RAIn 2025
Trabajo Práctico Nº3

Tema: Unidad 2 - Crawler y Scraper

Fecha Inicio: 06/05/2025 **Fecha de Entrega:** 23/05/2025

Autor: Jorge Justino Riera **LU:** 5575 **Carrera:** Ing. Inf **Plan:** 2010

Repositorio código fuente: https://github.com/nicrom8b/rain/tree/main/tp3

1. Crawler

Planifique, diseñe y construya un crawler para recolectar al menos 30 enlaces de noticias de la sección https://www.infobae.com/deportes . De cada una de estas noticias, extraiga todos los enlaces a otras notas de deporte. Cree las estructuras de datos necesarias para mantener dichas referencias (un enlace de nota apuntando a otros enlaces de notas) y construya una representación tipo grafo de lo recolectado, preste especial atención al tema de las aristas entre los nodos, ya que pueden existir enlaces de doble entrada (entrada y salida) o enlaces "más fuerte" cuando existen más de un enlace entre 2 notas distintas. El Grafo puede ser desarrollado con cualquier herramienta o librería gráfica, pero el crawler debe ser capaz de exportar o proveer la estructura de datos necesaria para dicha representación.

Ejecución

Tomo un recorte del output donde se visualiza la noticia 30 y sus links.

```
Resumen de referencias (noticia -> cantidad de referencias):
https://www.infobae.com/deportes/2025/05/23/impacto-en-el-futbol-argentino-angel-di-maria-tomo-una-decision-respecto-a-su-futuro-profesional/ -> 11 referencias
https://www.infobae.com/deportes/2025/05/23/los-3-argentinos-que-real-madrid-tiene-en-carpeta-para-reforzar-el-equipo-de-xabi-alonso/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/23/la-accidentada-jornada-de-viernes-en-monaco-tres-banderas-rojas-doble-golpe-de-hadjar-y-la-dura-sancion-a-stroll-por-chocar-con-leclerc/ -> 10 referencias
https://www.infobae.com/deportes/2025/05/23/la-insolita-expulsion-de-un-arquero-campeon-del-mundo-en-su-ultimo-partido-como-profesional/ -> 10 referencias
https://www.infobae.com/deportes/2025/05/23/inter-enfrenta-al-como-y-napoli-recibe-a-cagliari-en-una-definicion-apasionante-para-conocer-al-campeon-de-la-serie-a/ -> 6 referencias
https://www.infobae.com/deportes/2025/05/23/el-gesto-ironico-de-verstappen-a-colapinto-en-medio-de-la-segunda-practica-del-gp-de-monaco-en-la-f1/ -> 14 referencias
https://www.infobae.com/deportes/2025/05/23/ronaldo-dejo-de-ser-propietario-del-valladolid-la-millonaria-cifra-por-la-que-vendio-sus-acciones/ -> 10 referencias
https://www.infobae.com/deportes/2025/05/23/el-sueno-de-faustino-oro-en-el-masters-de-sharjah-esta-a-un-paso-de-alcanzar-su-primera-norma-de-gran-maestro/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/23/tiene-cinco-habitaciones-y-costo-mas-de-12-millones-de-dolares-el-lujoso-yate-que-max-verstappen-compro-en-monaco/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/23/quienes-son-las-dos-promesas-de-river-que-firmaron-contrato-y-el-club-los-blindo-con-clausulas-millonarias/ -> 11 referencias
https://www.infobae.com/deportes/2025/05/23/se-iniciaron-conversaciones-el-poderoso-de-europa-que-acelero-por-franco-mastantuono-y-puso-en-alerta-a-river-plate/ -> 10 referencias
https://www.infobae.com/deportes/2025/05/22/colapinto-le-explico-a-su-companero-pierre-gasly-como-se-prepara-el-mejor-mate-el-video-de-la-reaccion-del-frances-al-probar/ -> 12 referencias
https://www.infobae.com/deportes/2025/05/23/haran-una-prueba-piloto-en-la-ciudad-de-buenos-aires-para-la-vuelta-de-los-hinchas-visitantes-tras-12-anos/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/23/las-perlitas-en-los-festejos-del-napoli-la-bandera-de-maradona-con-la-camiseta-de-boca-y-un-record-sin-precedentes/ -> 10 referencias
https://www.infobae.com/deportes/2025/05/24/de-un-casco-en-homenaje-a-senna-a-una-livery-de-la-decada-del-60-los-disenos-especiales-del-gp-de-monaco-de-la-formula-1/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/23/colapinto-hablo-tras-quedar-ultimo-en-el-primer-dia-de-actividad-en-monaco-de-las-dificultades-del-coche-a-los-problemas-por-el-trafico/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/24/infantino-afirmo-que-cristiano-ronaldo-podria-irse-del-al-nassr-para-jugar-el-mundial-de-clubes-la-furiosa-desmentida/ -> 11 referencias
https://www.infobae.com/deportes/2025/05/23/la-estrella-del-futbol-mundial-que-podria-ser-companero-de-lionel-messi-en-el-inter-miami/ -> 10 referencias
https://www.infobae.com/deportes/2025/05/22/en-europa-afirman-que-alpine-le-hizo-una-oferta-a-checo-perez-que-pasara-con-franco-colapinto-en-la-f1/ -> 10 referencias
https://www.infobae.com/deportes/2025/05/23/franco-colapinto-afrontara-las-primeras-practicas-del-gran-premio-de-monaco-con-alpine-hora-tv-y-todo-lo-que-hay-que-saber/ -> 6 referencias
https://www.infobae.com/deportes/2025/05/22/conmocion-por-la-muerte-de-una-boxeadora-de-25-anos-la-dura-acusacion-a-los-medicos-por-un-diagnostico-tardio-de-cancer/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/22/las-sorpresivas-declaraciones-de-un-jugador-de-brasil-hoy-la-seleccion-importa-cero/ -> 10 referencias
https://www.infobae.com/deportes/2025/05/23/papelon-del-santos-en-la-vuelta-de-neymar-quedo-eliminado-de-la-copa-de-brasil-ante-un-equipo-de-segunda/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/24/ancelotti-hablo-antes-de-su-despedida-del-real-madrid-el-gran-elogio-para-la-seleccion-de-brasil/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/22/tension-en-brasil-el-video-de-la-entrevista-con-philippe-coutinho-que-fue-interrumpida-por-una-balacera/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/23/el-fuerte-choque-de-frente-de-oscar-piastri-en-la-segunda-practica-del-viernes-en-monaco/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/23/la-tajante-decision-que-tomo-el-manchester-united-en-medio-de-su-crisis-institucional/ -> 12 referencias
https://www.infobae.com/deportes/2025/05/23/la-dura-sancion-que-recibio-el-jugador-de-gimnasia-de-jujuy-que-quebro-al-delantero-de-nueva-chicago/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/23/la-frase-con-la-que-flavio-briatore-respaldo-a-franco-colapinto-y-apunto-contra-jack-doohan/ -> 9 referencias
https://www.infobae.com/deportes/2025/05/23/las-revelaciones-de-fernando-tornello-la-voz-de-la-formula-1-sobre-el-futuro-de-colapinto-en-alpine/ -> 10 referencias
```

En este recorte del output, se visualiza un resumen de referencias en el que se puede ver las noticias y su cantidad de referencias.

```
Estructura de referencias exportada a references.json

Construyendo y visualizando el grafo de referencias...

Grafo exportado como grafo_referencias.png
Grafo exportado como grafo_referencias.gexf (GEXF)
(1) (* |infra:argocd)jriera:1/ (mainx) $
```

Para finalizar el output muestra, que se guarda un json la estructura de referencias y se construyen los gráficos

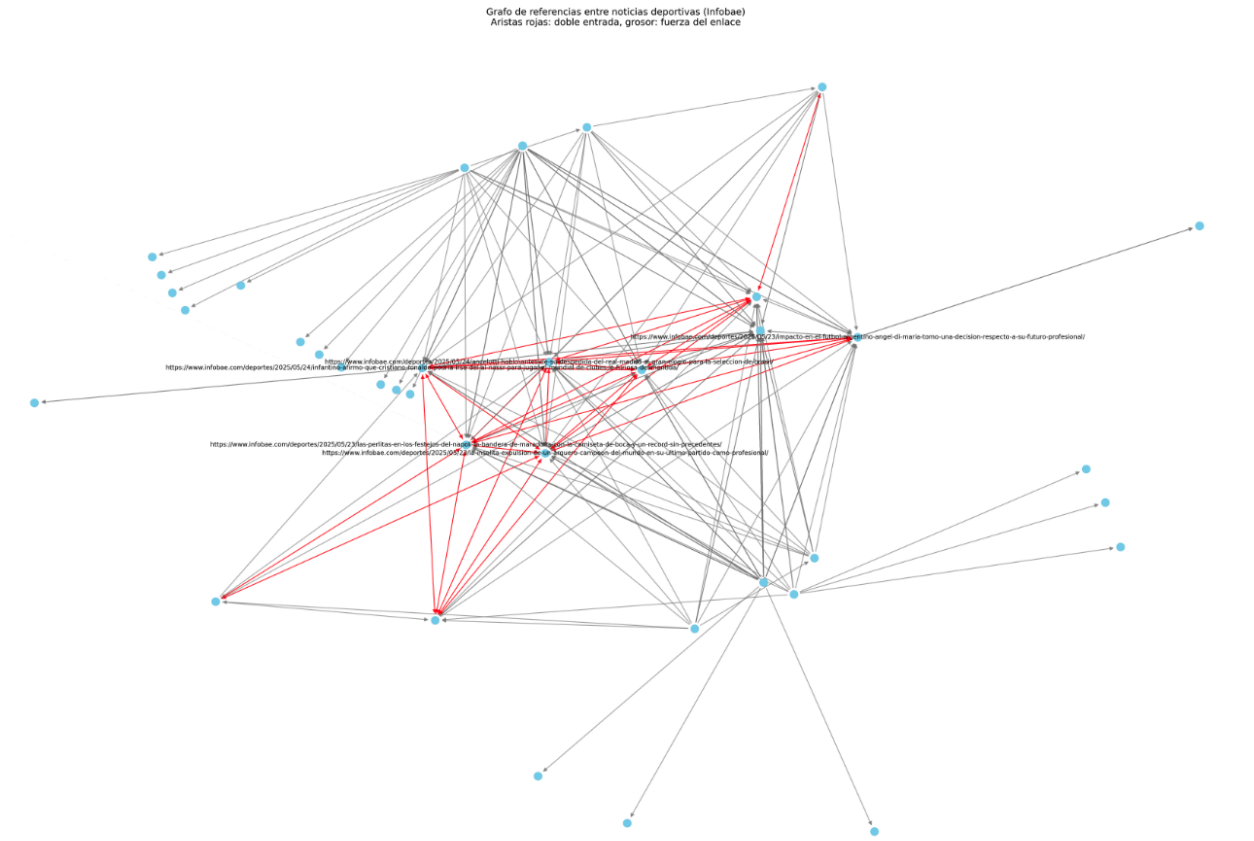


Gráfico grafo de referencias

Explicación

¿Qué representa el grafo?

- **Nodos**:

Cada nodo representa una noticia deportiva de Infobae. El identificador del nodo es la URL de la noticia.

- **Aristas (flechas):**

Una arista dirigida desde el nodo A al nodo B significa que la noticia A contiene un enlace hacia la noticia B.
El grosor de la arista indica cuántas veces A enlaza a B (fuerza del enlace).

- **Color de las aristas:**

Rojo: La arista es “doble”, es decir, existe una arista en ambos sentidos entre dos nodos (A enlaza a B y B enlaza a A).
Gris: La arista es simple, solo va en un sentido.

- **Etiquetas:**

Por defecto, solo los 5 nodos con mayor grado (más conexiones) muestran su URL como etiqueta para evitar superposición y hacer el grafo más legible.

- **Distribución:**

El layout Kamada-Kawai y el escalado de posiciones buscan que los nodos estén lo más separados posible, para que las aristas y etiquetas no se encimen.

Script

Este script es un recolector y visualizador de referencias entre noticias deportivas de Infobae. Básicamente, navega por la web, junta noticias, analiza cómo se enlazan entre sí y arma un grafo para visualización de referencias.

Librerias:

- **requests**: Para hacer las descargas de las páginas web (HTTP requests).
- **BeautifulSoup (bs4)**: Para parsear el HTML y encontrar los links dentro de cada noticia.
- **networkx**: Para construir y manipular el grafo (la red de noticias y enlaces).
- **matplotlib.pyplot**: Para dibujar y guardar la imagen del grafo.
- **urllib.parse**: Para manejar y unir URLs de manera robusta.
- **re**: Para usar expresiones regulares y filtrar solo las URLs de noticias reales.
- **json**: Para guardar la estructura de referencias en un archivo JSON.
- **scipy**: Indirectamente, porque la usa NetworkX para el layout Kamada-Kawai (mejora distribución de nodos).

Funciones Principales

- **get_news_links(base_url, min_links=30)**

Se mete en la página principal de deportes de Infobae y junta al menos 30 links de noticias reales (no secciones ni banners).

- **extract_sports_links(news_url)**

Dada una noticia, busca todos los links a otras noticias deportivas dentro de esa página. Devuelve una lista de URLs absolutas.

- **build_graph_weighted(references)**

Toma el diccionario de referencias (quién enlaza a quién) y arma un grafo dirigido, donde cada arista tiene un peso (cuántas veces A enlaza a B). Al final este grafico se exporta en formato GEXF. Para visualizarlo utilice la herramienta Gephi (https://gephi.org/)en MacOS.

- **main():** Es el corazón del programa. Hace todo el pipeline:
    - Junta las noticias.
    - Para cada noticia, busca a cuáles otras enlaza.
    - Imprime un resumen.
    - Guarda la estructura en JSON.
    - Construye el grafo y lo dibuja, ajustando la visualización para que se vea bien.
    - Exporta el grafo en formatos útiles (PNG y GEXF para Gephi).

```
def main():
    # 1. Recolectar enlaces de noticias
    print("Recolectando enlaces de noticias...")
```

```python
noticia_regex = re.compile(r"^/deportes/\d{4}/\d{2}/\d{2}/.+")
news_links = get_news_links(BASE_URL)
print(f"Se recolectaron {len(news_links)} enlaces de noticias.")
for i, url in enumerate(news_links, 1):
    print(f"[{i}] {url}")

# 2. Para cada noticia, recolectar referencias a otras noticias deportivas
reales
references = dict()  # clave: noticia, valor: lista de noticias referenciadas
for idx, news_url in enumerate(news_links, 1):
    print(f"\nNoticias deportivas referenciadas en la noticia [{idx}]:
{news_url}")
    try:
        links = extract_sports_links(news_url)
        references[news_url] = sorted(links)
        if not links:
            print("  [Sin enlaces a otras noticias deportivas en esta noticia]")
        for i, lnk in enumerate(sorted(links), 1):
            print(f"  [{i}] {lnk}")
    except Exception as e:
        print(f"  Error extrayendo enlaces de {news_url}: {e}")
        references[news_url] = []

# 3. Imprimir resumen de la estructura de referencias
print("\nResumen de referencias (noticia -> cantidad de referencias):")
for k, v in references.items():
    print(f"{k} -> {len(v)} referencias")

# 4. Exportar la estructura a JSON
with open("references.json", "w", encoding="utf-8") as f:
    json.dump(references, f, ensure_ascii=False, indent=2)
print("\nEstructura de referencias exportada a references.json")

# 5. Construir y visualizar el grafo
print("\nConstruyendo y visualizando el grafo de referencias...")
G = build_graph_weighted(references)
plt.figure(figsize=(24, 16), dpi=200)  # Aumenta aún más el tamaño de la figura
pos = nx.kamada_kawai_layout(G)
# Escalar posiciones para separar más los nodos
for k in pos:
    pos[k] = pos[k] * 2.5  # Escala las posiciones para mayor separación
```

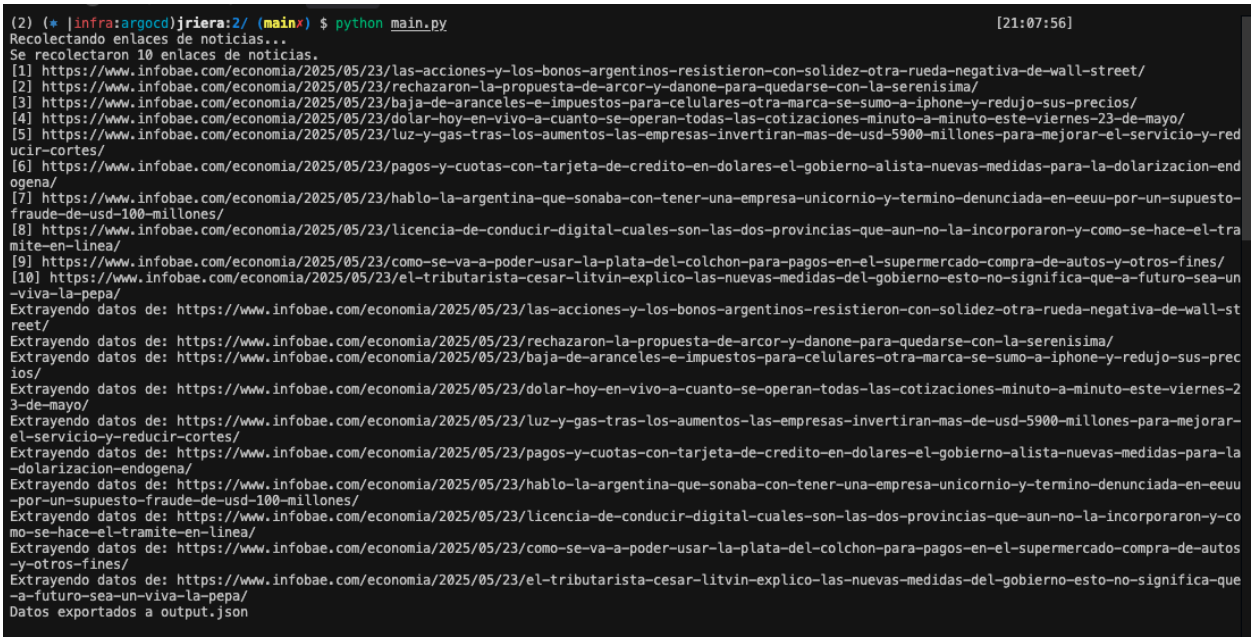
```
# Dibujar nodos
nx.draw_networkx_nodes(G, pos, node_size=100, node_color='skyblue')

# Dibujar aristas con grosor proporcional al peso
all_weights = [d['weight'] for u, v, d in G.edges(data=True)]
max_weight = max(all_weights) if all_weights else 1
edges = G.edges(data=True)
for u, v, d in edges:
    width = 1 + 4 * (d['weight'] - 1) / max_weight  # más grueso si hay más
enlaces
    color = 'red' if G.has_edge(v, u) else 'gray'  # rojo si hay doble entrada
    nx.draw_networkx_edges(G, pos, edgelist=[(u, v)], width=width,
edge_color=color, arrowsize=10, alpha=0.7)

# Etiquetas solo para los nodos con mayor grado
degrees = dict(G.degree())
top_nodes = sorted(degrees, key=degrees.get, reverse=True)[:5]
labels = {n: n for n in top_nodes}
nx.draw_networkx_labels(G, pos, labels, font_size=8)
plt.title("Grafo de referencias entre noticias deportivas (Infobae)\nAristas
rojas: doble entrada, grosor: fuerza del enlace")
plt.axis('off')
plt.tight_layout()
plt.savefig("grafo_referencias.png", dpi=300)
print("\nGrafo exportado como grafo_referencias.png")

# Exportar a GEXF (mejor formato para Gephi, la herramienta de visualización que
estoy usando en MacOS)
nx.write_gexf(G, "grafo_referencias.gexf")
print("Grafo exportado como grafo_referencias.gexf (GEXF)")

plt.show()
```

2. Scraping

Realice un web scraping de la siguiente URL: https://www.infobae.com/economia/ De esta URL recolecte las primeras 10 noticias, identificando por cada una el Título, Resumen, Autor de la nota, Listado de imágenes (ubicación del archivo) y el Cuerpo de la misma. A continuación realice un análisis textual sencillo, tokenize dichos documentos,

elimine las stop-words y liste los 100 términos más frecuentes. En el mismo sentido realice un stemming y vuelva a listar los 100 términos más frecuentes.

Ejecución



```
Top 100 términos más frecuentes en los cuerpos de las noticias:
dólares: 32
pesos: 26
millones: 25
sistema: 20
gobierno: 17
según: 16
dólar: 13
litvin: 13
oficial: 12
día: 12
ser: 12
nuevo: 12
argentina: 11
baja: 11
empresa: 11
precios: 11
may: 10
venta: 10
usd: 10
fin: 10
opción: 10
mastellone: 10
nacional: 10
impuestos: 10
explicó: 10
acciones: 9
ciento: 9
ahorros: 9
medidas: 9
argentinos: 9
cambio: 9
contrato: 9
semana: 9
valor: 9
hacia: 9
operaciones: 9
pagos: 9
mercados: 8
viernes: 8
presidente: 8
banco: 8
dinero: 8
parte: 8
manera: 8
cuotas: 8
mayor: 8
puede: 8
tarjeta: 8
```

```
Top 100 raíces (stems) más frecuentes en los cuerpos de las noticias:
dolar: 32
pag: 29
pes: 28
millon: 26
oper: 23
argentin: 22
baj: 22
qued: 21
nuev: 21
pued: 21
sistem: 21
empres: 18
oficial: 17
gobiern: 17
med: 16
segun: 16
merc: 15
inform: 15
preci: 15
hac: 15
may: 14
mayor: 14
econom: 14
cambi: 14
ingres: 14
dol: 13
part: 13
ahorr: 13
explic: 13
contrat: 13
impuest: 13
litvin: 13
declar: 12
dia: 12
cas: 12
tarjet: 12
ser: 12
contribuyent: 12
accion: 11
activ: 11
public: 11
banc: 11
deb: 11
fin: 11
opcion: 11
valor: 11
cuent: 11
gener: 11
import: 10
vent: 10
usd: 10
establec: 10
mastellon: 10
```

Explicación

El proyecto realiza un scraping (requests + BeautifulSoup) de las 10 primeras noticias de la sección economía de Infobae, extrae información estructurada de cada noticia (título, resumen, autor, imágenes, cuerpo) y realiza un análisis textual básico sobre el cuerpo de las noticias, listando los términos y raíces (stems) más frecuentes.

Las principales librerías utilizadas son:

- **requests**: Para realizar las request HTTP y descargar el HTML.
- **beautifulsoup4**: Para parsear el HTML y extraer información estructurada de las páginas (enlaces, títulos, autores, imágenes, cuerpo, etc.).
- **ntlk:** para el procesamiento del texto, con la cual nos permite tekenizar el texto en palabras, obtener el listado de stopwords en español para ser filtrados y SnowballStemmer para realizar el stemming.
- **collections.Counter**: Para contar la frecuencia de los términos.
- Json: Para exportar la estructura a archivos.

Funciones principales

- **get_news_links(base_url, max_links=10)**
  Realiza el scraping de la portada de economía y recolecta los enlaces de las primeras 10 noticias reales, filtrando por patrón de URL.

Flujo:
1. Descargar HTML → 2. Parsear HTML → 3. Buscar todos los enlaces → 4. Filtrar solo los de noticias reales (por patrón y dominio) → 5. Evitar duplicados → 6. Devolver los primeros 10

A continuacion muestro el núcleo de búsqueda y filtrado de enlaces

```
for link in soup.find_all('a', href=True):
       # Recorre todos los enlaces <a href=...> de la página
       href = link['href']  # Obtiene la URL del enlace
       # Descarta enlaces externos (que no sean de Infobae Economía)
       if href.startswith('http') and 'infobae.com/economia/' not in href:
           continue
       # Solo considera enlaces que contengan la palabra 'economia', que no sean
anclas (#) ni tengan 'undefined'
       if 'economia' in href and not href.startswith('#') and 'undefined' not in href:
           full_url = urljoin(base_url, href)  # Convierte enlaces relativos en URLs
absolutas
           path = urlparse(full_url).path      # Extrae solo la parte de la ruta de la
URL para comparar con el patrón
           if noticia_regex.match(path):       # Solo acepta URLs que tengan el
formato de noticia real
               if full_url not in news_links:  # Evita duplicados
```

```
                news_links.append(full_url) # Agrega el enlace válido a la lista
        if len(news_links) >= max_links:        # Detiene el proceso cuando ya se
tienen los 10 enlaces requeridos
            break
```

- **extract_news_data(news_url)**
  Descarga y parsea cada noticia individual, extrayendo:
    - Extraer título: Busca el <h1>.
    - Extraer resumen: Busca el <h2> o un .
    - Extraer autor: Busca en , <meta name='author'> o en JSON-LD.
    - Extraer imágenes: Busca todos los <img> relevantes y elimina duplicados.
    - Extraer cuerpo: Busca el contenedor principal del texto y concatena los párrafos.
    - Manejo de errores: Si algo falla, retorna los campos vacíos.
- **analizar_texto_cuerpos(noticias)**
    - Tokeniza todos los cuerpos de las noticias.
    - Elimina stopwords en español.
    - Cuenta la frecuencia de los términos y exporta el top 100 a top_100_terminos.txt.
    - Aplica stemming a los tokens, cuenta la frecuencia de las raíces y exporta el top 100 a top_100_stems.txt.
- **main():** orquesta todo el proceso.

Conclución

El ranking de stems nos ayuda a identificar los conceptos clave de forma clara, sin que las variaciones de género, número o conjugación nos distraigan. Es muy útil para analizar la frecuencia de palabras, crear nubes de palabras o hacer agrupaciones temáticas.

Comparando los dos rankings:

El ranking de términos nos muestra la frecuencia exacta de las palabras, lo que sirve para ver cuáles son las palabras clave tal como aparecen en el texto.
Por otro lado, el ranking de stems agrupa las variantes, mostrando la raíz del concepto y reduciendo las diferencias entre ellas. Esto es ideal para análisis semántico.