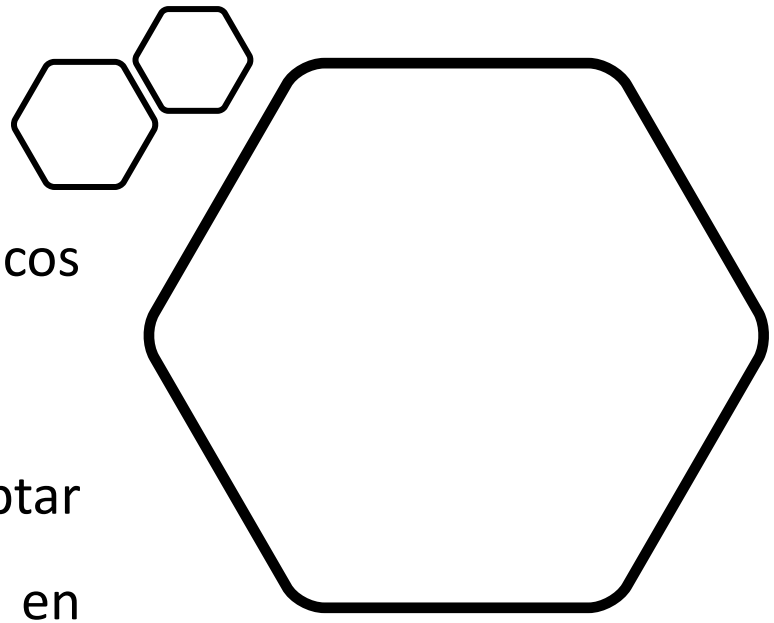
López Tamayo Diego

Nazará Sosa Emilio

A Twitter Social Network Analysis of Colmex economists

Proyecto final. Economía de Redes

Introducción



- Twitter es una plataforma a través de la cual los académicos pueden comunicarse con audiencias más amplias.
- A menudo, se ha acusado a los economistas de adoptar actitudes de superioridad y ser distantes con el público en general (Fourcade et al., 2015).
- Esta idea se ha atribuido a la baja difusión de la ciencia económica con el público y la falta de diversidad en la profesión (Crawford et al., 2018).

Motivación

Tres investigadoras de la Universidad de Reading encuentran diferencias en la forma en que los investigadores se comunican con audiencias no académicas.

(Della Giusta et al. , 2018)

Motivación

- Recopilan como datos miles de tweets de las cuentas de los 25 mejores economistas y 25 científicos identificados por IDEAS y Science.

- Descubren que los **economistas tuitean menos, mencionan a menos personas y tienen menos interacciones con extraños.**

- Además utilizan un **lenguaje menos accesible y un tono más distante** que un grupo comparable de científicos.

(Della Giusta et al. , 2018)

¿Podemos identificar un patrón similar en la comunidad tuitera del Colegio de México?

¿Cómo se relacionan en Twitter los economistas del CEE comparados con sus equivalentes de otros Centro de Estudios*?

* CEAA, CEDUA, CEE, CEH, CEI, CELL, CES, ADM

(7 centros de estudios + administrativos/biblioteca)

Objetivos, preguntas y método

Objetivo primario:

Identificar las **comunidades principales** de la red tuitera del Colmex.

¿Existe una mayor interacción entre personas del mismo centro de estudios?
¿Cuál es la estructura de la red?

Construir red dirigida y no dirigida de "**followers and following**" de la comunidad tuitera Colmex.

Objetivo secundario:

Identificar **agentes centrales** de la red y observar su **relación** con agentes externos a la red.

¿Cómo se relacionan los agentes principales con agentes externos a la red?
¿Existen diferencias entre centros de estudio?

Generar estadística descriptiva sobre los principales seguidores y seguidos por centro e identificar patrones en **follows** hacia fuera de la comunidad Colmex.

Fuente de datos y procesamiento.

1. Identificar agentes a través de búsqueda individual y apoyo de la comunidad.
   Base de datos con campos:

   Twitter ID, Username, Centro de estudios, Actualmente en Colmex

2. Twitter API para obtener información de *following* de los agentes y formar la red dirigida Colmex en Python.

3. Generar estadística descriptiva y visualización de la red con herramientas de *graph-tool* y *networkx.*

4. Identificar nodos de mayor centralidad y observar su *relación* con agentes externos (no Colmex). Analizar diferencias por Centro de Estudios.

Revisión del código implementado

```
# Hacemos un for loop maravilloso lleno de if's y while's internos y vamos llenado la base df
# Vamos a iterar para todas las observaciones de colmex
for num in range(colmex.shape[0]):
    id =  colmex['id'].iloc[num] # Guardamos el id en una variable
    url = f'https://api.twitter.com/2/users/{id}/following?max_results=1000'  # Generamos el URL del request de 1000 following
    req = requests.get(url,headers=headers) # Hacemos el request
    time.sleep(65) # Esperamos 70 segundos porque podemos hacer 15 request en una ventana de 15 minutos
    if 'data' in req.json().keys(): # Revisamos que sea un request exitoso (cuando son cuentas privadas marca error)
        following = req.json() # Pasamos el json object a un diccionario de Python
        data = dict()
        data['data'] = []
        data['data'].extend(following['data'])  # Pasamos el diccionario a una lista con los datos
        df_loop = pd.json_normalize(data, record_path =['data']) # Convertimos a un pandas dataframe con método normalize
        df_loop = df_loop[['id', 'name', 'username']] # Nos quedamos con las columnas que nos interesan (por si hay extras)
        df_loop = df_loop.rename(columns={"id": "target", # Cambiamos los nombres del target
        "name":"t_name","username":"t_username"})
        df_loop["id"]=colmex['id'].iloc[num]
        df_loop["username"]=colmex['username'].iloc[num]
        df_loop["status"]=colmex['status'].iloc[num]
        df_loop["center"]=colmex['center'].iloc[num]
        df = df.append(df_loop, ignore_index = True)  # Hacemos append del dataframe que itera (df_loop) al dataframe principal (df)
        while 'next_token' in following['meta']: # Si tiene más de 1,000 followings, revisamos el next_token en los metadatos
            pag_num = following['meta']['next_token'] # Guardamos en next_token en la variable page_num
            url=f'https://api.twitter.com/2/users/{id}/following?max_results=1000&pagination_token={pag_num}'
            req = requests.get(url,headers=headers) # Hacemos nuevamente el request de la siguiente página
            time.sleep(65) # Volvemos a esperar y se repite el proceso, el while se rompe cuando ya no haya un next_token
            following = req.json()
            data = dict()
            data['data'] = []
            data['data'].extend(following['data'])
            df_loop = pd.json_normalize(data, record_path =['data'])
            df_loop = df_loop.rename(columns={"id": "target",
            "name":"t_name","username":"t_username"})
            df_loop["id"]=colmex['id'].iloc[num]
            df_loop["username"]=colmex['username'].iloc[num]
            df_loop["status"]=colmex['status'].iloc[num]
```

ername	status	center	target	t_username	t_name
MGF91	former	CEE	4745020953	PlanasRodriguez	NuriaRodriguezPlanas
MGF91	former	CEE	222575693	HosteriaLaBota	Hostería La Bota.
MGF91	former	CEE	172526279	Adri_35	Adriana Garcia
MGF91	former	CEE	902275011486068736	AnaBerthaGtz	Ana B Gutiérrez
MGF91	former	CEE	335593579	Pauagudelo	Pau Agudelo
...	...	...	...	...	...
968 Alextuto	former	CEI	167138820	khristiebloom	Cristina G 🕉
863968 Alextuto	former	CEI	141417962	Memo_e	Guillermo Esquivel
176863968 Alextuto	former	CEI	145815514	ElMikelAntonio	Ai dise gratis. C-137
176863968 Alextuto	former	CEI	3362741	sopitas	Sopitas
63 176863968 Alextuto	former	CEI	126672013	diegoluna_	diego luna

76364 rows × 7 columns

Limpieza de Google Forms

Objetivo: Importar encuesta y limpiar:

Archivo resultado: 1_colmex_limpio.csv. Última descarga 09/04/2020 00:00 hrs 328 registros en total. 310 registros netos.

```
# Importamos la base original
original = pd.read_csv(pwd + "/0_google_forms.csv")
# Duplicamos la base para no modificar la original
colmex = original.copy()
# Cambiamos el nombre de las columnas
colmex = colmex.rename(columns={colmex.columns[0]: 'time',
                                colmex.columns[1]: 'status',
                                colmex.columns[2]: 'center',
                                colmex.columns[3]: 'username',
                                colmex.columns[4]: 'id'})
# Seleccionamos las columnas relevantes
colmex = colmex[["id", "username","status","center"]]
# Creamos dos diccionarios para mapear los nombres dic_status y dic_center
dic_status = {'En la comunidad actualmente': 'current',
    'Ex-alumno, ex-becario, ex-profesor, ex-administrativo': 'former'}
dic_center = {
    "Centro de Estudios Económicos (CEE)": "CEE",
    "Centro de Estudios Históricos (CEH)": "CEH",
    "Centro de Estudios Internacionales (CEI)":"CEI",
    "Centro de Estudios Económicos (CEE)" : "CEE",
    "Centro de Estudios Lingüísticos y Literarios (CELL)" : "CELL",
    "Centro de Estudios de Asia y África (CEAA)" : "CEAA",
    "Centro de Estudios Demográficos, Urbanos y Ambientales (CEDUA)" : "CEDUA",
    "Centro de Estudios Sociológicos (CES)" : "CES",
    "Biblioteca/Administrativo/Idiomas" : "ADM"}
# Remplazamos los valores de acuerdo a los diccionarios
colmex=colmex.replace({"status": dic_status})
colmex=colmex.replace({"center": dic_center})
# Quitamos IDs duplicados
colmex = colmex.drop_duplicates(subset=['id'])
# Quitamos usernames mal escritos con @ al inicio
colmex['username'] = colmex['username'].str.replace('@','')
colmex.reset_index(drop=True)
# Guardamos la base limpia
colmex.to_csv (pwd + "/1_colmex_limpio.csv", index = False)
colmex.info()
```

1. Limpieza de Google Forms

Objetivo: Importar encuesta y limpiar

Archivo resultado: 1_colmex_limpio.csv. Última descarga 09/04/2020 00:00 hrs 328 registros en total. 310 registros netos.

[2]

```
# Importamos la base original
original = pd.read_csv(pwd + "/0_google_forms.csv")

# Duplicamos la base para no modificar la original
colmex = original.copy()

# Cambiamos el nombre de las columnas
colmex = colmex.rename(columns={colmex.columns[0]: 'time',
                        colmex.columns[1]: 'status',
                        colmex.columns[2]: 'center',
                        colmex.columns[3]: 'username',
                        colmex.columns[4]: 'id'})
# Seleccionamos las columnas relevantes
colmex = colmex[["id", "username","status","center"]]
# Creamos dos diccionarios para mapear los nombres dic_status y dic_center
dic_status = {'En la comunidad actualmente': 'current',
    'Ex-alumno, ex-becario, ex-profesor, ex-administrativo': 'former'}
dic_center = {
    "Centro de Estudios Económicos (CEE)": "CEE",
     "Centro de Estudios Históricos (CEH)": "CEH",
     "Centro de Estudios Internacionales (CEI)":"CEI",
     "Centro de Estudios Económicos (CEE)" : "CEE",
     "Centro de Estudios Lingüísticos y Literarios (CELL)" : "CELL",
     "Centro de Estudios de Asia y África (CEAA)" : "CEAA",
     "Centro de Estudios Demográficos, Urbanos y Ambientales (CEDUA)" : "CEDUA",
     "Centro de Estudios Sociológicos (CES)" : "CES",
     "Biblioteca/Administrativo/Idiomas" : "ADM"}
# Remplazamos los valores de acuerdo a los diccionarios
colmex=colmex.replace({"status": dic_status})
colmex=colmex.replace({"center": dic_center})
# Quitamos IDs duplicados
colmex = colmex.drop_duplicates(subset=['id'])
# Quitamos usernames mal escritos con @ al inicio
colmex['username'] = colmex['username'].str.replace('@','')
colmex.reset_index(drop=True)
# Guardamos la base limpia
colmex.to_csv (pwd + "/1_colmex_limpio.csv", index = False)
colmex
```

	id	username	status	center
0	94496757	MGF91	former	CEE
1	52667287	magoreyes	current	CEE

2. Descarga de datos desde API Twitter

Objetivo: Descargar datos de following de la comunidad Colmex

Archivo resultado: 2_red_colmex_completa.csv con 276364 rows × 7 columns

[-]

```
# Creamos un data frame vacío con las columnas finales
df = pd.DataFrame(columns=['id','username','status','center','target','t_username','t_name'])

# Hacemos un for loop maravilloso lleno de if's y while's internos y vamos llenado la base df
# Vamos a iterar para todas las observaciones de colmex
for num in range(colmex.shape[0]):
    id =  colmex['id'].iloc[num] # Guardamos el id en una variable
    url = f'https://api.twitter.com/2/users/{id}/following?max_results=1000'  # Generamos el URL del request de 1000 following
    req = requests.get(url,headers=headers) # Hacemos el request
    time.sleep(65) # Esperamos 70 segundos porque podemos hacer 15 request en una ventana de 15 minutos
    if 'data' in req.json().keys(): # Revisamos que sea un request exitoso (cuando son cuentas privadas marca error)
        following = req.json() # Pasamos el json object a un diccionario de Python
        data = dict()
        data['data'] = []
        data['data'].extend(following['data'])  # Pasamos el diccionario a una lista con los datos
        df_loop = pd.json_normalize(data, record_path =['data']) # Convertimos a un pandas dataframe con método normalize
        df_loop = df_loop[['id', 'name', 'username']] # Nos quedamos con las columnas que nos interesan (por si hay extras)
        df_loop = df_loop.rename(columns={"id": "target", # Cambiamos los nombres del target
        "name":"t_name","username":"t_username"})
        df_loop["id"]=colmex['id'].iloc[num]
        df_loop["username"]=colmex['username'].iloc[num]
        df_loop["status"]=colmex['status'].iloc[num]
        df_loop["center"]=colmex['center'].iloc[num]
        df = df.append(df_loop, ignore_index = True)  # Hacemos append del dataframe que itera (df_loop) al dataframe principal (df)
        while 'next_token' in following['meta']: # Si tiene más de 1,000 followings, revisamos el next_token en los metadatos
            pag_num = following['meta']['next_token'] # Guardamos en next_token en la variable page_num
            url=f'https://api.twitter.com/2/users/{id}/following?max_results=1000&pagination_token={pag_num}'
            req = requests.get(url,headers=headers) # Hacemos nuevamente el request de la siguiente página
            time.sleep(65) # Volvemos a esperar y se repite el proceso, el while se rompe cuando ya no hay un next_token
            following = req.json()
            data = dict()
            data['data'] = []
            data['data'].extend(following['data'])
            df_loop = pd.json_normalize(data, record_path =['data'])
            df_loop = df_loop.rename(columns={"id": "target",
            "name":"t_name","username":"t_username"})
            df_loop["id"]=colmex['id'].iloc[num]
            df_loop["username"]=colmex['username'].iloc[num]
            df_loop["status"]=colmex['status'].iloc[num]
            df_loop["center"]=colmex['center'].iloc[num]
            df = df.append(df_loop, ignore_index = True)
        else:
            continue
    else:
        continue

# Guardamos la base como csv
df.to_csv (pwd + "/2_red_colmex_completa.csv", index = False)
```

3. Procesar base completa de followings y obtener red intera Colmex

Objetivo: Tener la red interna Colmex para estadística descriptiva y análisis de red.

11184 rows × 7 columns

[-]
```
red_colmex = red_colmex.rename(columns={'id':'source'})
# Revisamos si el username target está en los username de Colmex para quedarnos con la red interna.
red_interna = red_colmex.loc[red_colmex['t_username'].isin(colmex.username)]
red_interna
```

A partir del dataframe `red_interna` se obtiene la estadística descriptiva, gráficas y redes presentadas en el estudio

Red dirigida

[-]
```
edges_interna_dir = red_interna[['source', 'target']].reset_index(drop=True)  # Creamos red dirigida (dir)
# Tenemos la red, en donde nos interesa conocer la dirección de los nodos para saber quienes son las personas a las que más siguen, los nodos objetivos.
edges_interna_dir.shape
```

Red no dirigida

[-]
```
edges_interna_nd = red_interna[['source', 'target']].reset_index(drop=True)

# Que nos ordene las lineas en orden para que queden duplicadas, por ejemplo 0,1 y 1,0
sorted_cols = edges_interna_nd[['source', 'target']].apply(sorted, axis=1)
# Reemplazamos las columnas por las columnas ordenadas con el método de string
edges_interna_nd['source'] = sorted_cols.str[0]; # Ahora source son todos los menores
edges_interna_nd['target'] = sorted_cols.str[1]; # target son todos los mayores

# Nos quedamos con los duplicados (Quienes tienen Followback)
edges_interna_nd = edges_interna_nd[edges_interna_nd.duplicated()] # keep=False

# Tamaño después de dropear duplicados
edges_interna_nd.shape
```

Resultados

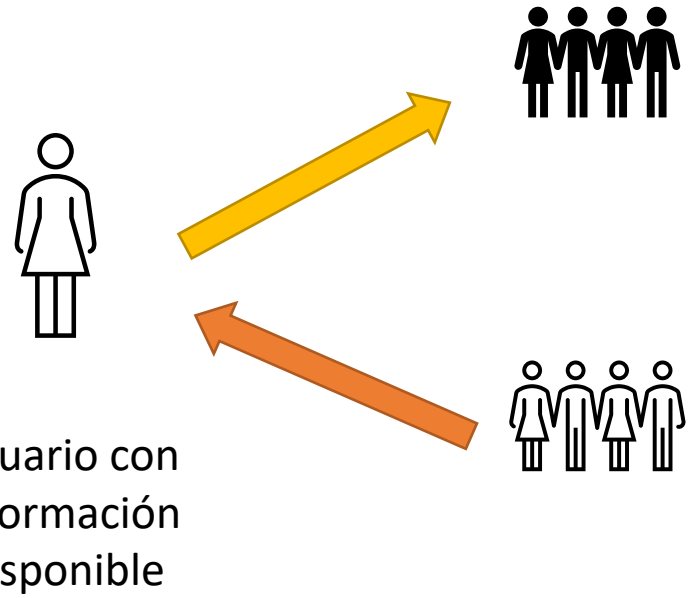
310 usuarios Colmex registrados:

Después de remover cuentas privadas y/o con restricciones:

299 con información disponible

Se obtienen los following de todas las cuentas con información disponible:

276,364 registros



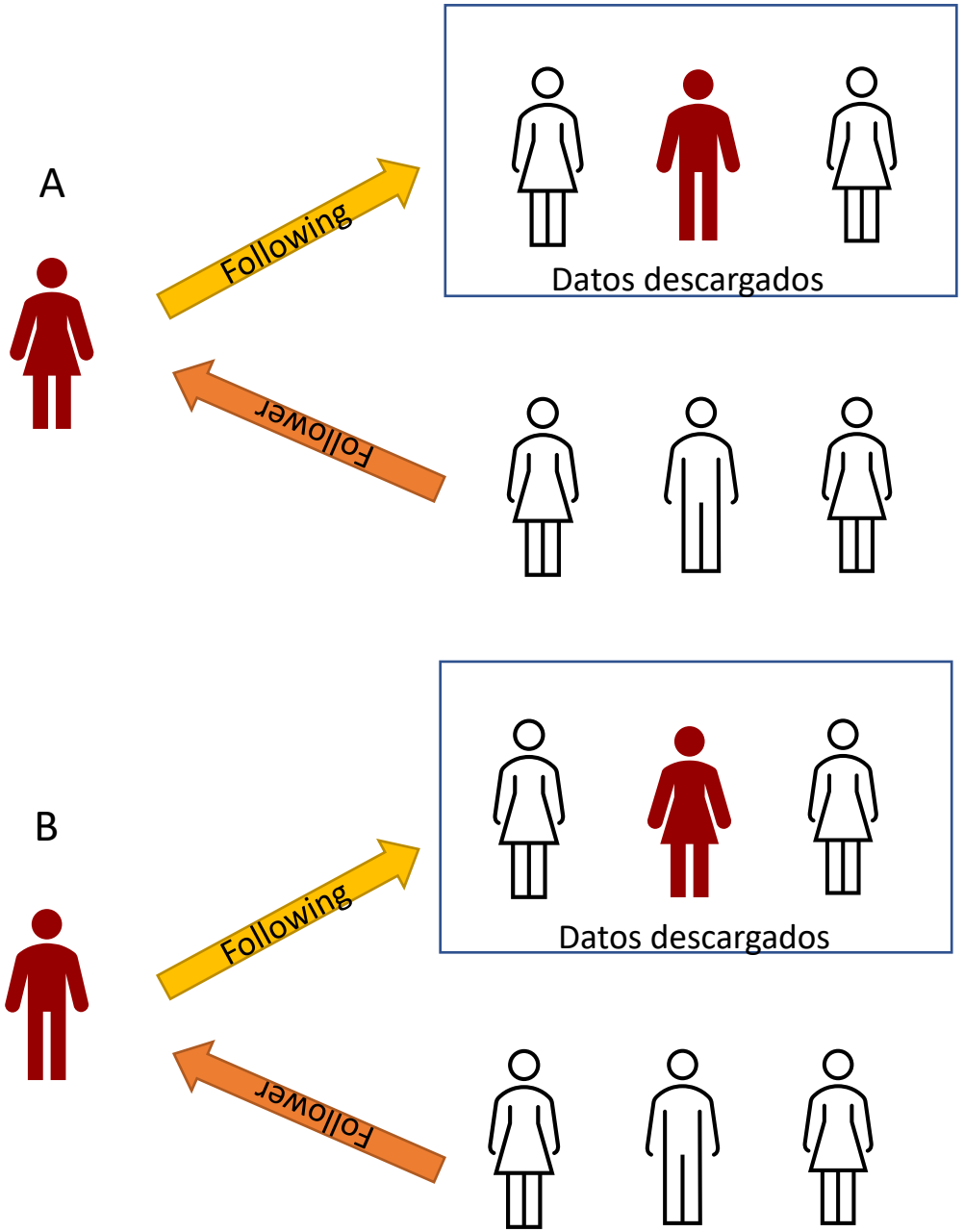
Usuario con información disponible

Following **(Información descargada)**

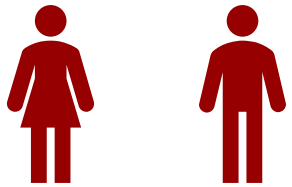
Cuentas a las que sigue.

Followers

Cuentas que lo siguen



Identificando la red Colmex



Usuarios (nodos) A y B
son comunidad Colmex.

Utilizando los datos de **following**, identificamos los edges de la red. Conociendo la relación entre A y B:

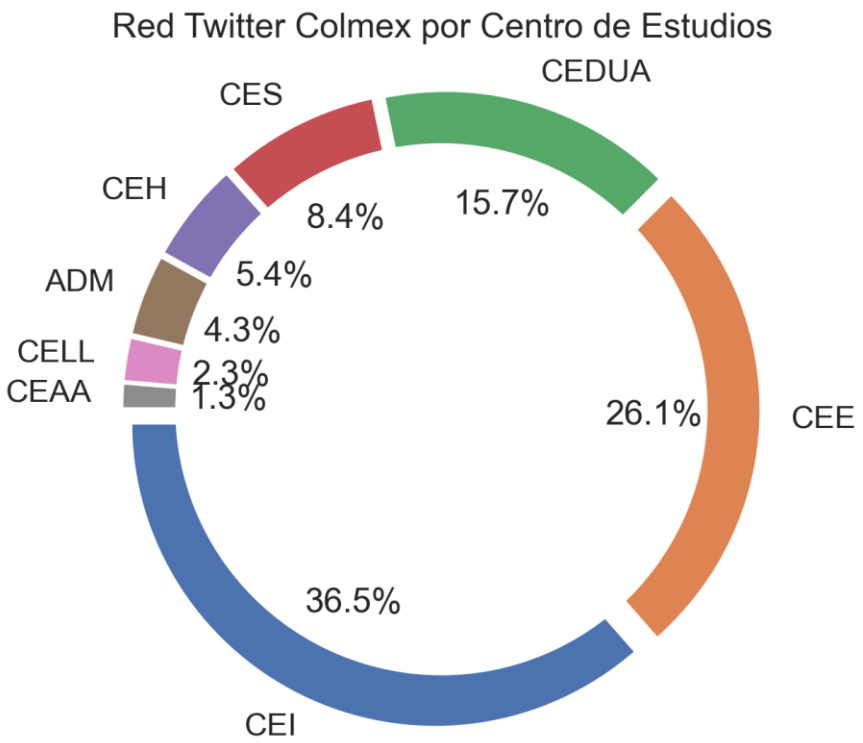
Red diridiga: 2 edges (AB , BA)

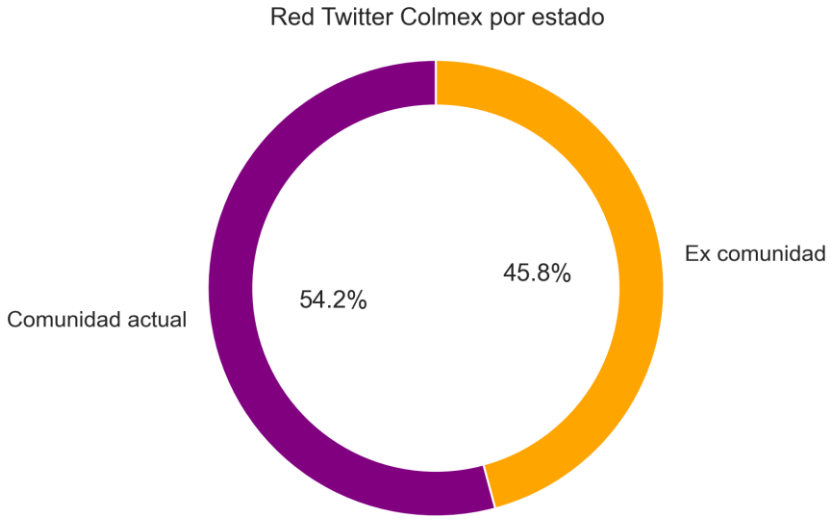
Red no dirigida: 1 edge (AB)

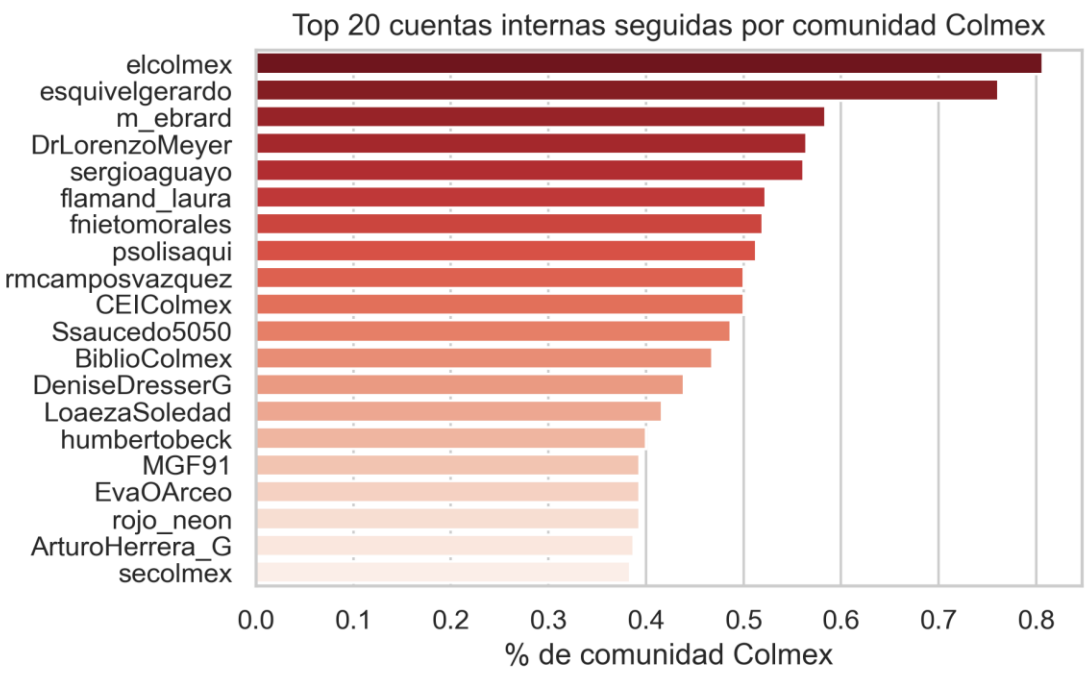
Notar que si el seguimiento no es mutuo (**followbacks**), en la red no dirigida no se forma el edge.

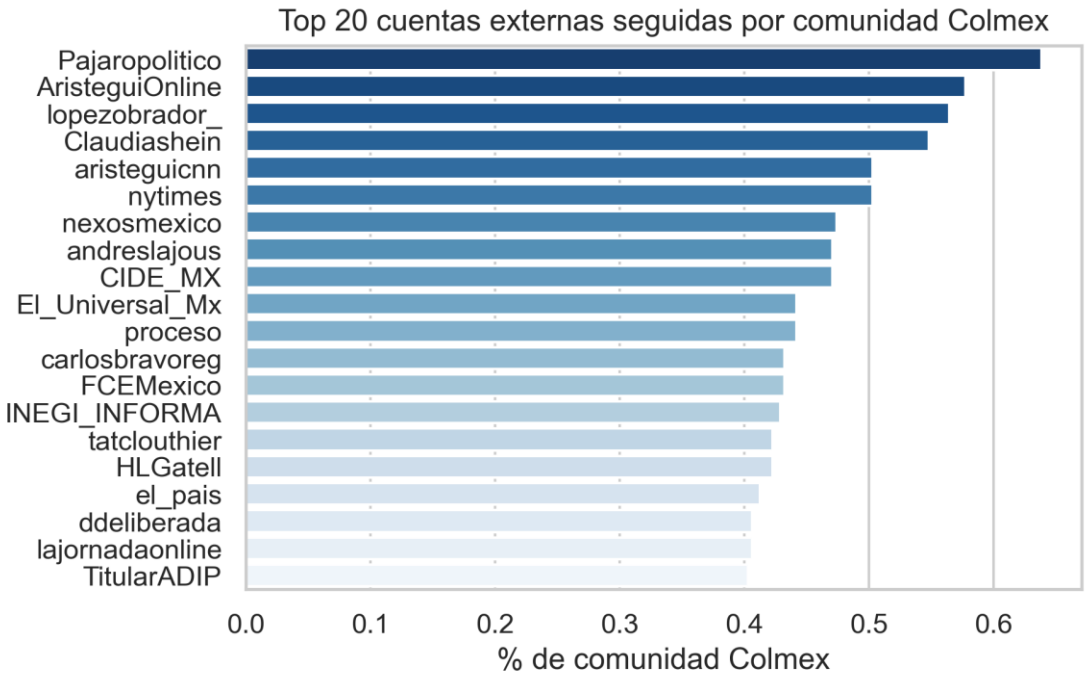
Identificando la red Colmex

299
usuarios en
red Colmex





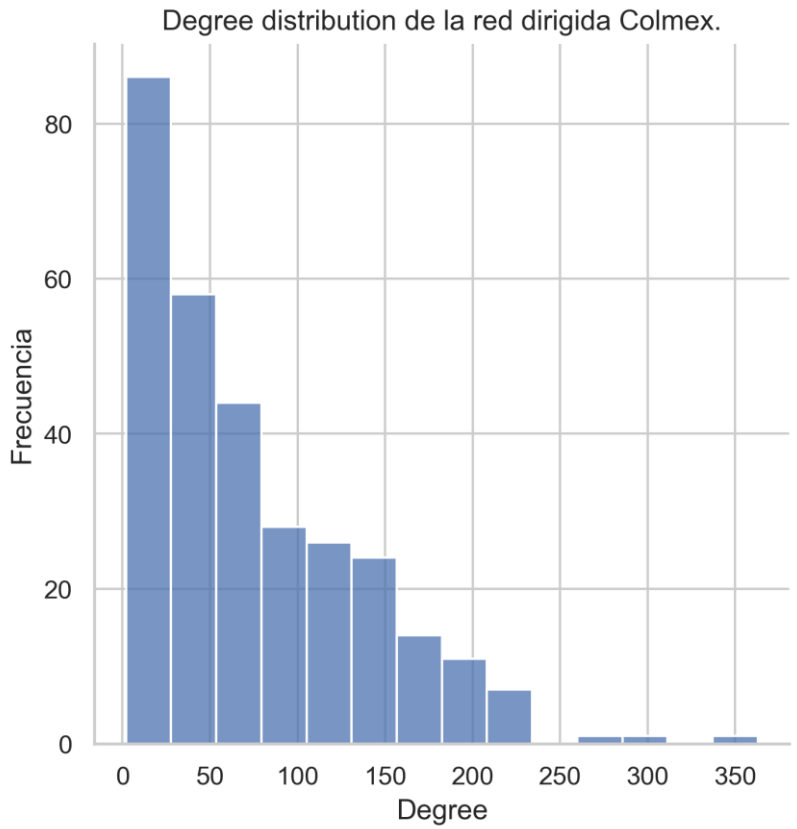

Top 20 cuentas internas seguidas por comunidad Colmex
elcolmex
esquivelgerardo
m_ebrard
DrLorenzoMeyer
sergioaguayo
flamand_laura
fnietomorales
psolisaqui
rmcamposvazquez
CEIColmex
Ssaucedo5050
BiblioColmex
DeniseDresserG
LoaezaSoledad
humbertobeck
MGF91
EvaOArceo
rojo_neon
ArturoHerrera_G
secolmex
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
% de comunidad Colmex


Top 20 cuentas externas seguidas por comunidad Colmex
Pajaropolitico
AristeguiOnline
lopezobrador_
Claudiashein
aristeguicnn
nytimes
nexosmexico
andreslajous
CIDE_MX
El_Universal_Mx
proceso
carlosbravoreg
FCEMexico
INEGI_INFORMA
tatclouthier
HLGatell
el_pais
ddeliberada
lajornadaonline
TitularADIP
0.0
0.1
0.2
0.3
0.4
0.5
0.6
% de comunidad Colmex

Notar que red no dirigida descarta 53 nodos que no tienen ningún followback en la comunidad Colmex

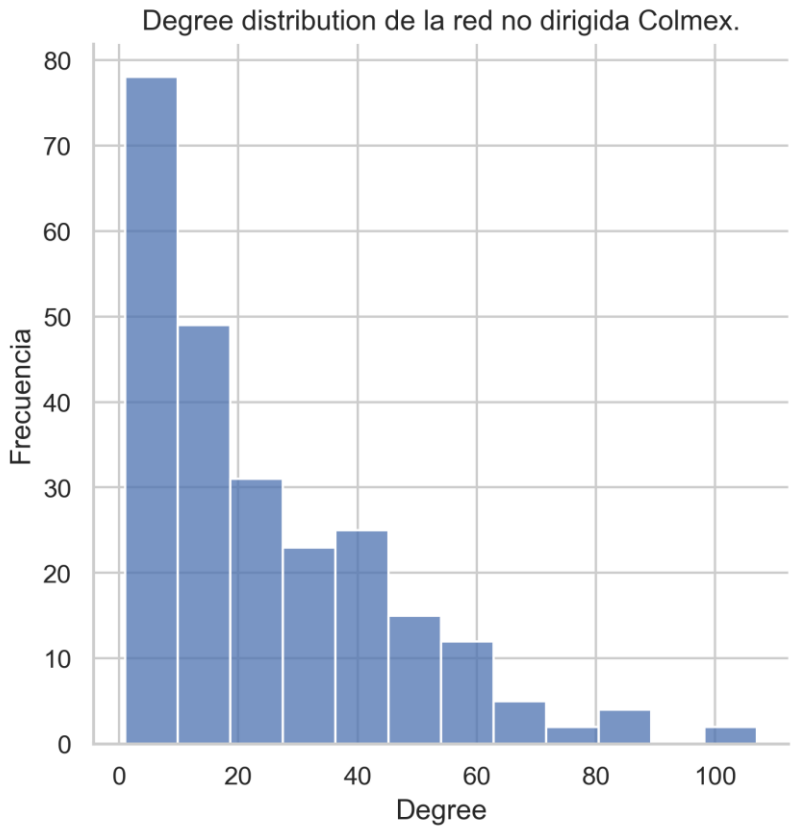
Red interna dirigida

- 299 nodos/usuarios
- 11,184 edges/follows dirigidos



Red interna no dirigida

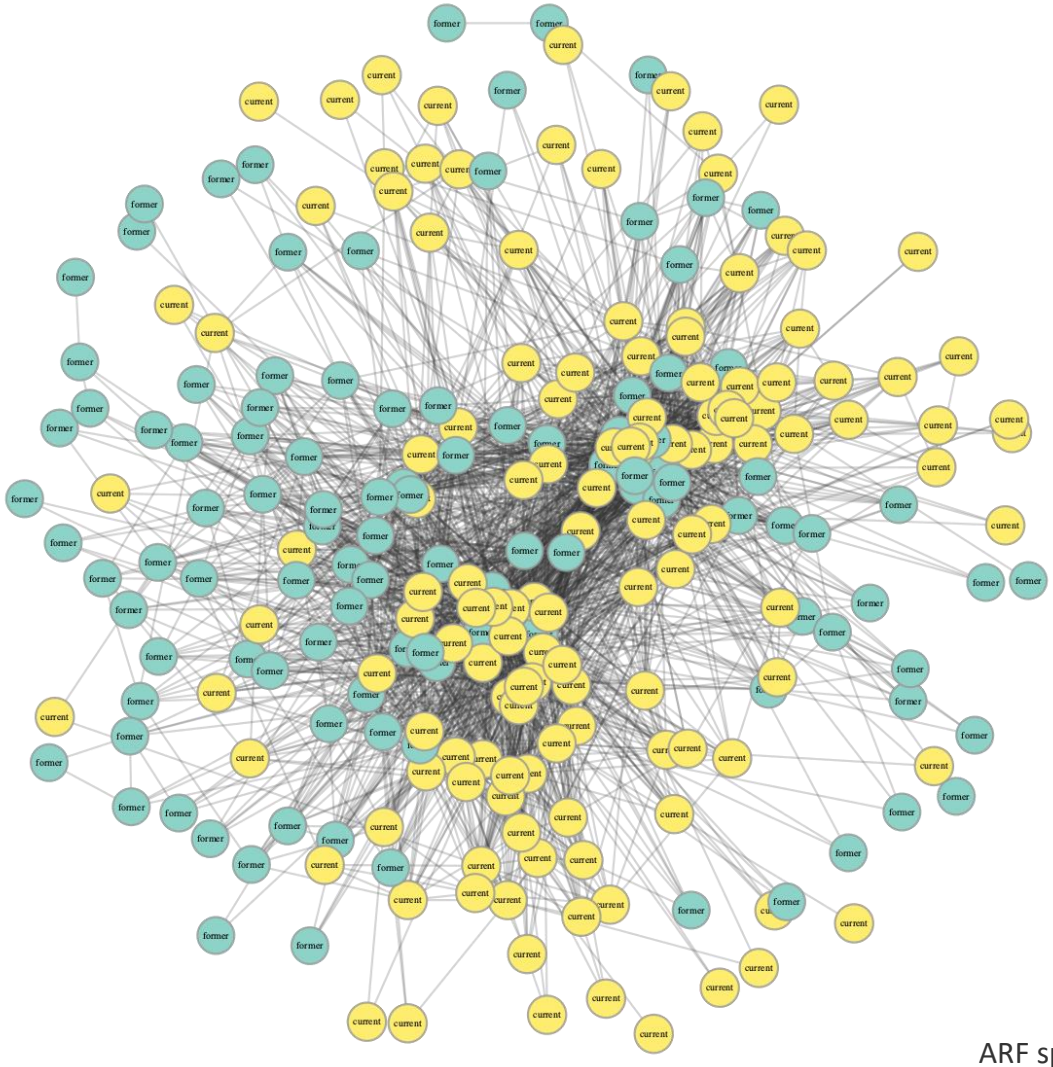
- 246 nodos/usuarios
- 2,951 edges/followbacks



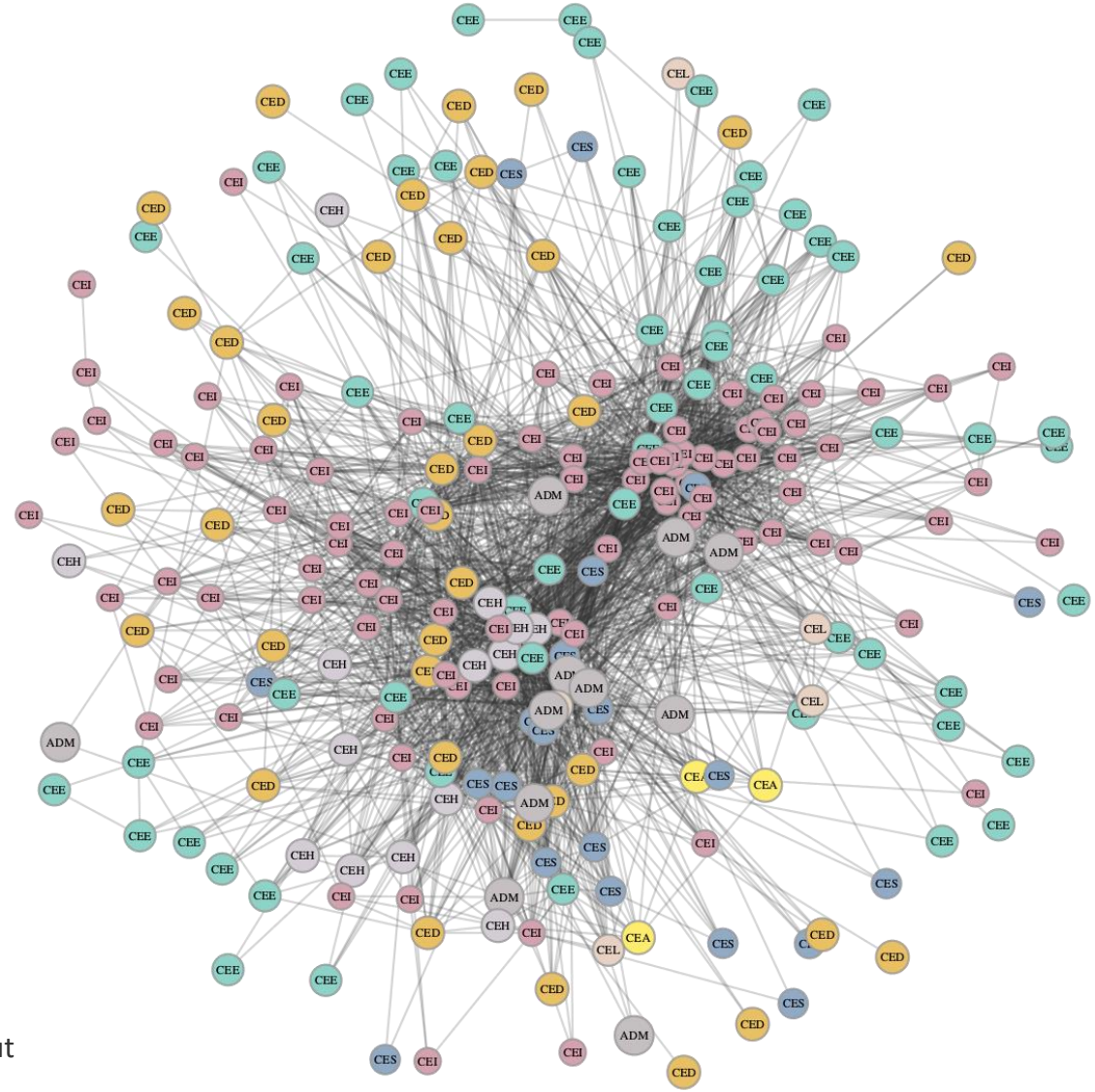
Para facilitar la visualización de la red, presentamos la **red no dirigida** (3 mil edges aprox vs 11 mil de la dirigida)

Red interna por status:

Current = Actualmente en Colmex **Former = Ex comunidad**



Red interna por centro de estudios:
CEAA, CEDUA, CEE, CEH, CEI, CELL, CES, ADM
(7 centros de estudios + administrativos/biblioteca)



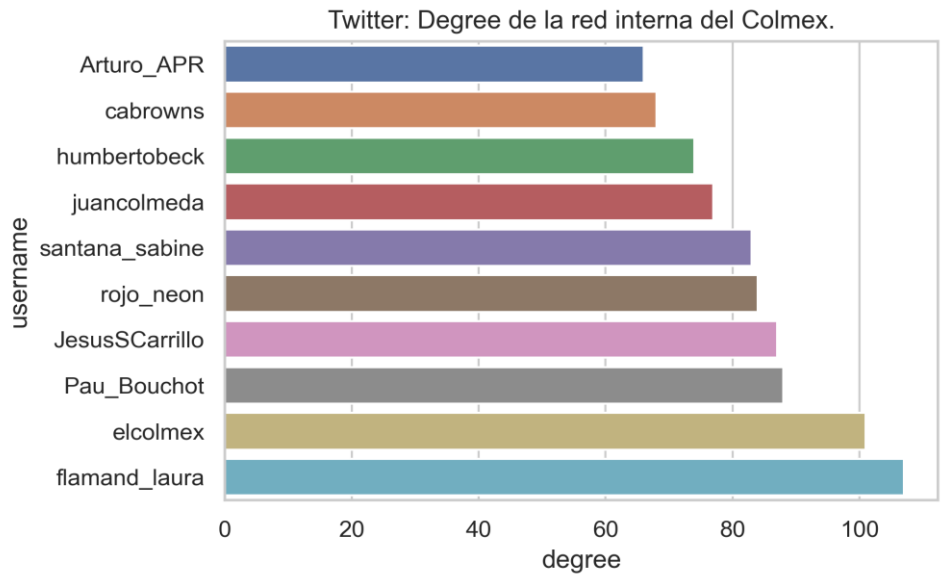
ARF spring-block layout

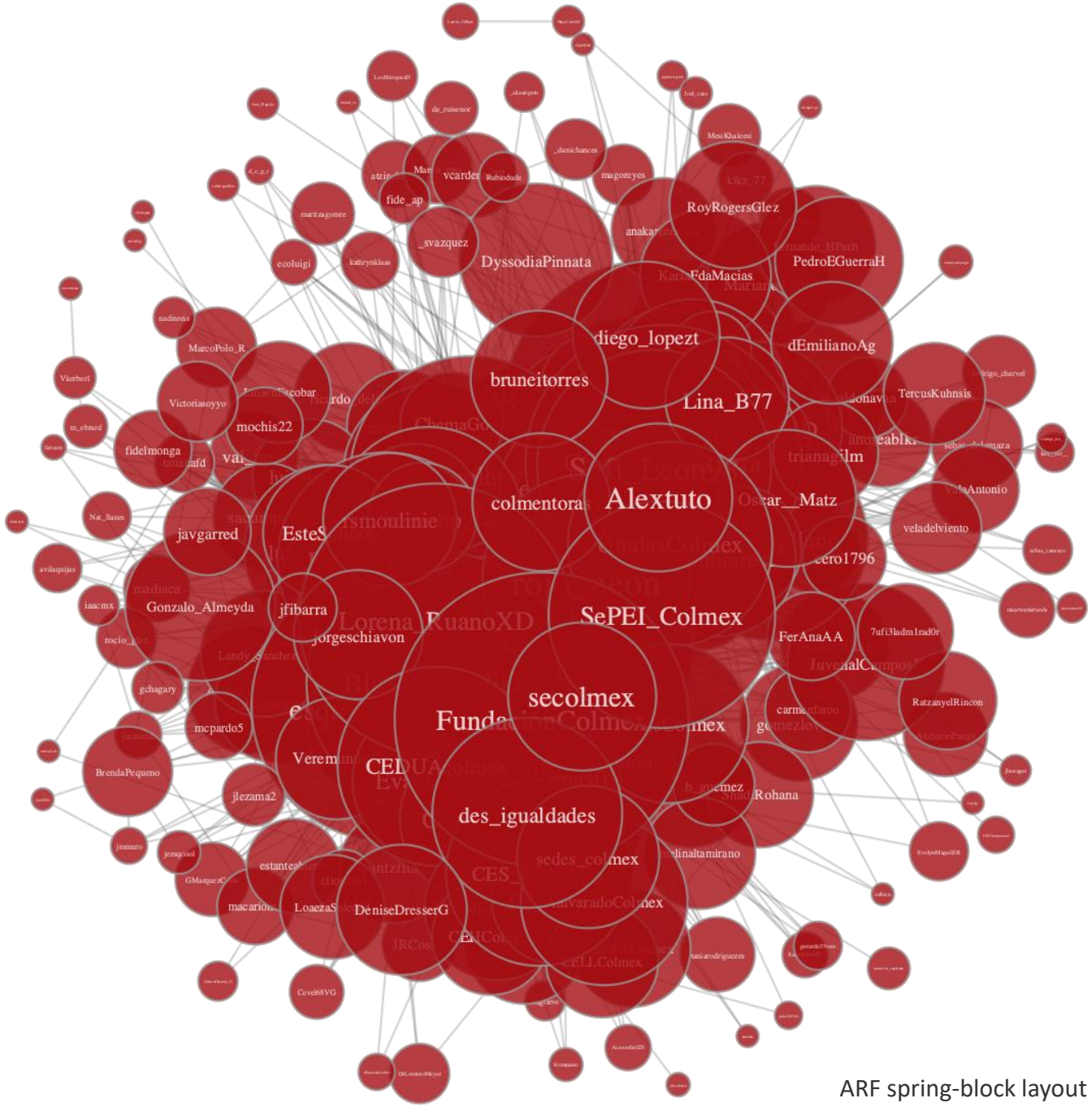
Visualización de Red interna no dirigida

Red interna con etiquetas de username.

Tamaño de nodos en función de degree centrality.

Presentamos los top 10 usuarios con mayor degree a continuación:

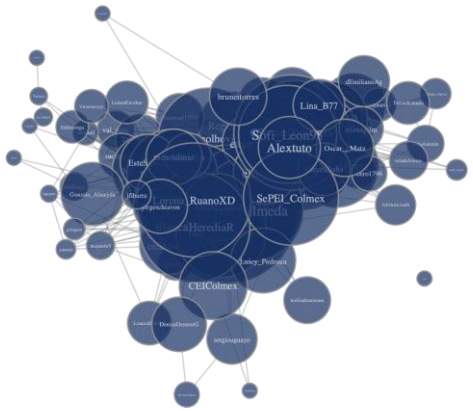




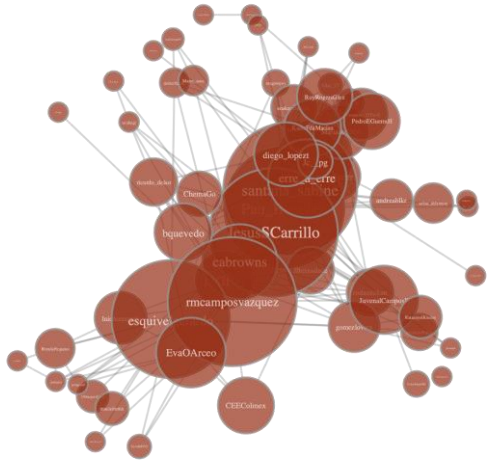
ARF spring-block layout

Red interna con etiquetas de username para los centros: Tamaño de nodos en función de degree centrality.

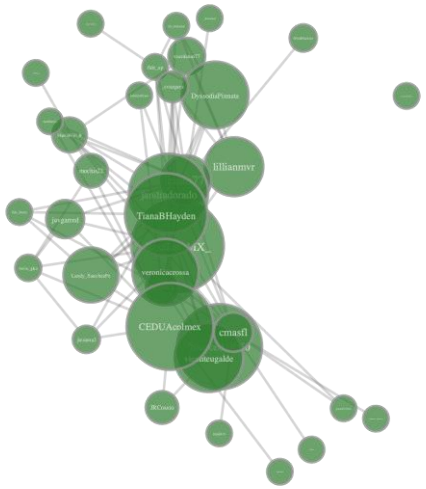
CEI



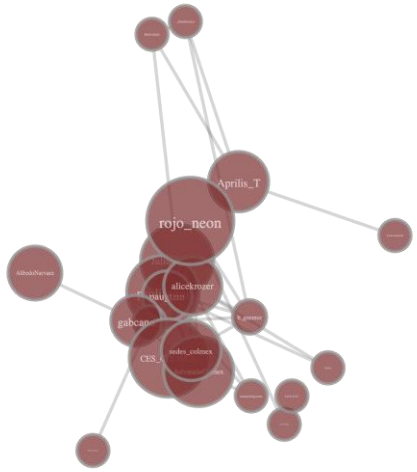
CEE



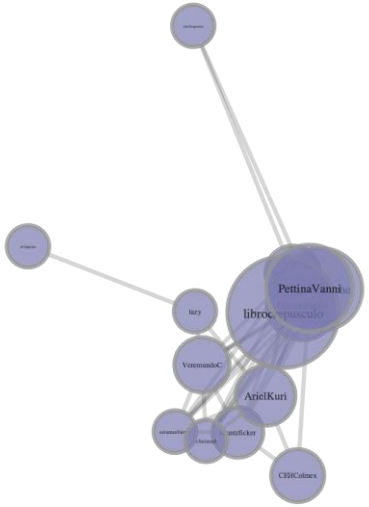
CEDUA



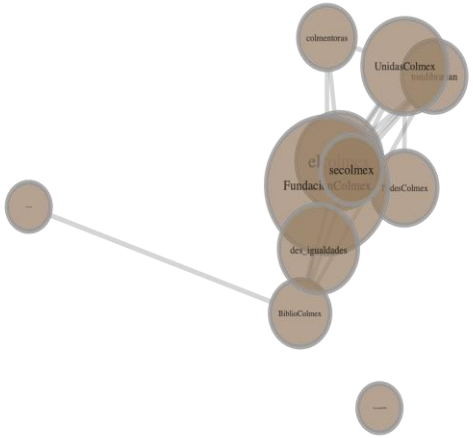
CES



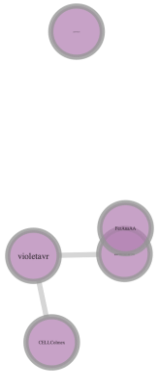
CEH



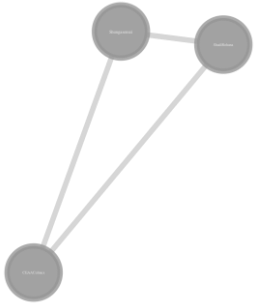
ADM

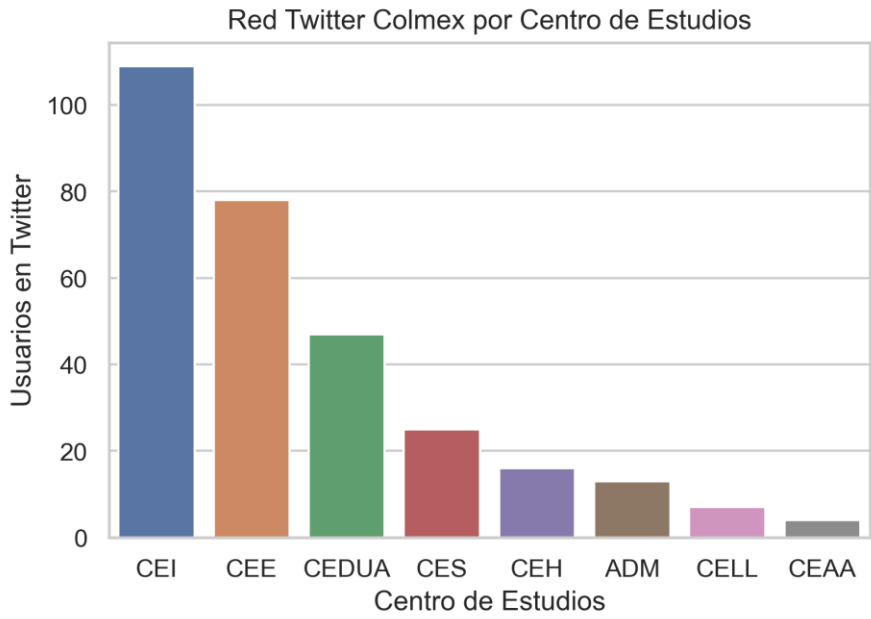


CELL



CEAA

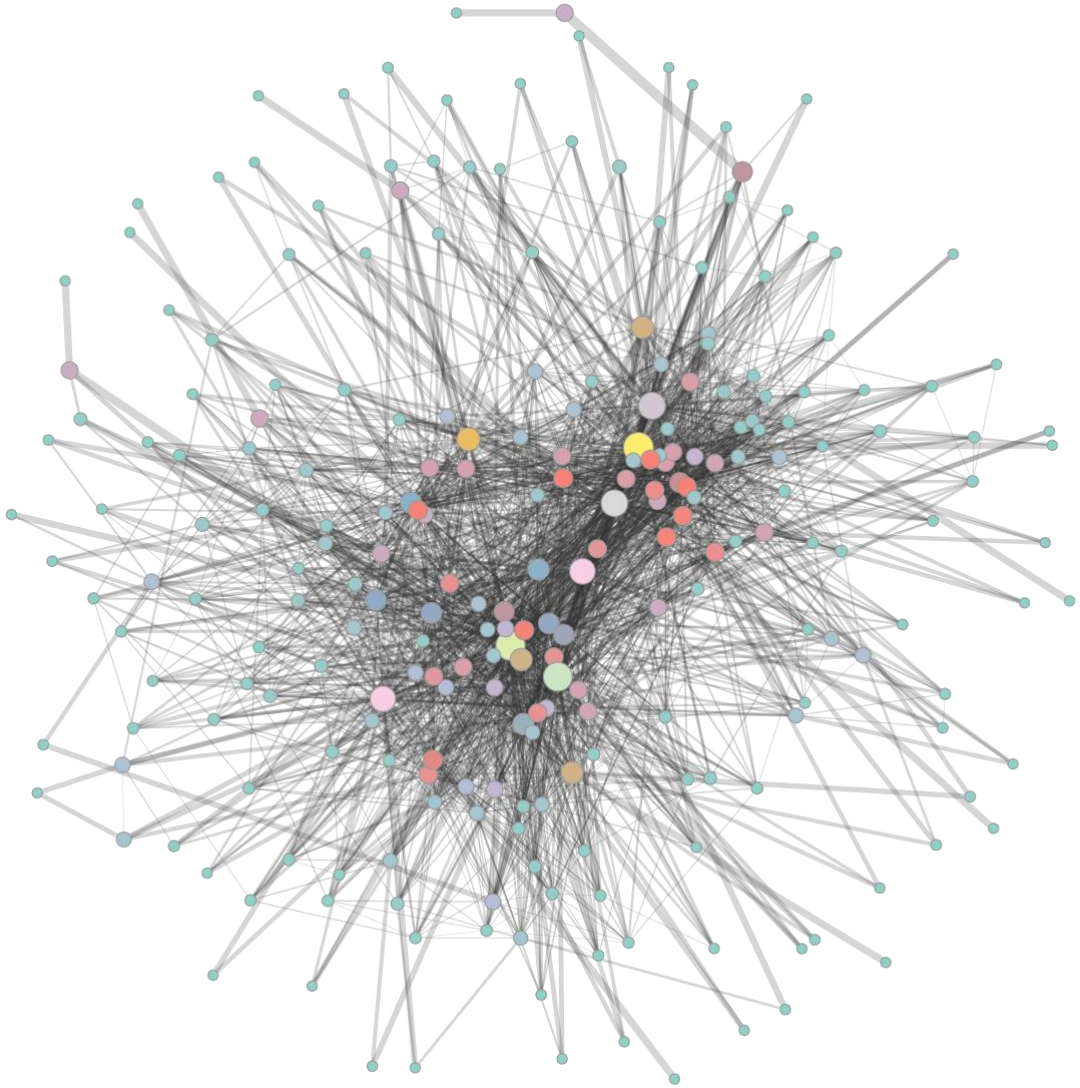




Red interna Colmex / Medidas principales	Red dirigida	Red no dirigida	Intuición
Assortativity Coefficient por status: Actual/Ex	0.119	0.174	La asortatividad es la preferencia de los nodos de una red por unirse a otros que le son similares en alguna características. Existe baja correlación entre conexiones de personas actualmente en la red con otras que formaron parte.
Assortativity Coefficient por Centro de estudios:	0.183	0.245	Hay un coeficiente positivo por centro de estudios. Existe una tendencia a juntarse con personas del mismo centro.
Global average clustering. (NetworkX)	0.471	0.48	Con un valor cercano a ½ ,la comunidad en general está agrupada.
Usuario con mayor Betweenness centrality:	0.0816	0.068	Con un grado tan bajo significa que no hay algún nodo que tenga control sobre la red, se puede pasar información por diversos caminos.
Usuario con mayor Closenness centrality:	0.846	0.622	Hay nodos, en especial en la red dirigida que pueden transmitir más eficientemente la información debido a su cercanía con los demás nodos
Máximo k-core ((k, k) D-core = k-core)	68	22	La máxima subgráfica de la red dirigida que se puede formar donde todos los nodos tengan al menos k conecciones es con 68 aristas, lo cual nos dice que la red no es tan dispersa.

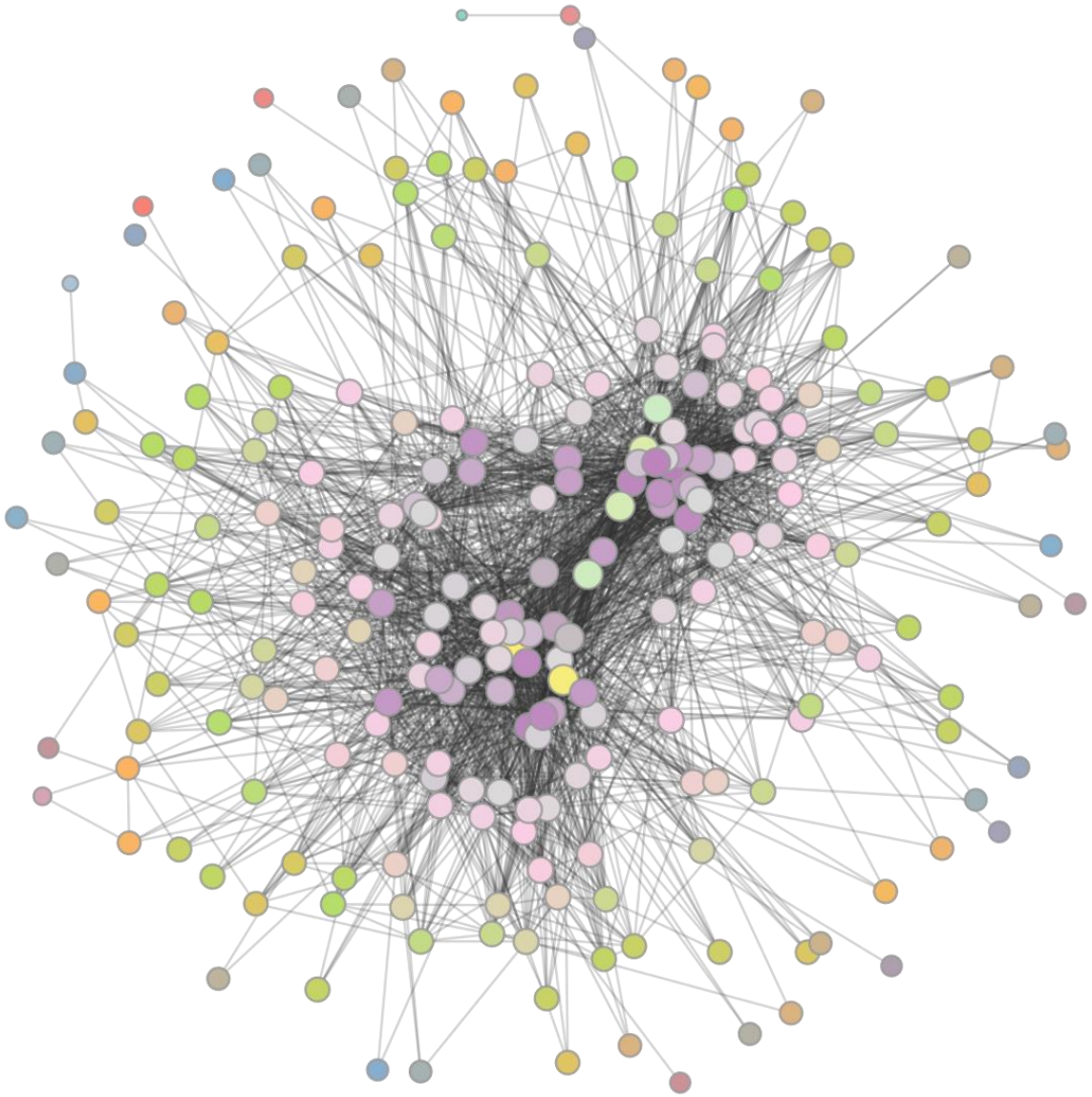
Betweenness centrality para hacer gruesos los edge y los vertex

Notemos que la gran interacción que existe entre los nodos permite que la red no esté cargada a nodos en particular, hay diversos medios para transmitir la información.



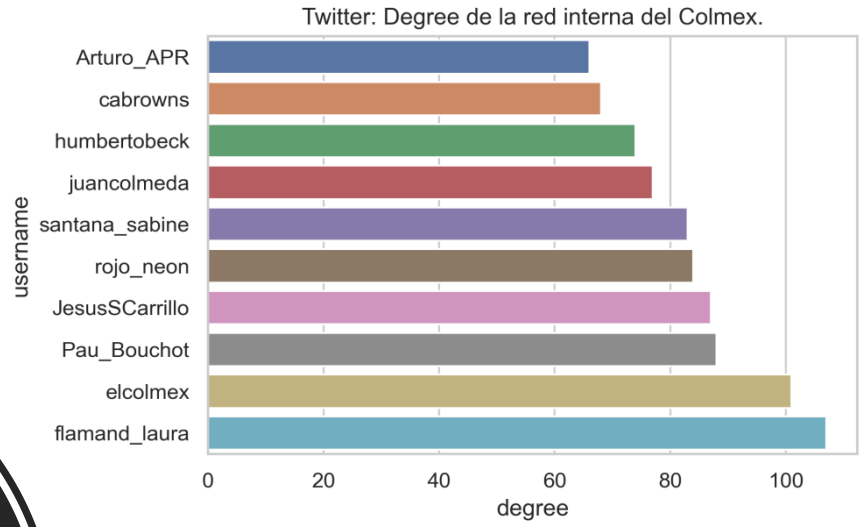
Closenness centrality para hacer gruesos los edge y los vertex

Notemos que hay gran cercanía de los nodos en relación con los demás nodos, se puede transmitir con gran facilidad la información en toda la red.

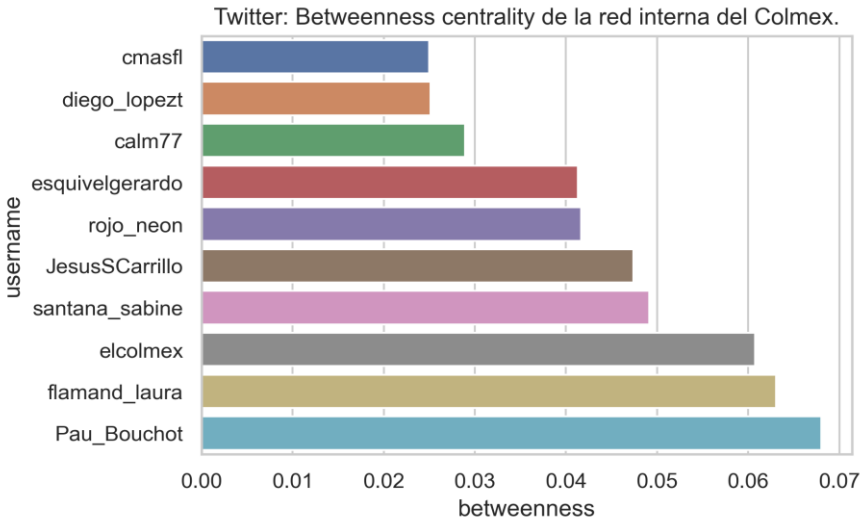


ARF spring-block layout

Resultados para usuarios principales de la red

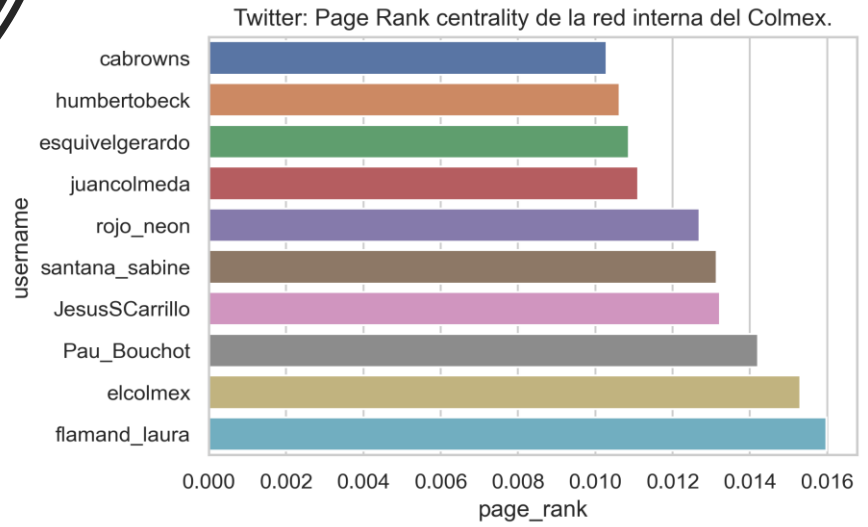


Degree centrality nos ayuda a encontrar nodos populares en la red.

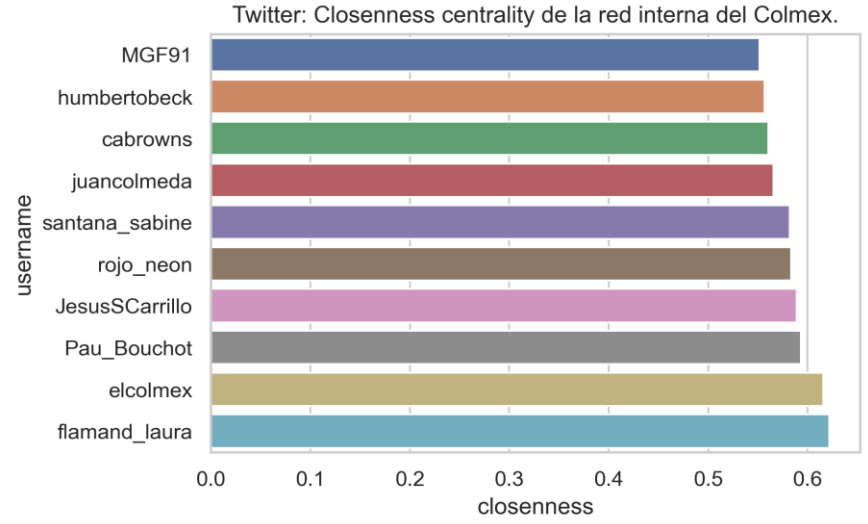


Betweeness centrality es una forma de detectar la influencia que tiene un nodo sobre el flujo de información.

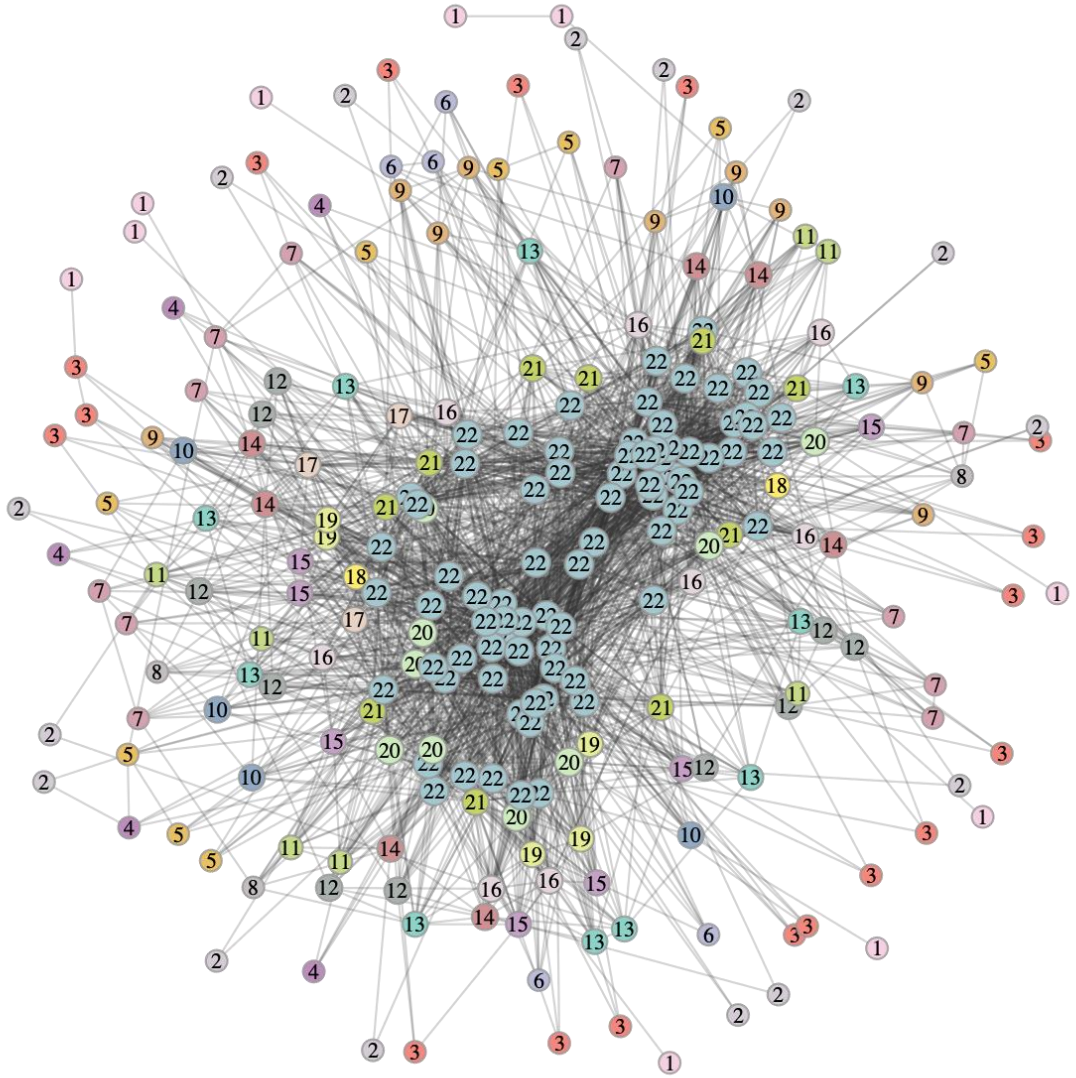
Page Rank centrality mide la influencia transitiva o la conectividad de los nodos.. Una puntuación alta significa que un nodo está conectado a otros nodos que tienen puntuaciones altas..



Closenness centrality es una forma de detectar nodos que pueden difundir información de manera muy eficiente a través de una red.



K-core decomposition:



ARF spring-block layout

Algoritmo de descomposición en k-núcleos

El algoritmo identifica una red no dirigida en la que cada subred tiene un vértice de grado k como máximo. Es decir, algún vértice en el subgrafo toca k o menos de los bordes del subgrafo.

Observamos que el máximo k-graph tiene 22 conexiones, lo que indica una red bien conectada (comunicada).

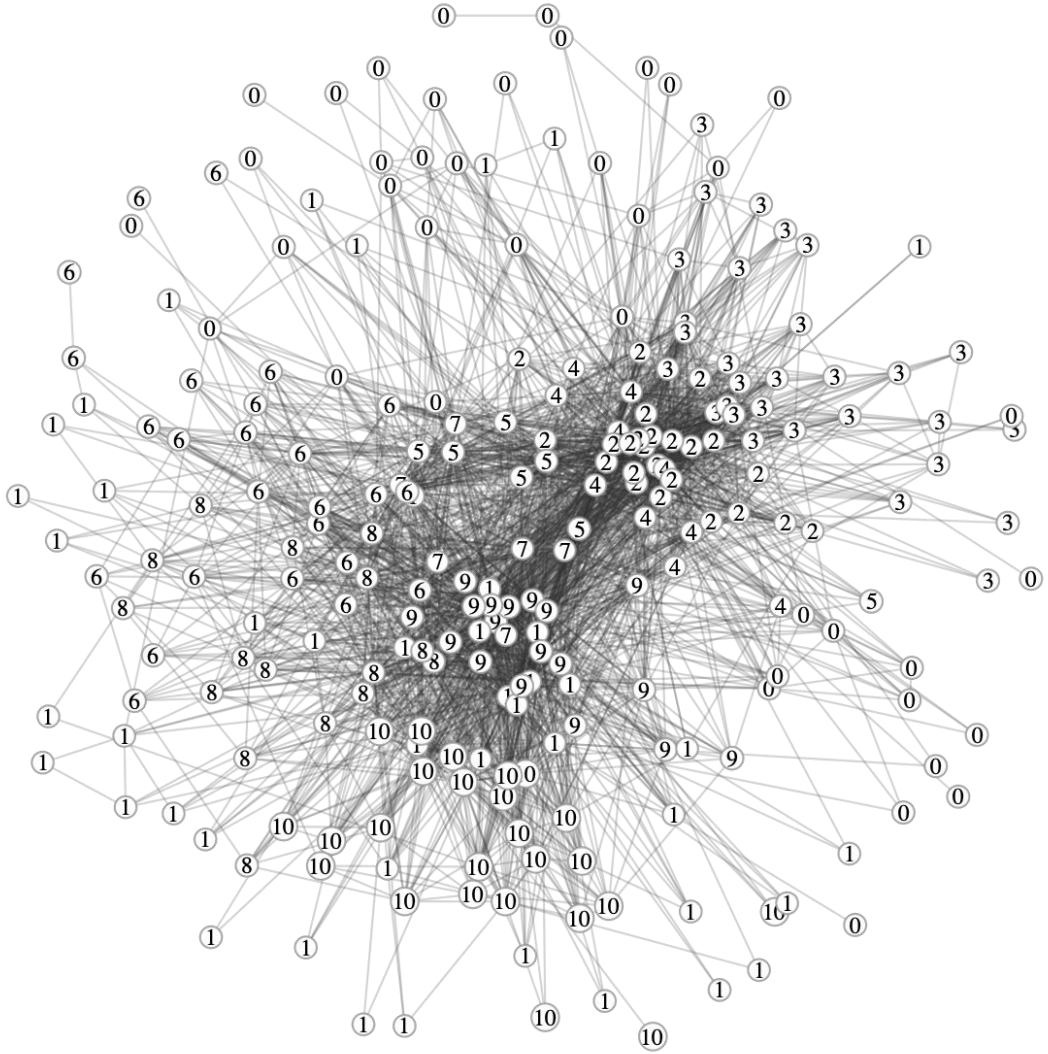
Análisis de comunidades

Community detection (minimize block model) Tiago P. Peixoto **:**

11 comunidades detectadas por este método.

Se realiza un análisis de las comunidades detectadas obteniendo la composición de cada una por centro de estudio. La siguiente tabla muestra el **centro con mayor porcentaje en cada bloque**:

Block	Centro	Usuarios del centro	% del bloque	Observaciones
0	CEE	28	70	Principalmente alumnos y profesores actuales del centro
1	CES	12	27	Única comunidad con mayoría del CES, seguido de CEDUA (24%)
2	CEI	24	100	Única comunidad exclusova del CEI, balance actuales y ex
3	CEI	15	50	Mayoría estudiantes actuales del CEI
4	CEE	4	40	Cuatro ex comunidad sin muchas conexiones
5	CEI	5	71	Mayoría del CEI, actuales y ex en misma proporción
6	CEI	23	92	Mayoría del CEI, todos ex comunidad
7	CEE	3	50	Tres estudiantes actuales sin muchas conexiones
8	CEI	8	50	La mitad son profesores del CEI (actuales y ex)
9	CEI	7	39	Mayoría estudiantes actuales del CEI
10	CEDUA	5	20	Única comunidad con mayoría del CEDUA



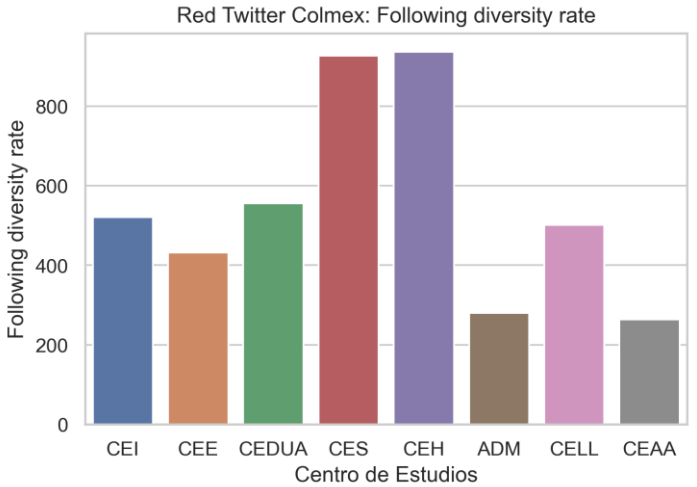
ARF spring-block layout

Objetivo secundario:

¿Qué sucede con la comunidad externa?

- Notemos que al normalizar el número de cuentas seguidas por el número de usuarios en cada centro de estudios obtenemos el **Following rate:** CES y CEH son los centros que siguen en promedio a más cuentas externas al Colmex.
- Cuando quitamos las cuentas repetidas (si dos usuarios del mismo centro siguen a una cuenta, se toma una en cuenta una sola vez) y normalizamos obtenemos el **Following diversity rate:** De nuevo el CES y CEH son los centros que siguen más cuentas únicas externas al Colmex.
- El CEE queda en 6to lugar en este índice, indicando que **los economistas del Colmex tienden en promedio a seguir menos cuentas externas que sus pares de otros centros de estudios**, superando únicamente a la Biblioteca/Administracíón y el CEAA que es el centro con menos tuiteros.

Centro de estudios	Usuarios Colmex	Cuentas externas seguidas	Cuentas externas únicas seguidas	Following rate	Following diversity rate
CEI	109	108,614	56,799	996.46	521.09
CEE	78	58,689	33,692	752.42	431.95
CEDUA	47	36,375	26,109	773.94	555.51
CES	25	31,535	23,174	1,261.40	926.96
CEH	16	20,493	14,989	1,280.81	936.81
ADM	13	4,564	3,641	351.08	280.08
CELL	7	3,838	3,511	548.29	501.57
CEAA	4	1,072	1,055	268.00	263.75



Conclusiones

- La red del Colmex en general está bien diversificada, no tiene nodos que tengan el control de la red y la información puede ser transmitida por diversos mecanismos

- Internamente es una red bien comunicada, mientras que de manera externa si hay diferencias en cuánto a cada centro de estudios.

- Los miembros del CEE tienden a seguir menos cuentas externas a la comunidad que los demás centros de estudios.

Gracias

por su atención

A Twitter Social Network Analysis of Colmex economists

López Tamayo Diego

Nazará Sosa Emilio