

Universitat Rovira i Virgili  
Escola Tècnica Superior d'Enginyeria

# **SISTEMES DISTRIBUÏTS**

## **Práctica 2**

### **AUTORS:**

DAVID NAVA FERNANDEZ  
LLUIS ORIOL COLOM NICHOLS

### **DOCENT:**

PEDRO ANTONIO GARCÍA LÓPEZ  
JOSEP SAMPE DOMENECH

**18/06/2021**

**2020 - 2021**

# ÍNDICE

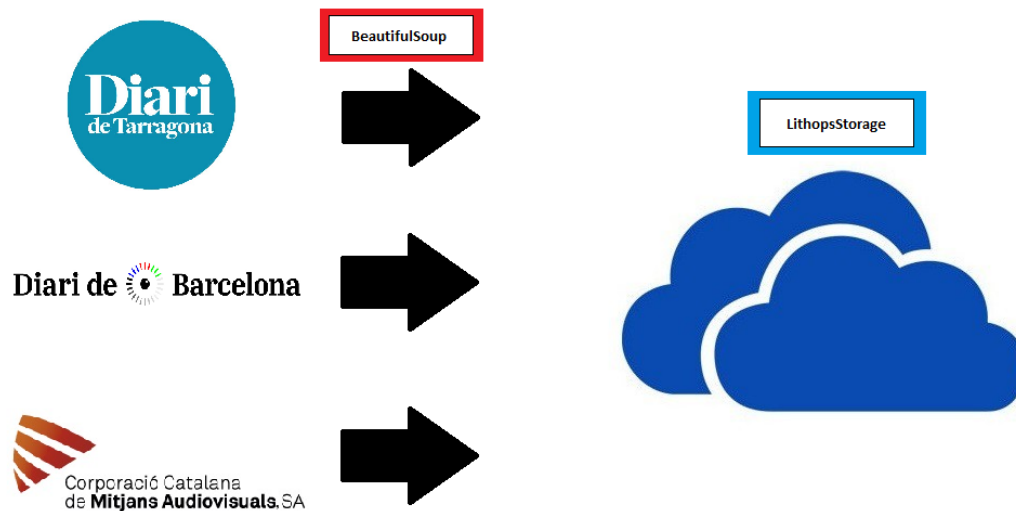
INTRODUCCIÓ	3
ARQUITECTURA I DISSENY	4
DATASET	6
NUM. FUNCIONS	8

# INTRODUCCIÓ

El nostre programa es basa en la idea de recopilar notícies de diferents diaris (diaridetarragona, diaridebarcelona i ccma), tractar-les de forma que poguem analitzar algunes dades bàsiques i guardar tota aquesta informació al cloud.

El nostre programa principal permetrà tant realitzar cerques per afegir noves notícies al nostre dataset, així com realitzar queries per buscar notícies en funció de diferents criteris.

# ARQUITECTURA I DISSENY



Gràcies a l'import de BeautifulSoup som capaços d'obtindre els links de les diferents notícies que apareixen a qualsevol dels diaris mencionats anteriorment donat un tema en concret:

(en aquest cas primer obtenim el num. de pàgines de notícies que n'hi han i a continuació procedim a obtindrer tots els links)

```
def get_links():  
    ....# Auxiliar variables.  
    ....link_to_news = []  
  
    ....# We create HTML parser.  
    ....r = requests.get('https://www.diaridebarcelona.cat/search?q='+sys.argv[1], headers=header)  
    ....soup = BeautifulSoup(r.text, 'html.parser')  
  
    ....# Get the number of pages in the website.  
    ....frame = soup.find(lambda tag: tag.name == 'li' and tag.get('class') == ['first'])  
    ....if frame is not None:  
    ....    pages = frame.find("a").get('href')  
    ....    pages = pages.split("=")[-1]  
    ....else:  
    ....    pages = 0  
  
    ....# Get the links to the news.  
    ....count = 0  
    ....for i in range(int(pages)+1):  
    ....    r = requests.get('https://www.diaridebarcelona.cat/search?q='+SEARCH_KEY+'&start='+str(i), headers=header)  
    ....    soup = BeautifulSoup(r.text, 'html.parser')  
  
    ....    for news in soup.find_all(class_='col-sm-6 col-lg-3 mb-20px mb-lg-30px'):  
    ....        count +=1  
    ....        ....# We get the link to the news page.  
    ....        ....news = news.find(class_='h1 modul-petit')  
    ....        ....link_to_news.append(news.find("a").get('href'))  
    ....  
    ....return link_to_news, count
```

Un cop obtinguts tots els links referents a un tema, procedim a recollir totes les dades d'aquesta i analitzar-les, això ho farem amb diferents processos, de forma que podem paral·lelitzar l'execució de diferents notícies:

(podem veure en el cas següent com obtenim el títol d'una notícia i la entrada corresponent)

```
def process_news(link):
    analyzer = SentimentIntensityAnalyzer()
    news_format = {}
    r = requests.get(link, headers=header)
    soup = BeautifulSoup(r.text, 'html.parser')

    # Put news link.
    news_format['link'] = link

    # Get news header.
    title = soup.find("div", class_='title-opening-section')
    if title is None:
        return # Case we are threatting an opinion...
    news_format['title'] = title.text.replace("\n", "").replace("\t", "").replace("\"", "")

    # Get news starter.
    description = soup.find("div", class_='description')
    if description is None:
        return # Case we are threatting an interview...
    news_format['starter'] = description.text.replace("\n", "").replace("\t", "").replace("\xa0", " ").replace("\"", "")
```

Podem paral·lelitzar aquesta execució gràcies a l'import de lithops.multiprocessing Pool:

```
# Start cloud multiprocessing.
with Pool() as pool:
    result = pool.map(process_news, link_to_news)
    print(result)
```

# DATASET

El nostre dataset consta del conjunt de notícies cercades amb anterioritat. Aconsegum guardar-les mitjançant:

```
...# Store the news content to the cloud COS.
...storage = Storage()
...storage.put_object(bucket='news-bucket', key=SEARCH_KEY+'diaridebarcelona/'+news_format['title'].replace(" ", "_")+'.json', body = json.dumps(news_format))
```

Instrucció la qual ens assegura que les notícies obtingudes pel diaridebarcelona tindran el format en el qual primerament s'indicarà el nom de la cerca realitzada, a continuació el nom del diari i finalment el títol de la notícia obtinguda i el format .json.

(aquest n'és un exemple del nostre dataset donada la cerca de “iogurt” i diferents resultats obtinguts pels diferents diaris)

Storage / cloud-object-storage / news-bucket

Transfers Details Actions...

<input type="checkbox"/>	iogurt/ccma/Les_indústries_là...eixat_de_comprar_Pascual.json	SQL	918 bytes	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/ccma/Llet_Nostra_augm... de_llet,_postres_i_iogurt.json	SQL	1.0 KB	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/ccma/Olesa_vota_a_fav..._el_dret_als_referèndums.json	SQL	749 bytes	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/ccma/Pascual_es_compr... instal·lar-hi_una_fàbrica.json	SQL	784 bytes	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/ccma/Premi_especial_Z... lties_i_trastorns_mentals.json	SQL	2.1 KB	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/ccma/Quatre_nens_con... de_colònies_a_Viladasens.json	SQL	2.3 KB	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/ccma/Sardines_iogurt... _mantenir_el_cervell_jove.json	SQL	4.2 KB	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/ccma/Som_els_bacteris_que_mengem.json	SQL	391 bytes	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/ccma/Tremosa_alerta_q... ítiques_contra_Catalunya.json	SQL	1.1 KB	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/ccma/UP_protestarà_a... drà_el_boicot_a_Pascual.json	SQL	1.0 KB	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/ccma/Un_grup_antifeixi... partit_ultradretà_en_alça.json	SQL	561 bytes	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/diaridebarcelona/_Ben_&_Jerry's_contra_Trump.json	SQL	984 bytes	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/diaridebarcelona/_El_p... c_dels_Aliments,_a_debat.json	SQL	12.2 KB	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/diaridebarcelona/_Jo_m... _més_feliç_al_teu_costat.json	SQL	3.3 KB	2021-06-18 11:13 AM	:
<input type="checkbox"/>	iogurt/diaridetarragona/Un_yo..._a_la_playa_de_Coma-ruga.json	SQL	1.0 KB	2021-06-18 11:13 AM	:

Si obrim qualsevol notícia, aquesta serà tota la informació que podrem observar d'ella:

```
▼ link: "https://www.ccma.cat/324/llet-nostra-augmenta-en-prop-dun-5-la-seva-facturacio-amb-la-venda-de-llet-postres-i-iogurt/noticia/1653079/"
▼ title: "Llet Nostra augmenta en prop d'un 5% la seva facturació amb la venda de llet, postres i iogurt"
▼ starter: "Barcelona (ACN).- La llet de cooperatives catalanes Llet Nostra ha augmentat durant el 2011 la seva facturació amb 21,6 milions d'euros, un 4,85% més que l'any anterior. El gruix de les vendes correspon a la llet en bri, que ha crescut un 5,35%, mentre que la novetat és que comença a notar-se l'entrada progressiva que la marca està fent des de l'estiu passat al mercat del fred, amb 15 productes de postres i iogurts, que ha suposat un 1,85% de la facturació (1,3 MEUR), durant el segon semestre de l'any. En total, s'han comercialitzat 32,7 milions de litres de llets de la granges de la cooperativa, dos milions i un 7,50% més que el 2010, segons fonts de Llet Nostra."

date: "19/03/2012"
body: " "
sentiment: 0.2732
words_number: 142
```

Aquesta conté el link d'obtenció de la notícia, el títol i l'entradeta d'aquesta, la data de posting, la descripció de la notícia (que en aquest cas no en té), el sentiment que suggereix (entre -1 i 1) i el número de paraules que conté.

## NUM. FUNCIONS

Al llarg del nostre programa podem observar diferents situacions en les quals utilitzem threads o la llibreria pool per tal de facilitar la computació de certes tasques.

En un principi, podem observar com creem tres fils per tal de d'encarregar-se de la execució de cadascun dels diaris que tractem:

```
ccma_thread = Thread(target=ccma_query)
ccma_thread.start()
dtg_thread = Thread(target=dtg_query)
dtg_thread.start()
dbc_thread = Thread(target=dbc_query)
dbc_thread.start()
```

A més, dins d'aquests threads realitzarem el pool per processar les notícies més ràpidament:

```
def dbc_query():
    link_to_news = dbc_get_links()

    # Start cloud multiprocessing.
    with Pool() as pool:
        result = pool.map(dbc_process_news, link_to_news)
    count = sum(result)
```

Finalment, també utilitzarem el pool per fer el load de les notícies que hi ha al cloud referents al tema buscat:

```
storage = Storage()
news_list = storage.list_keys(BUCKET,SEARCH_KEY+'/')
with Pool() as pool:
    news = pool.map(get_object_cloud, news_list)
return news
```