

Plan para el taller de la [Media Party](http://lanyrd.com/2013/hhba/scpdzb/)

piratepad: <http://piratepad.net/bGIhPhqLAz>

Intro

=====

Aplicación que vamos a desarrollar: <http://opensas.github.io/mapa-cultura>

Repo de github: <https://github.com/opensas/mapa-cultura>

1. Obtener y procesar datos

=====

ckan: <http://ckan.org/>

Portal de datos: <http://datospublicos.gob.ar/datos>

Datasets de cultura:

<http://datospublicos.gob.ar/data/dataset/mapa-del-sistema-de-informacion-cultural>

[Dataset de lugares geográficos]

(<http://datospublicos.gob.ar/data/dataset/mapa-del-sistema-de-informacion-cultural/resource/5f24b04b-65ea-4f52-84e8-f12f196a29b2>)

OpenRefine: <http://openrefine.org/>

1. Facet por tipo - mostrar faceta

2. provincia:

facetar por provincia

trim - remove two whitespaces

cluster

3. eliminar puntos sin direcciones

facet by blank - remove matching

4. eliminar columnas que no queremos - mostrar historial

2. Geocodificar nuestra información utilizando servicios web

=====

Explicar: web service, rest, json

UI versus API - un ejemplo:

hands on: buscar Av. Corrientes 456, Ciudad Autónoma de Buenos Aires, Argentina

Open Street Map UI (para seres humanos)

UI: <http://www.openstreetmap.org>

UI: entrada: caja de texto  
salida: mapa en pantalla

--

OpenStreetMap API (para aplicaciones)

<http://open.mapquestapi.com/nominatim/v1/search.php?format=json&q=Av. Corrientes 456, Ciudad Autónoma de Buenos Aires, Argentina>

(Av. Corrientes 456)[[open.mapquestapi.com/nominatim/v1/search.php?format=json&q=Av. Corrientes 456, Ciudad Autónoma de Buenos Aires, Argentina](http://open.mapquestapi.com/nominatim/v1/search.php?format=json&q=Av. Corrientes 456, Ciudad Autónoma de Buenos Aires, Argentina)]

API: entrada: REST (url)  
salida: json

avenida Corrientes 456, ciudad de buenos aires, argentina

(Av. Corrientes 456 con google maps)[<http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address=Av. Corrientes 456, Ciudad Autónoma de Buenos Aires, Argentina>]

--

hands on: geocode

-----

mapa\_cultura-step\_1 (geocode).csv

1. filtrar los primeros 10 items - marcar manualmente  
facet by star

2. crear columna direccion\_geo

value + ', ' +  
cells.provincia.value + ', Argentina'

3. crear columna osm\_geo

'<http://open.mapquestapi.com/nominatim/v1/search.php?format=json&q=>' +  
value.escape('url')

4. eliminamos las que no trajeron resultado - columna status

value.parseJson().length()  
filter  
remove rows  
remove column status

5. add lat, lon column

```
columna lon: value.parseJson()[0].lon.toNumber()
```

```
columna lat: value.parseJson()[0].lat.toNumber()
```

remove column direccion\_geo, osm\_geo

exportar como cultura-tmp

mostrar el archivo

3. Exponer nuestra información como un web service usando CartoDB

=====

intro CartoDB

ir al CartoDB de devel

<https://devel.cartodb.com/dashboard/>

2. mostrar cultura, el dataset real

agregar campos

hacer una consulta

```
select * from cultura where subtipo like '%teatr%'
```

```
select * from cultura where nombre like '%Konex%'
```

wizards - category - tipo

visualize - publish

share

--

Mostrar la api que crea CartoDB

[http://devel.cartodb.com/api/v2/sql?q=select \\* from cultura limit 10](http://devel.cartodb.com/api/v2/sql?q=select * from cultura limit 10)

[http://devel.cartodb.com/api/v2/sql?q=select \\* from cultura where nombre = 'Ciudad Cultural Konex'](http://devel.cartodb.com/api/v2/sql?q=select * from cultura where nombre = 'Ciudad Cultural Konex')

4. Github - Publicar y compartir nuestro proyecto en github

explicar los repos:

cada repo es autonomo, y contiene todo el historial  
git nos permite sincronizar repos

tenemos el repositorio original (opensas)  
nuestro repositorio forkeado en github (sscarano)  
nuestro repositorio clonado localmente

me logueo como opensas

<https://github.com/sscarano>

busco mapa-cultura

<https://github.com/opensas/mapa-cultura>

lo forkeamos

en mi maquina local, clonamos el repo (el mio!)

git clone <https://github.com/sscarano/mapa-cultura.git>

cd mapa-cultura

ejecutamos la aplicación

http-server

<http://localhost:8080>

## 5. La aplicación Javascript

=====

mostrar la funcionalidad

filtrar por tipos, por calle rivadavia, corriente, etc..

explicar qué es un servidor web  
servidor web de archivos -  
o dinámico

explicar qué se ejecuta en el cliente, y qué en el servidor

mostrar como por cada operacion realizamos consultas contra el web service

<http://devel.cartodb.com/api/v2/sql?q=>

select

tipo, subtipo, nombre, direccion, telefono, email, web, lat, lon

from

cultura

where

(lower(nombre) like '%konex%' or lower(direccion) like '%konex%') and  
(lower(tipo) in ('espacios culturales'))

copiar y pegar y mostrar el json

## 6. Poniendo en producción con github

=====

ir a settings: <https://github.com/sscarano/mapa-cultura/settings>

git checkout -b gh-pages

(mostramos la nueva rama)

git branch -a

(hacemos algun cambio)

git add .

git commit -m 'deploy'

git push origin

git push --set-upstream origin gh-pages

vamos de vuelta a settings

ir a settings: <https://github.com/sscarano/mapa-cultura/settings>

---

## 7. mejorando nuestra aplicación y aportando cambios

=====

crear un nuevo ticket:

<https://github.com/sscarano/mapa-cultura/issues>

git checkout master

git status

arreglamos el issue

git status

git add index.html

git status

git commit -m 'fixes #1, achicamos la caja de búsqueda'

git push

mostramos el commit

creamos el pull-request  
<https://github.com/sscarano/mapa-cultura/pulls>

--

8. phonegap

<https://build.phonegap.com/>

<https://build.phonegap.com/apps/529950/share>

--

si queda tiempo, volvemos a la agenda, haciendo un repaso de lo visto