**GeoNLP REPO:

Extracting Geographical Insights from Unstructured Text**

NoteNotebook 1. webScraping2Map

Code Summary / Resumen del Código

ENGLISH:

This Jupyter Notebook contains a Python script for web scraping, text processing with spaCy, and geocoding location entities to display them on a Folium map. It also provides the option remove false positives and to export the map data to a shapefile and create a zip archive.

SPANISH:

Este cuaderno de Jupyter contiene un script en Python para realizar web scraping, procesamiento de texto con spaCy y geocodificación de entidades de ubicación para mostrarlas en un mapa de Folium. También proporciona la opción para remover falsos positivos y de exportar los datos del mapa a un archivo shapefile y crear un archivo zip.

Celda 1: Import necessary libraries / Celda 1: Importar bibliotecas necesarias

ENGLISH: In this cell, necessary libraries are imported, including requests, BeautifulSoup, spaCy, pandas, and more.

SPANISH: En esta celda, se importan las bibliotecas necesarias, incluyendo requests, BeautifulSoup, spaCy, pandas y más.

In [1]:
```
# Celda 1: Importar bibliotecas necesarias (Import necessary libraries)
import requests
from bs4 import BeautifulSoup
import spacy
import pandas as pd
from IPython.display import display
from ipywidgets import Checkbox, Button, VBox, Layout, Output, HTML, GridspecLayout
from geopy.geocoders import OpenCage
from geopy.exc import GeocoderTimedOut
import folium
import geopandas as gpd
import zipfile
import os
```

Celda 2: Define checkboxes and doc as global variables / Celda 2: Definir casillas de verificación y doc como variables globales

ENGLISH: Global variables for checkboxes and `doc` are defined in this cell.

SPANISH: Variables globales para casillas de verificación y `doc` se definen en esta celda.

In [2]:
```
# Celda 2: Definir casillas de verificación y doc como variables globales
checkboxes = []
doc = None
location_df = None  # Definir location_df como una variable global (Define location_df
```

Celda 3: Function to extract the content of a URL / Celda 3: Función para extraer el contenido de una URL

ENGLISH: This function, `scrape_url(url)`, extracts the content of a given URL and returns it as text.

SPANISH: Esta función, `scrape_url(url)`, extrae el contenido de una URL dada y lo devuelve como texto.

In [3]:
```
# Celda 3: Función para extraer el contenido de una URL
def scrape_url(url):
    try:
        print("Obteniendo contenido de la URL... (Scraping URL...)")
        response = requests.get(url)
        response.raise_for_status()
        html_content = response.content
        soup = BeautifulSoup(html_content, "html.parser")
        paragraphs = soup.find_all("p")
        return "\n".join([p.text for p in paragraphs])
    except requests.RequestException as e:
        print(f"Error al obtener la URL: {e}")
        return ""
```

Celda 4: Function to process text with spaCy based on language / Celda 4: Función para procesar texto con spaCy según el idioma

ENGLISH: The `process_text_with_spacy(text, language)` function processes text using spaCy based on the specified language ('en' or 'es').

SPANISH: La función `process_text_with_spacy(text, language)` procesa el texto utilizando spaCy según el idioma especificado ('en' o 'es').

In [4]:
```
# Celda 4: Función para procesar texto con spaCy según el idioma
def process_text_with_spacy(text, language):
    try:
        print("Procesando texto con spaCy... (Processing text with spaCy...)")
        instructions_en = "Por favor, seleccione las entidades de ubicación para mostr
        instructions_es = "Language not supported. Please choose 'en' or 'es'."

        if language.lower() == 'en':
            nlp = spacy.load('en_core_web_sm')
            instructions = instructions_en
        elif language.lower() == 'es':
            nlp = spacy.load('es_core_news_sm')
            instructions = instructions_es
        else:
```

```
            print("Idioma no compatible. Por favor, elija 'en' o 'es'. (Language not s
            return None

        print(instructions)
        return nlp(text)
    except Exception as e:
        print(f"Error al procesar el texto con spaCy: {e}")
        return None
```

Celda 5: Function to display text with interactive checkboxes / Celda 5: Función para mostrar texto con casillas de verificación interactivas

ENGLISH: This function, `display_text_with_interactive_checkboxes(text, entity_info)`, displays text with interactive checkboxes for location entities.

SPANISH: Esta función, `display_text_with_interactive_checkboxes(text, entity_info)`, muestra el texto con casillas de verificación interactivas para las entidades de ubicación.

In [5]:
```
# Celda 5: Función para mostrar texto con casillas de verificación interactivas
def display_text_with_interactive_checkboxes(text, entity_info):
    items = []

    for start, end, label, entity_text in entity_info:
        checkbox = Checkbox(value=True, description=entity_text, disabled=False)
        checkboxes.append(checkbox)
        items.append(checkbox)

    original_text_output = Output(layout=Layout(height='100%', overflow='auto'))
    with original_text_output:
        display(HTML(f"<p>{text}</p>"))

    checkboxes_output = VBox(items, layout=Layout(height='100%', overflow='auto'))
    return checkboxes, original_text_output, checkboxes_output
```

Celda 6: Function to handle the confirm button click / Celda 6: Función para manejar el clic del botón de confirmación

ENGLISH: The `on_confirm_click(change)` function handles the click event of the confirmation button, displaying selected location entities and a Folium map.

SPANISH: La función `on_confirm_click(change)` maneja el evento de clic en el botón de confirmación, mostrando las entidades de ubicación seleccionadas y un mapa de Folium.

In [6]:
```
# Celda 6: Función para manejar el clic del botón de confirmación
def on_confirm_click(change):
    global doc, location_df

    checked_ents = [box.description for box in checkboxes if box.value]
    df = pd.DataFrame({'Ubicación': checked_ents})
    display(df)

    # Geocodificar ubicaciones y mostrar el mapa de Folium
```

```
    locs = list(set([ent.text for ent in doc.ents if ent.label_ == 'LOC']))
    location_df = geocode_locations(locs)
    folium_map = create_folium_map(location_df)
    display(folium_map)

    # Exportar el mapa a un shapefile y crear un archivo zip
    if location_df is not None:
        output_shapefile = 'map_data.shp'
        output_zip = 'map_data.zip'
        export_to_shapefile_and_zip(location_df, output_shapefile, output_zip)
    else:
        print("Datos de ubicación no disponibles para exportar. (Location data not ava
```

Celda 7: Function to geocode locations / Celda 7: Función para geocodificar ubicaciones

ENGLISH: This function, `geocode_locations(locs)`, geocodes location names using the OpenCage Geocoder.

SPANISH: Esta función, `geocode_locations(locs)`, geocodifica nombres de ubicaciones utilizando el Geocodificador OpenCage.

In [7]:
```
# Celda 7: Función para geocodificar ubicaciones
def geocode_locations(locs):
    df = pd.DataFrame(columns=['lugar', 'latitud', 'longitud'])
    geolocator = OpenCage(api_key='YOUR_API_KEY')

    for place in locs:
        try:
            print(f"Geocodificando ubicación: {place}... (Geocoding location: {place}.
            location = geolocator.geocode(place, timeout=10)
        except GeocoderTimedOut:
            print("Tiempo de espera agotado para el lugar: {}".format(place))
            continue

        if location is not None:
            latitude = location.latitude
            longitude = location.longitude
            new_row = pd.DataFrame({'lugar': [place], 'latitud': [latitude], 'longitud
            df = pd.concat([df, new_row], ignore_index=True)

    return df
```

Celda 8: Function to create a Folium map / Celda 8: Función para crear un mapa de Folium

ENGLISH: The `create_folium_map(df)` function creates a Folium map based on the provided DataFrame of location data.

SPANISH: La función `create_folium_map(df)` crea un mapa de Folium basado en el DataFrame proporcionado de datos de ubicación.

In [8]:
```
# Celda 8: Función para crear un mapa de Folium
def create_folium_map(df):
    center_lat = df['latitud'].mean()
    center_long = df['longitud'].mean()
```

```
    m = folium.Map(location=[center_lat, center_long], zoom_start=1)

    for i, row in df.iterrows():
        folium.Marker(location=[row['latitud'], row['longitud']], tooltip=row['lugar']

    return m
```

Celda 9: Function to export the map data to a shapefile and create a zip archive / Celda 9: Función para exportar los datos del mapa a un shapefile y crear un archivo zip

ENGLISH: This function, `export_to_shapefile_and_zip(df, output_shapefile, output_zip)`, exports map data to a shapefile and creates a zip archive.

SPANISH: Esta función, `export_to_shapefile_and_zip(df, output_shapefile, output_zip)`, exporta los datos del mapa a un archivo shapefile y crea un archivo zip.

In [9]:
```
# Celda 9: Función para exportar los datos del mapa a un shapefile y crear un archivo
def export_to_shapefile_and_zip(df, output_shapefile, output_zip):
    try:
        # Exportar el DataFrame a un shapefile utilizando geopandas
        gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df['longitud'], df['lat
        gdf.to_file(output_shapefile)

        # Crear un archivo zip que contiene el shapefile
        with zipfile.ZipFile(output_zip, 'w') as zipf:
            zipf.write(output_shapefile, os.path.basename(output_shapefile))

        print(f"Shapefile '{output_shapefile}' y archivo zip '{output_zip}' creados ex
    except Exception as e:
        print(f"Error al exportar datos al shapefile y archivo zip: {e}")
```

Celda 10: Main function / Celda 10: Función principal

ENGLISH: The `main()` function is the main entry point of the script, where the user is prompted to input a URL and language, and the entire workflow is executed.

SPANISH: La función `main()` es el punto de entrada principal del script, donde se le solicita al usuario que ingrese una URL y un idioma, y se ejecuta todo el flujo de trabajo.

In [10]:
```
# Celda 10: Función principal (Main function)
def main():
    global doc, location_df

    # Solicitar la URL al usuario (Ask the user for the URL)
    url = input("Ingrese la URL para hacer scrapping: ")
    text = scrape_url(url)

    if text:
        # Solicitar el idioma al usuario (Ask the user for the language)
        language = input("Ingrese el idioma del texto de entrada (en/es): ")
        doc = process_text_with_spacy(text, language)

        if doc:
            print("Texto procesado con éxito. (Text processed successfully.)")
```

```
            entity_info = [(ent.start_char, ent.end_char, ent.label_, ent.text) for er
            global checkboxes
            checkboxes, original_text_output, checkboxes_output = display_text_with_ir

            confirm_button = Button(description="Confirmar Selecciones (Confirm Select
            confirm_button.on_click(on_confirm_click)

            grid = GridspecLayout(1, 2, width='100%')
            grid[0, 0] = original_text_output
            grid[0, 1] = checkboxes_output

            display(VBox([grid, confirm_button]))
            print("Función ejecutada con éxito. (Function executed successfully.)")
```

Celda 11: Usage example in your main function (Cell 10) / Celda 11: Ejemplo de uso en su función principal (Celda 10)

ENGLISH: This cell contains an example of how to use the script by calling the `main()` function.

SPANISH: Esta celda contiene un ejemplo de cómo usar el script llamando a la función `main()`.

In [11]:
```
# Celda 11: Ejemplo de uso en su función principal (Celda 10) (Usage example in your m
if __name__ == "__main__":
    main()
```

```
Ingrese la URL para hacer scrapping: https://www.las2orillas.co/la-toma-diplomatica-d
e-francia-marquez-en-africa/
Obteniendo contenido de la URL... (Scraping URL...)
Ingrese el idioma del texto de entrada (en/es): es
Procesando texto con spaCy... (Processing text with spaCy...)
Language not supported. Please choose 'en' or 'es'.
Texto procesado con éxito. (Text processed successfully.)
VBox(children=(GridspecLayout(children=(Output(layout=Layout(grid_area='widget001', h
eight='100%', overflow='a…
Función ejecutada con éxito. (Function executed successfully.)
```

	Ubicación
0	Plaza de Bolívar
1	África
2	Estados Unidos
3	Europa
4	América
5	Brasil
6	Cuba
7	Senegal
8	Burkina Faso
9	África
10	Colombia
11	Argelia
12	Egipto
13	Marruecos
14	Kenia
15	Sudáfrica
16	Ghana
17	Benín
18	Nigeria
19	España
20	Kenia
21	Sudáfrica
22	Senegal
23	La Castellana Bogotá
24	Colombia

```
Geocodificando ubicación: La Castellana Bogotá... (Geocoding location: La Castellana
Bogotá...)
Geocodificando ubicación: Sudáfrica... (Geocoding location: Sudáfrica...)
Geocodificando ubicación: Kenia... (Geocoding location: Kenia...)
Geocodificando ubicación: Senegal... (Geocoding location: Senegal...)
Geocodificando ubicación: África... (Geocoding location: África...)
Geocodificando ubicación: Burkina Faso... (Geocoding location: Burkina Faso...)
Geocodificando ubicación: .Publicidad... (Geocoding location: .Publicidad...)
Geocodificando ubicación: Actualmente... (Geocoding location: Actualmente...)
Geocodificando ubicación: Unión... (Geocoding location: Unión...)
Geocodificando ubicación: Europa... (Geocoding location: Europa...)
Geocodificando ubicación: Benín... (Geocoding location: Benín...)
Geocodificando ubicación: Brasil... (Geocoding location: Brasil...)
Geocodificando ubicación: Pauta... (Geocoding location: Pauta...)
Geocodificando ubicación: Nigeria... (Geocoding location: Nigeria...)
Geocodificando ubicación: Márquez... (Geocoding location: Márquez...)
Geocodificando ubicación: España... (Geocoding location: España...)
Geocodificando ubicación: Argelia... (Geocoding location: Argelia...)
Geocodificando ubicación: Francia... (Geocoding location: Francia...)
Geocodificando ubicación: Egipto... (Geocoding location: Egipto...)
Geocodificando ubicación: Colombia... (Geocoding location: Colombia...)
Geocodificando ubicación: América... (Geocoding location: América...)
Geocodificando ubicación: Cuba... (Geocoding location: Cuba...)
Geocodificando ubicación: Marruecos... (Geocoding location: Marruecos...)
Geocodificando ubicación: Estados Unidos... (Geocoding location: Estados Unidos...)
Geocodificando ubicación: Ghana... (Geocoding location: Ghana...)
Geocodificando ubicación: Gobierno prometió... (Geocoding location: Gobierno prometi
ó...)
Geocodificando ubicación: Plaza de Bolívar... (Geocoding location: Plaza de Bolíva
r...)
```


```
Shapefile 'map_data.shp' y archivo zip 'map_data.zip' creados exitosamente. (Shapefil
e 'map_data.shp' and zip archive 'map_data.zip' created successfully.)
```