

# Proyecto Final

Natasha Pineda Loaiciga

## Table of contents

0.1	Reflexión . . . . .	6
0.2	Quarto . . . . .	6

##Objetivo Demostrar la aplicación de técnicas de web scraping, desarrollo de APIs, visualización y análisis de datos utilizando Selenium, Flask, Pandas y Seaborn para analizar un conjunto de datos obtenido de la web.

```
import os
import pandas as pd
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

def scrape_data():
    PAGINA_PRINCIPAL = "https://www.scrapethissite.com/pages/simple/"
    navegador = webdriver.Firefox()
    navegador.get(PAGINA_PRINCIPAL)
    navegador.implicitly_wait(10)

    datos = []
    try:
        paises = WebDriverWait(navegador, 10).until(
            EC.presence_of_all_elements_located((By.CSS_SELECTOR, '.country'))
        )
        for pais in paises:
            nombre = pais.find_element(By.CSS_SELECTOR, ".country-name").text
            capital = pais.find_element(By.CSS_SELECTOR, ".country-capital").text
            poblacion = pais.find_element(By.CSS_SELECTOR, ".country-population").text
            superficie = pais.find_element(By.CSS_SELECTOR, ".country-area").text
```

```

        datos.append({
            'nombre': nombre,
            'capital': capital,
            'poblacion': poblacion,
            'superficie': superficie
        })
    except Exception as e:
        raise e
    finally:
        navegador.quit()

df = pd.DataFrame(datos)
return df

df = scrape_data()
file_path = os.path.join(os.getcwd(), "países_exportados.csv")
df.to_csv(file_path, index=False)
df.head()

```

	nombre	capital	poblacion	superficie
0	Andorra	Andorra la Vella	84000	468.0
1	United Arab Emirates	Abu Dhabi	4975593	82880.0
2	Afghanistan	Kabul	29121286	647500.0
3	Antigua and Barbuda	St. John's	86754	443.0
4	Anguilla	The Valley	13254	102.0

```

from flask import Flask, jsonify, request

app = Flask(__name__)

@app.route('/datos', methods=['GET'])
def obtener_datos():
    df = pd.read_csv("países_exportados.csv")
    min_poblacion = request.args.get('min_poblacion', default=0, type=int)
    datos_filtrados = df[df['poblacion'].astype(int) > min_poblacion]
    resultado = datos_filtrados.to_dict(orient='records')
    return jsonify(resultado)

# Ejecutar Flask en segundo plano
from threading import Thread

```

```
server = Thread(target=lambda: app.run(debug=False, use_reloader=False))
server.start()
```

\* Serving Flask app '\_\_main\_\_'

```
import requests
import seaborn as sns
import matplotlib.pyplot as plt

url = 'http://127.0.0.1:5000/datos'
response = requests.get(url)

if response.status_code == 200:
    datos = response.json()
    df = pd.DataFrame(datos)
else:
    print("Error al consumir la API")
    df = pd.DataFrame()

if not df.empty:
    df['continente'] = df['nombre'].apply(lambda x: 'América' if x in ['Canadá', 'México', 'I']

    # Gráfico Categórico: Población por Continente
    plt.figure(figsize=(10, 6))
    sns.barplot(x='continente', y='poblacion', data=df, ci=None)
    plt.title('Población por Continente')
    plt.show()

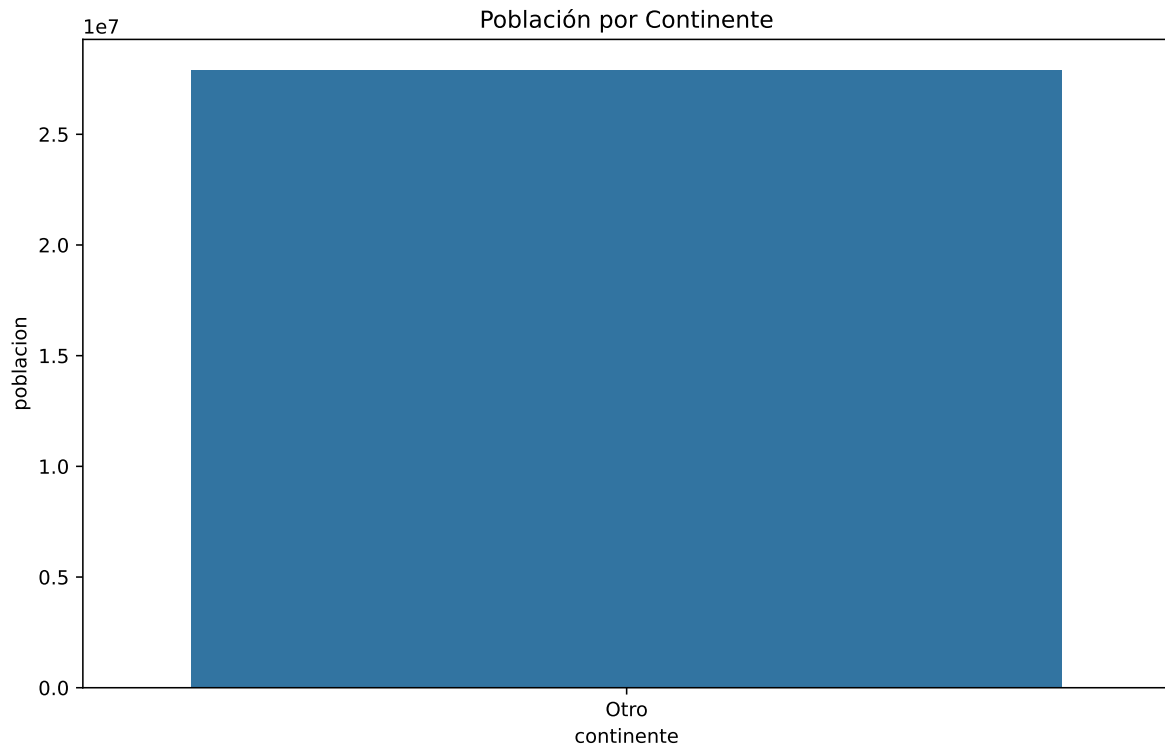
    # Gráfico Relacional: Relación entre Superficie y Población
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x='superficie', y='poblacion', data=df)
    plt.title('Relación entre Superficie y Población')
    plt.xlabel('Superficie (km²)')
    plt.ylabel('Población')
    plt.show()

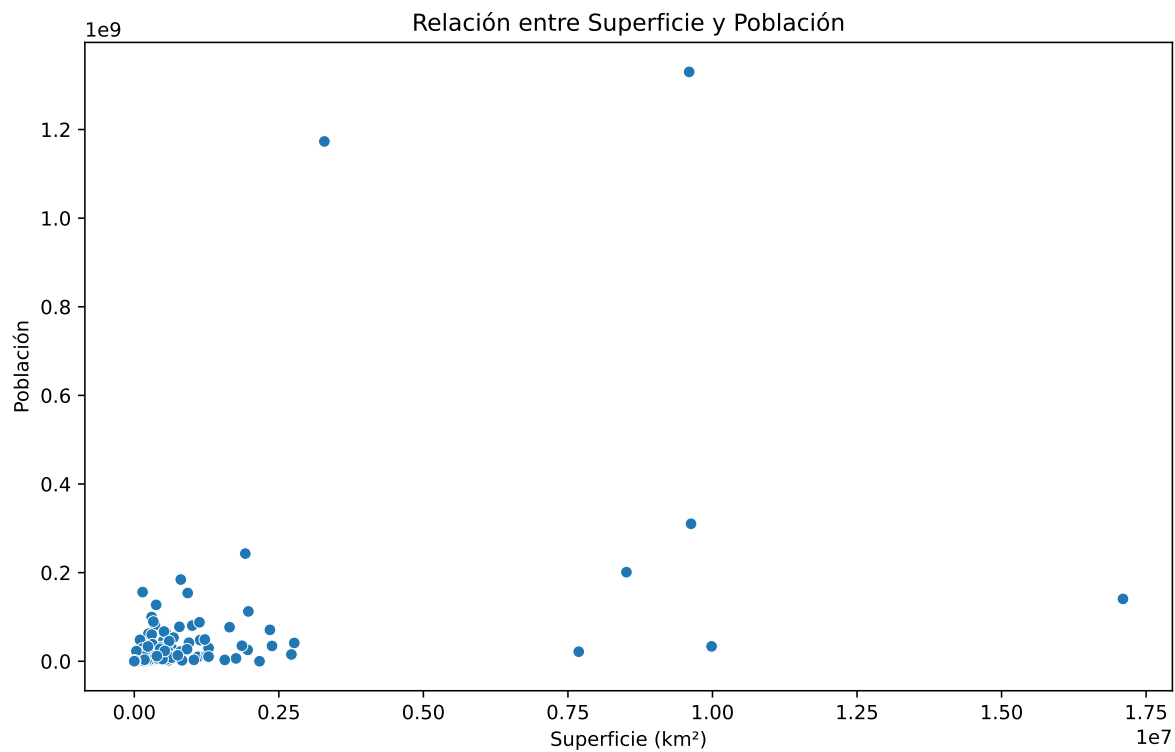
    # Gráfico de Distribución: Distribución de la Población
    plt.figure(figsize=(10, 6))
    sns.histplot(df['poblacion'], kde=True)
    plt.title('Distribución de la Población')
    plt.xlabel('Población')
    plt.show()
```

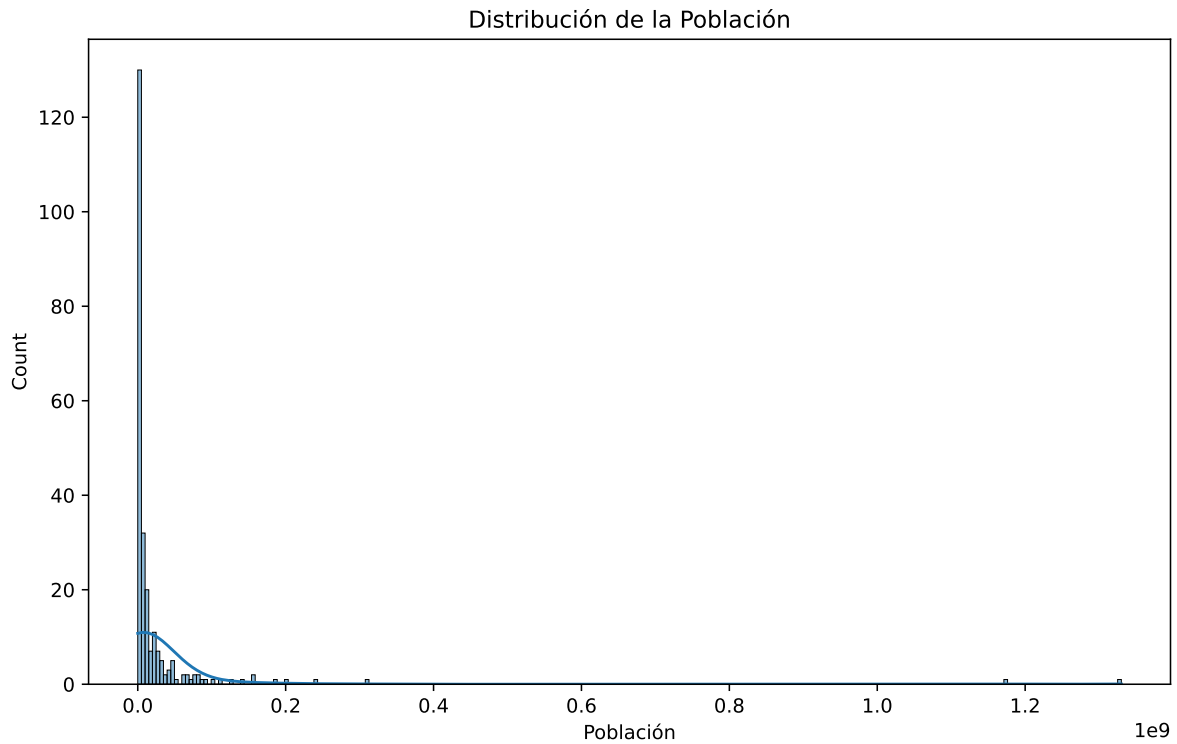
C:\Users\Natasha\AppData\Local\Temp\ipykernel\_6844\3903254270.py:20: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

```
sns.barplot(x='continente', y='poblacion', data=df, ci=None)
```







## 0.1 Reflexión

Profe muchas gracias por la paciencia este cuatrimestre. Mi reflexion es que necesito mas tiempo para practicar. Porque cuando lo logro me siento muy bien. Pero si me frustro cuando no lo puedo realizar. Espero nos puedas ayudar con videos estas vacaciones para refrescar lo aprendizaje y que no se me vaya a olvidar. Saludos

## 0.2 Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.