target EXCEPTION  
[ 14%] Configuring revision fingerprint  
-- Fingerprint unchanged, not regenerating  
[ 14%] Built target FingerPrintConfigure  
[ 16%] Built target FINGERPRINT  
[ 19%] Built target GRAPHCOMPRESSOR  
[ 26%] Built target IMPORT  
[ 28%] Built target LOGGER  
[ 30%] Built target MERCATOR  
[ 33%] Built target PHANTOMNODE  
[ 35%] Built target RESTRICTION  
[ 54%] Built target OSRM  
[ 57%] Built target RASTERSOURCE  
[ 59%] Built target osrm-datastore  
[ 80%] Built target osrm-extract  
[ 90%] Built target osrm-prepare  
[100%] Built target osrm-routed  
root@vagrant-ubuntu-trusty-64:/data/src/osrm-backend/build#
```



Implementació

Processar OSRM

INTRODUCCIÓ

OBJECTIUS

REQUERIMENTS

TECNOLOGIES

IMPLEMENTACIÓ

CREACIÓ ENTORN

TRACTAR DADES

CONVERSIÓ ARXIS

CONFIGURAR OSRM

PROCESSAR OSRM

CONFIGURAR VISOR

RESULTAT

CONCLUSIONS

1. Crear enllaç simbòlic del bycycle.lua a dins del directori build i profile/lib.

```
ln -s ../profiles/bycycle.lua profile.lua
```

```
ln -s ../profiles/lib/
```

2. Engregar l'eina *extract* per tal d'eliminar tota la informació irrellevant per crear rutes.

```
./osrm-extract biciclieta.osm
```

3. Continuar amb l'eina *prepare* per crear una estructura *nearest-neighbour* i un mapa de nodes que es creen juntament amb la jerarquia.

Quan el procés ha finalitzat, tots els arxius necessaris per crear rutes són al directori *build*.

```
[root@vagrant-ubuntu-trusty-64:/data/src/osrm-backend/build# ls
bicicleta.osm                bicicleta.osrm.ramIndex      Makefile
bicicleta.osrm               bicicleta.osrm.restrictions  osrm-datastore
bicicleta.osrm.core          bicicleta.osrm.timestamp    osrm-extract
bicicleta.osrm.ebg           CMakeCache.txt             osrm-prepare
bicicleta.osrm.edges         CMakeFiles                  osrm-routed
bicicleta.osrm.fileIndex     cmake_install.cmake         profile.lua
bicicleta.osrm.geometry     compile_commands.json       stxxl.errlog
bicicleta.osrm.hmgr          lib                          stxxl.log
bicicleta.osrm.names         libOSRM.a                   util
bicicleta.osrm.nodes         libosrm.pc
```



Implementació

Configurar el visor

1. Instal·lar Apache i nodejs per poder executar el frontend.v2 (visor del projecte).
2. Es descarrega el projecte a través del git com en ocasions anteriors
3. Substituir el servei de dades per defecte, d'aquesta manera es fan servir les dades generades a partir dels passos anteriors (bicicleta.osrm)

```
services: [{  
  label: 'Car (fastest)',  
  path: 'http://api-osrm-routed-patrick-develop.tilestream.net/viaroute'  
  path: 'http://localhost:5000/viaroute'  
}],
```

4. Finalment, engegar el servidor de rutes

```
./osrm-routed bicicleta.osrm
```

INTRODUCCIÓ

OBJECTIUS

REQUERIMENTS

TECNOLOGIES

IMPLEMENTACIÓ

CREACIÓ ENTORN

TRACTAR DADES

CONVERSIÓ ARXIUS

CONFIGURAR OSRM

PROCESSAR OSRM

CONFIGURAR VISOR

RESULTAT

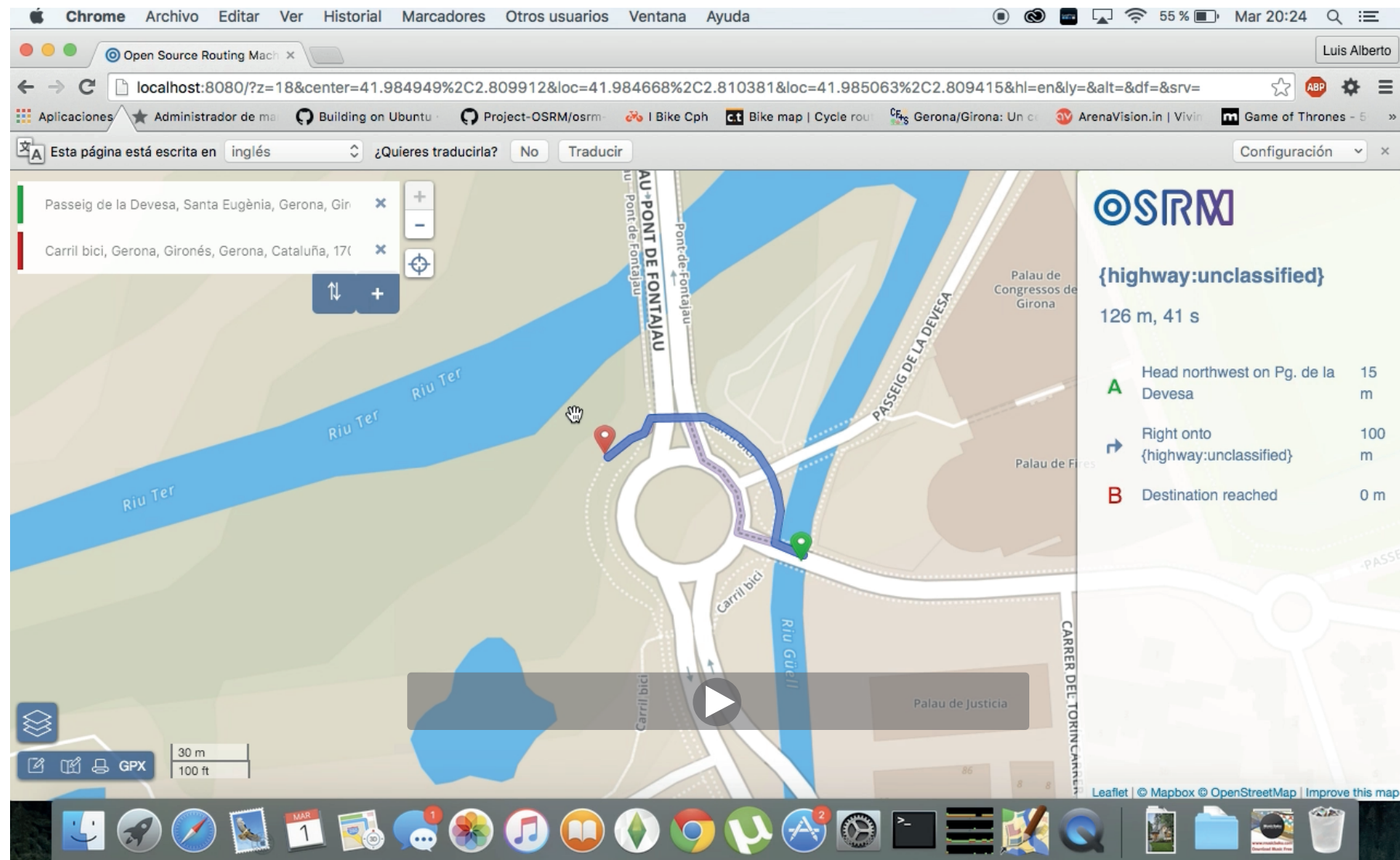
CONCLUSIONS



Resultat

bicicleta.shp → bicicleta.osm → bicicleta.osrm

INTRODUCCIÓ
OBJECTIUS
REQUERIMENTS
TECNOLOGIES
IMPLEMENTACIÓ
RESULTAT
CONCLUSIONS



Universitat Autònoma de Barcelona
18 de març de 2016

ACTUALITZACIÓ
SERVEI DE RUTES BASAT EN
OPEN SOURCE ROUTING MACHINE

17mtig 2015

UAB
Universitat Autònoma de Barcelona
Departament de Geografia

Ajuntament  de Girona

UMAT Unitat Municipal
d'Anàlisi Territorial



Conclusions

1. Creació de rutes amb perfil de bicicleta (velocitats, preferències...)
2. Interconnexió entre carrils bici i eixos viaris.
3. Ús de dades pròpies, actualitzables en qualsevol moment.
4. Possibilitat de sol·licitar i visualitzar rutes desde el visor del propi projecte OSRM.
5. Potencial inimaginable davant d' administracions que no gaudeixen d'un servei de routing precís per part de Google Maps, Bing...





Universitat Autònoma de Barcelona
18 de març de 2016

ACTUALITZACIÓ
SERVEI DE RUTES BASAT EN
OPEN SOURCE ROUTING MACHINE

17mtig 2015

UAB
Universitat Autònoma de Barcelona
Departament de Geografia

Ajuntament  de Girona

UMAT Unitat Municipal
d'Anàlisi Territorial

