

Visualització interactiva xarxes de col·laboració de recerca

Xarxes GREC

Descripció, guia d'instal·lació i utilització

Desenvolupat per Albert Díaz-Guilera, Oscar Fajardo Fontiveros, Arnau Quinquilà Capdevila

Complexity Lab Barcelona (<http://clabb.eu>), Departament de Física de la Matèria Condensada i Universitat de Barcelona Institute of Complex Systems (UBICS), Universitat de Barcelona

Copyright © 2017 Universitat de Barcelona

1. Introducció

Aquesta guia pretén guiar a un usuari amb coneixements prèvis en desenvolupament/manteniment web i maneig en l'executat d'scripts en la instal·lació de l'aplicació “Visualització interactiva xarxes de col·laboració de recerca” (en endavant, Xarxes GREC).

Abans de començar, però, descriurem de forma breu els dos components tot indicant-ne el seu àmbit d'us.

1.1 Xarxes GREC

Xarxes GREC és un conjunt d'aplicacions en backend i frontend que permet comprendre la col·laboració científica de la institució en la qual és implementada. Amb aquest component podrem generar xarxes de col·laboració científica i publicar-les en el format que considerem més adient. Cal tenir clar que tots els components presentats en aquesta guia estan desenvolupats per ser utilitzats en el marc del sistema de gestió de la recerca **GREC**

A Xarxes GREC podrem representar la coautoría en facultats i departaments, en les quals xarxes els nodes poden ser (depenen de la modalitat) investigadors, departaments, facultats o grups de recerca. Xarxes GREC consta d'una part *backend* i una *frontend*. En la part *backend* podrem generar el conjunt d'arxius necessaris per representar, en diversos nivells, les xarxes de col·laboració científica de la institució en que vulguem instal·lar el component. Un cops generats aquests arxius, serà a través de la part *frontend* que podrem visualitzar-les, en les plataformes següents,

- **Aplicació web pública:** Portal web sense autenticació que ens permetrà seleccionar a través d'un menú la xarxa que volem visualitzar.
- **Aplicació amb autenticació:** Mateix portal web que en el cas de la plataforma pública, però en aquest cas dotat d'un sistema d'autenticació PHP que ens permetrà administrar un conjunt d'usuaris.
- **Component incrustat:** Component en miniatura que ens permetrà veure una previsualització d'una xarxa donada. Aquest està dotat d'un botó que ens permetrà accedir al visor de la mateixa xarxa a pantalla completa i de forma interactiva.

2. Xarxes GREC: Dependències

Per l'ús dels programes de creació de xarxes (part *backend*) es requereix tenir, en el vostre maquinari, instal·lat,

- Python 3: En el cas que estiguem usant *Windows* o *Mac OS X* haurem de descarregar-lo i instal·lar-lo manualment. Podem descarregar Python des de la seva pàgina web oficial (<https://www.python.org/>). En cas que estiguem utilitzant una de les últimes distribucions de Linux, python 3 ja ve instal·lat de forma predeterminada.

- *NetworkX*: Paquet de Python que ens permetrà generar i treballar amb les xarxes de col·laboració. La instal·lació d'una dependència de Python es pot fer de moltes formes. La forma més senzilla és a partir del terminal del sistema, tot executant la comanda següent,

```
$ pip install networkx
```

En el cas que en el nostre ordinador ja tinguéssim instal·lat Python2, es produirà un conflicte a l'hora d'afegir la dependència al Python. Així doncs, hem de ser curosos a l'hora de executar la comanda *pip*, ja que podria estar instal·lant la dependència al Python2. Per estar segur que aquesta anirà inserida en l'entorn de Python3, haurem d'executar la comanda,

```
$ pip3 install networkx
```

que en el cas de no tenir ambdós versions de Python instal·lats ens donarà un error. En cas que no siguem capaços d'instal·lar *NetworkX* a través de *pip*, podem fer-ho manualment a partir de la font original. Ambdós metodologies estan explicades en la documentació oficial del paquet

<https://networkx.github.io/documentation/development/install.html>

Per la part de frontend només caldrà tenir un servidor web instal·lat per veure les aplicacions web que contindran les visualitzacions. A la secció 4 s'explica com generar-ne un localment usant Python 3.

3. Xarxes GREC: *Backend*

Començarem construint tots aquells arxius que requerim per tal de visualitzar les xarxes de col·laboració de la nostra institució. Sigui quina sigui la plataforma que vulguem usar per visualitzar-les, haurem de seguir tots els punts descrits a continuació. Anirem descrivint per fases els passos a seguir. D'ara en endavant treballarem en el directori *backend* que haureu trobat en el conjunt de fitxers que conformen Xarxes GREC.

3.1 Primera fase: Creació i format dels arxius d'entrada

Per arxius d'entrada entenem tots aquells arxius que continguin la informació sobre els investigadors, departaments, grups i facultats de la nostra institució. Aquests documents han de ser generats a partir del propi sistema de gestió de la recerca GREC. Els arxius generats han de tenir un format i estructura concrets. Si els arxius d'entrada no els formatem i estructurarem tal com s'especifica en aquesta documentació, els scripts de la part *backend* no seran capaços de generar les xarxes de col·laboració. Un cop haguem generat els arxius fent us de GREC, els haurem de situar a la carpeta *input_csv* dins del directori *backend*.

Els arxius a generar, així com el seu format específic, es detallen a continuació,

- **articles_investigadors.csv**: Arxiu que conté tots els articles indexats en la base de dades. L'arxiu consta de dues columnes: La primera, on apareix el codi de l'investigador autor de l'article, i en la segona, el codi identificador de l'article. Cada fila ha de contenir un únic investigador, i per tant, un article pot estar representat per diverses files. Les columnes seran doncs:

codi de l'investigador; codi de l'article

amb l'ordre especificat aquí i emprant punt i coma com a separador.

- **investigadors.csv**: Arxiu que assigna un investigador a un departament i facultat. Les columnes seran doncs:

codi de l'investigador;codi del departament;codi de la facultat

amb l'ordre especificat aquí i emprant punt i coma com a separador.

- **noms_departaments_centres.csv**: Arxiu que assigna els noms dels departaments i facultats amb els seus respectius codis identificactius. Les columnes han de ser:

codi element; nom element

amb l'ordre especificat aquí i emprant punt i coma com a separador.

- **investigador_grup_detall.csv**: Arxiu que assigna a cada investigador un grup, així com assigna un nom, primer i segon cognom a l'investigador. Les columnes han de ser:

codi del grup;codi de l'investigador;primer cognom de l'investigador;segon cognom de l'investigador;nom de l'investigador

amb l'ordre especificat aquí i emprant punt i coma com a separador.

- **grups.csv**: Arxiu que assigna el noms dels grups amb els seus respectius codis identificactius. Les columnes han de ser:

codi del grup;codi oficial del grup;nom del grup

amb l'ordre especificat aquí i emprant punt i coma com a separador.

En aquesta llista el nom del arxius apareix en negreta. El nom dels arxius ha de ser exactament com l'especificat. A més, tots ells han de ser en format **CSV** i en codificació **UTF-8**. El separador de les columnes ha de ser el **punt i coma** (;). Finalment,

Tots els arxius d'entrada hauran de ser introduïts **sense** capçalera

és a dir, no podran contenir els nom de les columnes que representen les dades. Això és així perquè el sistema identifica les columnes no pel seu nom, sinó per la posició que ocupen aquestes. Per aquesta raó és **molt important respectar la posició de les columnes tal com s'ha especificat**.

Un cop haguem formatat adequadament les columnes podrem procedir a copiar i enganxar els documents generats a la esmentada carpeta *input_csv*. Arribats en aquest punt, ja podem executar el script que generarà les xarxes de col·laboració.

3.2 Segona fase: Generar les xarxes

A hores d'ara ja està tot llest per generar les xarxes. Haurem doncs d'executar un script desenvolupat en Python 3 que farà aquesta tasca de forma automàtica. El script és autònom, però requereix la instal·lació prèvia de dependències. Depenen del sistema operatiu que estiguem usant aquest procés serà un o altre. En primer lloc, requerim tenir el mateix Python instal·lat a la nostra màquina (veure secció anterior).

Si hem seguit el passos especificats en la secció precedent, podrem passar a executar el script sense més problemes. Obrim un terminal i ens desplaceu fins a la carpeta *backend*. Un cop allí, executem la següent comanda,

```
$ python3 xarxes-grec.py
```

S'iniciarà l'execució del programa. Un resultat com el següent és el que us hauria d'aparèixer,



```
*****
*                                     *
*                               XARXES GREC                               *
*                                     *
*****

Script backend: Creació i gestió del model de dades de l'aplicació web

-----
(1) PASSOS PREVIS
-----

(1.1) ENCRIPCIÓ

Existeix la opció de mostrar els nodes (grups, departaments...) amb els seus IDs.
- Voleu que el programa usi els identificadors enlloc dels noms? [s/N]
```

Figure 1: Captura de la pantalla inicial de l'script que generarà els fitxers de les xarxes.

El programa us preguntarà en primer lloc si voleu que els investigadors, departaments, grups i facultats de la vostra institució surtin encriptats a les xarxes de col·laboració. És a dir, que enlloc d'aparèixer amb el seu nom, s'ensenyin amb el seu codi identificador, ocultant-ne així la seva identitat. En cas que vulgueu que els nodes surtin identificats amb els codis premeu **s** i posteriorment la tecla *enter*. En cas contrari, premeu **n** + *enter* o *enter* directament (en ser la opció predeterminada).

A continuació el script generarà una carpeta al directori amb el nom *data*, on hi generarà tots els arxius necessaris per després poder crear les xarxes de col·laboració.

El script no requereix de més interacció. Passats uns minuts (el temps dependrà de la mida de la vostra institució) el programa finalitzarà. Si no hi ha hagut cap problema podreu observar com en la carpeta *backend* ha aparegut un carpeta amb el nom *data*.

Aquesta carpeta serà l'element de informació central per poder generar les xarxes en l'aplicatiu web. Per tal de representar-les seguim els passos indicats en el capítol següent, dedicat a la part *frontend*.

4. Xarxes GREC: *Frontend*

En aquest capítol prendrem els arxius generats gràcies al programari *backend* específic i els usarem per provar l'aplicatiu web que ens mostrarà les xarxes de col·laboració.

Com ja hem explicat en la introducció, podem visualitzar les nostres xarxes de col·laboració de tres formes diferents, que presentem a continuació en forma de seccions.

4.1 Portal Públic

Per portal públic entenem una aplicació web que ens permet seleccionar les facultats o departaments que volem visualitzar a través d'una xarxa de col·laboració i que no ve protegit el seu accés amb un sistema d'acreditació d'usuari/contrasenya (és a dir, és de lliure accés).

El pas previ que ens cal per preparar el portal és copiar la carpeta *data* continguda en la carpeta *backend* (recordem que l'hem generada seguint el procediment descrit en l'apartat *Backend*) i l'enganxarem al directori *frontend/portal_public*. En aquesta secció el nostre directori mare serà justament *portal_public* contingut dins de *frontend*.

No calen més preparatius. El portal està llest per visualitzar les xarxes creades anteriorment.

Per tal de poder provar en local el portal hem de crear un servidor virtual. Ja que disposem de Python, l'usarem per crear-ne un localment. Així doncs, obrim un terminal en el nostre ordinador, ens desplaçem fins a la carpeta *frontend/portal_public* i executem la següent comanda,

```
$ python3 -m http.server
```

Això generarà un servidor local HTTP al port 8000 en la nostra màquina. El directori mare d'aquest servidor estarà situat en la carpeta on haguem executat la comanda anterior, que en aquest cas ha de ser en *frontend/portal_public*.

Per accedir ara al portal hem d'obrir el nostre navegador web habitual i dirigir-nos a l'adreça web `http://localhost:8000`, on podrem navegar i visualitzar les xarxes de col·laboració que estiguem més interessats.

No ens allargarem en la descripció del portal ja que aquest mateix està dotat d'una ajuda que podeu accedir, un cop creada una xarxa, des del botó *Informació general*.

4.2 Portal protegit amb acreditació (privat)

Per portal protegit amb acreditació entenem la mateixa aplicació web definida en la secció anterior, és a dir, el portal públic. En aquest però, l'aplicació ve dotada d'un sistema d'autenticació per mitjà d'usuari/contrasenya. Concretament, cada cop que un usuari vulgui accedir a l'aplicatiu s'haurà d'autenticar mitjançant les credencials oportunes.

Informem al lector que al llarg d'aquesta secció la nostra carpeta de treball serà *portal_privat*.

4.2.1 Gestió d'usuaris i contrasenyes

De forma predeterminada l'aplicatiu ve amb un únic usuari, amb les següents propietats:

- **Usuari:** xarxes
- **Contrasenya:** grec2016

Es poden afegir més usuaris o modificar l'existent accedint a l'arxiu *_login_users.php* dins la carpeta *app*. En aquest arxiu apareix una llista de PHP amb una expressió com la següent,

```
\$ users = array (
    'xarxes' => 'grec2016' ,
)
```

Podem afegir o modificar l'usuari existent sempre i quan seguim la pauta següent,

```
\$ users = array (
    'usuari_1' => 'contrasenya_1',
    'usuari_2' => 'contrasenya_2',
    'usuari_3' => 'contrasenya_3',
    .....
    'usuari_n' => 'contrasenya_n',
)
```

On substituïrem *usuari_i* i *contrasenya_i* per l'usuari i contrasenya desitjat per cada un d'aquests.

4.2.2 Provar en local el portal amb acreditació

Si desitgem provar en el nostre ordinador el portal privat haurem d'instal·lar localment un programa que emuli un servidor HTTP/PHP. Depenen del sistema operatiu que estiguem utilitzant, aquest procés serà un o altre.

En aquest cas instal·larem un servidor web local PHP a la nostra màquina. Aquest ve incorporat en el paquet d'instal·lació de PHP, que trobareu disponible, junt amb tots els detalls d'instal·lació, a <http://php.net/>.

Un cop instal·lat PHP a la nostra màquina ens cal transferir les dades generades en la primera part d'aquest manual al nostre directori. Així doncs haurem de copiar la carpeta *data*

continguda en la carpeta *backend* (recordem que l'hem generada seguint el procediment descrit en l'apartat *Backend*) i l'enganxarem al directori *frontend/portal_privat*.

Finalment executem PHP per tal de iniciar el servidor local. Per fer-ho, accedim al directori *portal_privat* des d'un terminal i executem la següent línia,

```
$ php -S localhost:8000
```

tal està descrit a la documentació oficial de PHP. Podeu trobar més detalls a <http://php.net/manual/en/features.commandline.webserver.php>.

No calen més preparatius. El portal està llest per visualitzar les xarxes creades anteriorment.

4.3 Component incrustat

Per component incrustat entenem un fragment de plana web reutilitzable i aïllat que permet ser integrat en qualsevol entorn web HTTP/Javascript.

Xarxa de Grups



Veure-ho a pantalla completa ↗

Xarxa de grups formada a partir de la coautoria de publicacions en revistes científiques dels investigadors de la facultat

Figure 2: Detall del component incrustat. Es mostra una xarxa de col·laboració entre grups de recerca.

Aquest component permet visualitzar una xarxa en miniatura, tot permetent-ne l'exploració dels nodes a partir d'un *tooltip*.

Tal com es pot observar, el botó *Veure-ho a pantalla completa*, permet explorar la xarxa en un visor a una nova pestanya.

Si volem provar localment el component, podem fer-ho emprant de nou el servidor HTTP local de Python 3. Així doncs, obrim un terminal en el nostre ordinador, ens desplaçem fins a la carpeta *frontend/portal_public* i executem la següent comanda,

```
$ python3 -m http.server
```

Incrustar el component en una plana web

Per tal d'incrustar el component requerirem afegir tres blocs de codi HTML a la plana web en el qual el vulguem afegir.

Abans de començar, però, hem de copiar la carpeta *app* a l'arrel de la nostra plana web. El document JSON que conté la informació de la xarxa a representar ha de ser situat a la carpeta *app/data*. El nom del document(s) que hi situem els haurem de tenir presents per vincular-los després amb l'aplicació. Suposem en aquest cas usem un arxiu anomenat *exemple.json*.

Passem ara a incrustar pas per pas el component a la nostra pàgina web:

1. En primer lloc hem d'afegir a l'etiqueta *head* del nostre document HTML les següents dependències,

```
<link href="app/styles/bootstrap.min.css" rel="stylesheet">
<link href="app/styles/mapstyle.css" rel="stylesheet">
<script type="text/javascript" src="app/js/jquery.js"></script>
<script type="text/javascript" src="app/js/d3.min.js"></script>
<script type="text/javascript" src="app/js/colorbrewer.js"></script>
<script type="text/javascript" src="app/js/appMap.simplified.js"></script>
<link href="app/styles/jumbotron-narrow.css" rel="stylesheet">
```

2. En segon lloc afegim el bloc HTML on s'incrustarà la xarxa. Així doncs,

```
<div id="embeded-xarxes"></div>
```

En aquest *div*, degudament identificat, s'hi injectarà la xarxa a partir de les dades que l'hi vinculem. És molt important tenir en ment que l'amplada de la xarxa vindrà donada per la mateixa del seu *div* immediatament superior (*parent div*).

3. Un cop afegida la miniatura de la xarxa podem afegir també el botó que permet accedir al visor i un peu amb una petita descripció. Així doncs, just després de la línia que hem afegit, insertem el bloc següent,

```

<div>
  <a href="graphgenerator.html" id="botoxgraf" target="_blank">
    <span class="glyphicon glyphicon-resize-full" id="full-icon"
      aria-hidden="true">
      </span>
      <span id="full-text">
        Veure-ho a pantalla completa
      </span>
    </a>
  </div>
<div id="explicacio" class="well well-large">
  Detalls sobre la xarxa
</div>

```

- Finalment, abans de tancar l'etiqueta *body* del document HTML afegim el següent bloc de codi, que defineix un script Javascript a partir del qual indicarem a l'aplicació on està i quin nom té el document JSON que conté informació sobre la xarxa a representar.

```

<script type='text/javascript'>

$(document).ready(function( ){
  d3.json('./app/data/exemple.json', function(error,graph) {
    localStorage.setItem('graphdata', JSON.stringify(graph));
    DrawGraph(graph, 'embeded-xarxes ');
  });
});
</script>

```

La funció *DrawGraph* és la que s'encarrega d'iniciar la representació de la xarxa. El segon argument d'aquesta funció és el que indica l'identificador del div on ha de ser injectada la xarxa. Cal remarcar que aquest últim *script* de Javascript pot ser aïllat i ficat en un arxiu javascript si es prefereix. En aquest l'hem situat al propi cos de l'arxiu HTML per simplicitat.