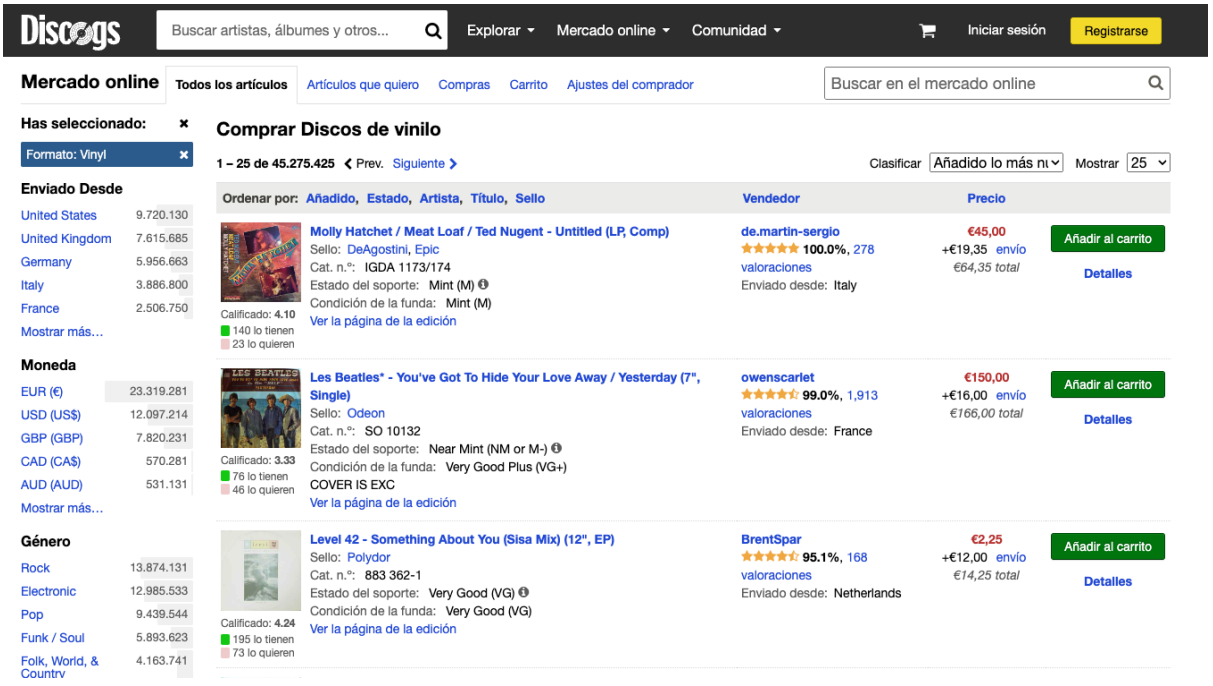
Práctica 1: Recopilación de canciones de Michael Jackson

1. Contexto

En esta práctica se realiza un scrapper del sitio web **Discogs** (https://www.discogs.com/es/), plataforma para descubrir y coleccionar música. Se trata de una gran base de datos que contiene información sobre 15 millones de releases, 8 millones de artistas y 1.7 millones de sellos discográficos. Así mismo, actúa como marketplace, permitiendo a los usuarios vender, comprar e intercambiar música en diversos formatos: CD, Vinilo, etc.



Discos de vinilo a la venta recién añadidos en Discogs

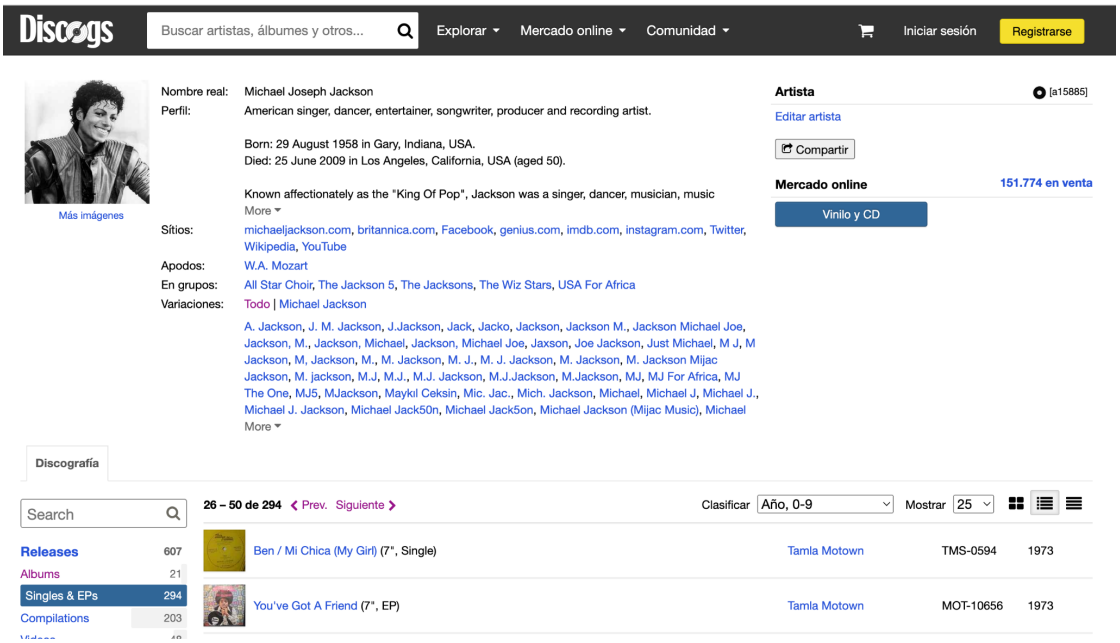
2. Título

En esta práctica obtendremos el listado de canciones de Michael Jackson mediante scrapping de la página Discogs

De esta forma, un título adecuado sería. “**Singles y EPs de Michael Jackson**”

3. Descripción del dataset

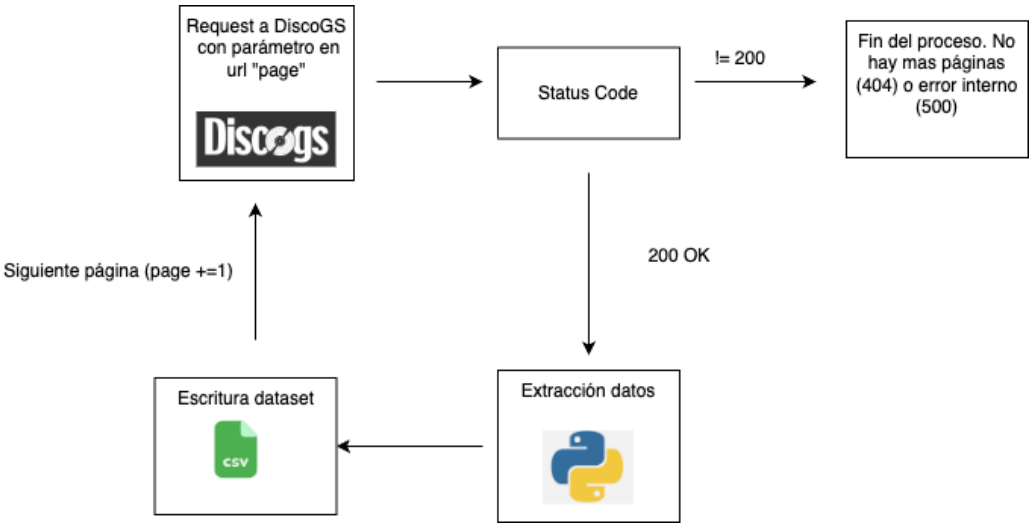
En este dataset se obtendrá la lista completa de singles de Michael Jackson. En total, contendrá 294 registros, dónde se indicará información para cada single. Para cada single, extraemos su título, artista, año, país y sello discográfico. Guardaremos los datos en un archivo csv.



Página de dónde se obtendrán los datos

4. Representación gráfica

Esquema del funcionamiento del scrapper



5. Contenido

El dataset se guarda en formato CSV y contiene los siguientes campos, que indican información sobre cada uno de los singles.

Campo	Descripción	Tipo de datos	Ejemplo
title	Título del single	Texto	“Rock With You”
artist	Artista(s)	Texto	“Michael Jackson / Anita Ward”
year	Año de publicación	Número	1972
country	País de publicación	Texto	“Brazil”
label	Sello Discográfico	Texto	“Motown”

Contiene datos comprendidos entre 1972 y 2020. Aunque Michael Jackson falleció en 2009, se realizaron algunos lanzamientos de singles y colaboraciones de forma póstuma.

6. Propietario

No se han encontrado datasets similares a este. Además, al tratarse del listado de canciones de un artista, considero que los datos son públicos y no tienen propietario. Tendrán propietario los temas en sí, pero no un listado de todos ellos, que es lo que se obtiene en esta práctica.

Desde un punto de vista ético, no hallamos ningún inconveniente. Los datos extraídos únicamente se pueden emplear de forma lúdica y divulgativa, sin ningún fin comercial o militar.

7. Inspiración

Recientemente he recuperado una cadena de música con tocadiscos del trastero de mis padres, junto con una colección de vinilos de los años 80

Cadena de música como la mía

Esto ha despertado en mí un interés por el coleccionismo de música, concretamente de vinilos, con lo cuál suelo consultar bastante la web de **Discogs** para comprar vinilos.

Como aficionado a la música, me parece interesante poder obtener programáticamente una lista de creaciones de un artista. De la misma forma, se podría configurar el scrapper para obtener álbumes, compilatorios, grabaciones, etc.

Primeramente probé a scrapear vinilos de diversos artistas, pero obtenía un dataset demasiado pequeño (~10 registros en el mejor de los casos) y no me permitía implementar la navegación en el scrapper, ya que todo se mostraba en la misma página de resultados.

He escogido obtener Singles y EP de Michael Jackson porque se trata de un artista prolífico, ya que así se obtiene un buen dataset y puedo navegar entre páginas con el scraper

1 – 25 de 294 ❮ Prev. **1** 2 3 … 11 12 Siguiente ❯

Páginas de resultados para la búsqueda de Singles y EPs de Michael Jackson

8. Licencia

Una licencia adecuada para estos datos sería **CCO: Public Domain License.**

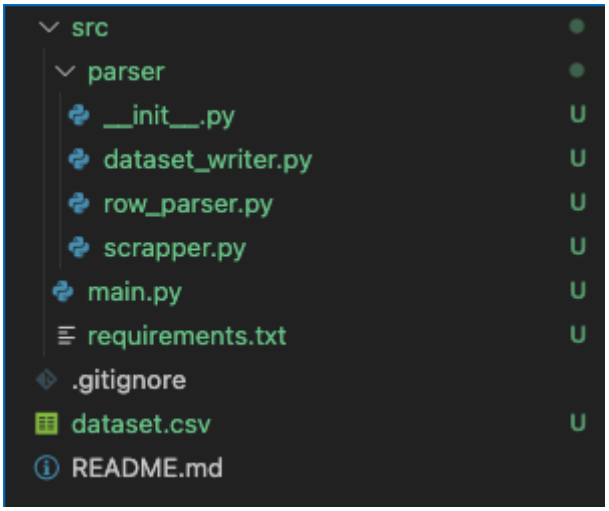
Esta licencia desiste de cualquier derecho de propiedad intelectual, liberando el documento al dominio público. Permite la copia, modificación, distribución y comunicación pública, incluso para fines comerciales, sin autorización previa.

Por otra parte, no se ofrece ninguna garantía respecto a la obra, y el autor renuncia a cualquier responsabilidad por el uso de la obra.

Considero que mi dataset “Singles y EPs de Michael Jackson” se ajusta a esta licencia, ya que los datos se encuentran fácilmente disponibles en internet y són de dominio público.

9. Código

El código se organiza mediante la siguiente estructura de archivos y carpetas



Clase Scrapper

Realiza la petición para obtener el HTML de la página y convertirlo con BeautifulSoup.

Luego, itera a través de las páginas de resultados mediante la modificación del query param “page” de la URL.

Para cada página de resultados, escribe los datos en un CSV mediante la clase Dataset, que se muestra en el siguiente apartado.

Cuando el código de respuesta es diferente a 200, normalmente 404 para indicar que ya no hay más páginas, el scrapper se para de ejecutar.

Así mismo, muestra logs por pantalla sobre el progreso de scrapping.

El scrapper incluye las siguientes medidas para evitar ser bloqueado:

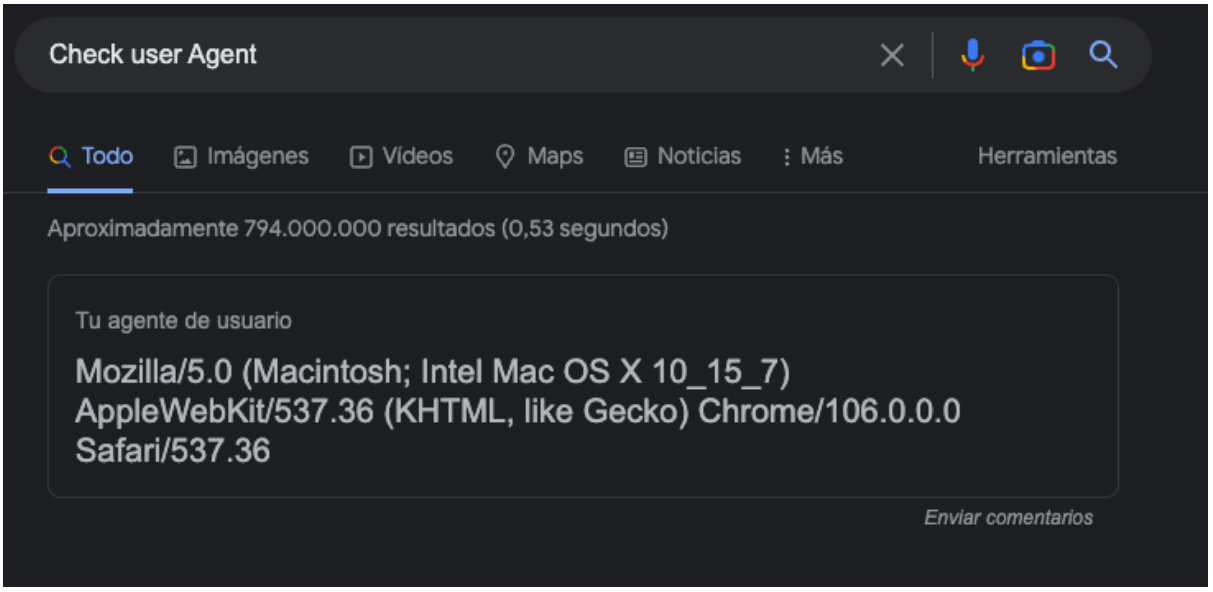
- Espera de 2 segundos entre peticiones

Se realiza entre llamadas a la página web mediante el comando sleep(2)

```
while True:
    page = self._fetch_page(page_num)

    if page.status_code != 200:
        print(f'Finished! Results: {self.total_results}')
        break

    print(f'--- Scrapping page {page_num} ---')

    self._scrape_page(page)
    page_num += 1
    sleep(2)

return
```

- Modificación del user-agent

Obtengo mi user-agent real y lo inserto en la llamada al sitio web



Clase DatasetWriter

Se encarga de crear el archivo del dataset y permite escribir filas. Utiliza la librería **csv** de Python.

src > parser > dataset_writer.py > ...

```python
import csv


class DatasetWriter:
    headers = ['title', 'artist', 'year', 'country', 'label']

    def init(self):
        with open('dataset.csv', 'w') as dataset:
            writer = csv.writer(dataset)
            writer.writerow(self.headers)

    def write_row(self, row):
        with open('dataset.csv', 'a') as dataset:
            writer = csv.writer(dataset)
            writer.writerow(row)

```

Clase TableRowParser

Clase que permite extraer la información de cada resultado de búsqueda mediante BeautifulSoup.

src > parser > row_parser.py > ...

```python
class TableRowParser:
    def __init__(self, tr):
        self.tr = tr

    def get_artist(self):
        return self.tr.find('td', class_='artist').get_text()

    def get_title(self):
        title_data = self.tr.find('td', class_='title')
        return title_data.find('a', recursive=False).get_text()

    def get_label(self):
        return self.tr.find('td', class_="label").get_text()

    def get_country(self):
        return self.tr.find('td', class_="country").get_text()

    def get_year(self):
        return self.tr.find('td', class_="year").get_text()

```

Uso de TableRowParser en Scrapper

```
for tr in results:
    parser = TableRowParser(tr)

    title = parser.get_title()
    artist = parser.get_artist()
    year = parser.get_year()
    country = parser.get_country()
    label = parser.get_label()

    self.dataset.write_row([title, artist, year, country, label])
```

Archivo main.py

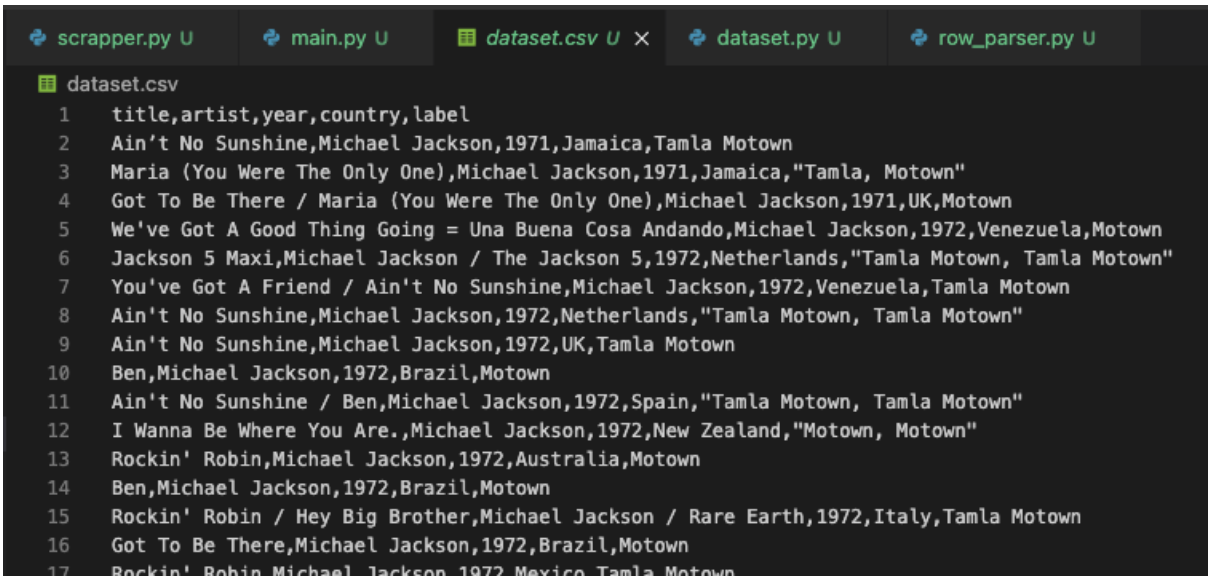
Archivo principal, dónde se crea una nueva instancia de la clase Scrapper y se ejecuta el método **scrape()** para iniciar el proceso.

```
src > main.py > ...

1   from parser.scrapper import PageScrapper
2
3   scrapper = PageScrapper()
4
5   scrapper.scrape()
6
```

Cuando iniciamos el scrapper, se muestran los siguientes logs de progreso

```
scrapper git:(master) ✗ /usr/bin/python3 /Users/daniel/Personal/scrapper/src/main.py
--- Scrapping page 1 ---
Results: 25
--- Scrapping page 2 ---
Results: 25
--- Scrapping page 3 ---
Results: 25
--- Scrapping page 4 ---
Results: 25
--- Scrapping page 5 ---
Results: 25
--- Scrapping page 6 ---
Results: 25
--- Scrapping page 7 ---
Results: 25
--- Scrapping page 8 ---
Results: 25
--- Scrapping page 9 ---
Results: 25
--- Scrapping page 10 ---
Results: 25
--- Scrapping page 11 ---
Results: 25
--- Scrapping page 12 ---
Results: 19
Finished! Results: 294
```

Así mismo, se ha generado el archivo dataset.csv



10. Dataset

El dataset ha sido subido a Zenodo y es accesible mediante la siguiente URL

- https://zenodo.org/record/7225553

Daniel Solá. (2022). Michael Jackson Singles & EPs [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7225553

11. Vídeo

El vídeo de la práctica se encuentra disponible en el siguiente enlace:

https://drive.google.com/file/d/1uOM0gk0DVFVNfhHp9EhZ2IGGqJxKsBzK/view?usp=share_link