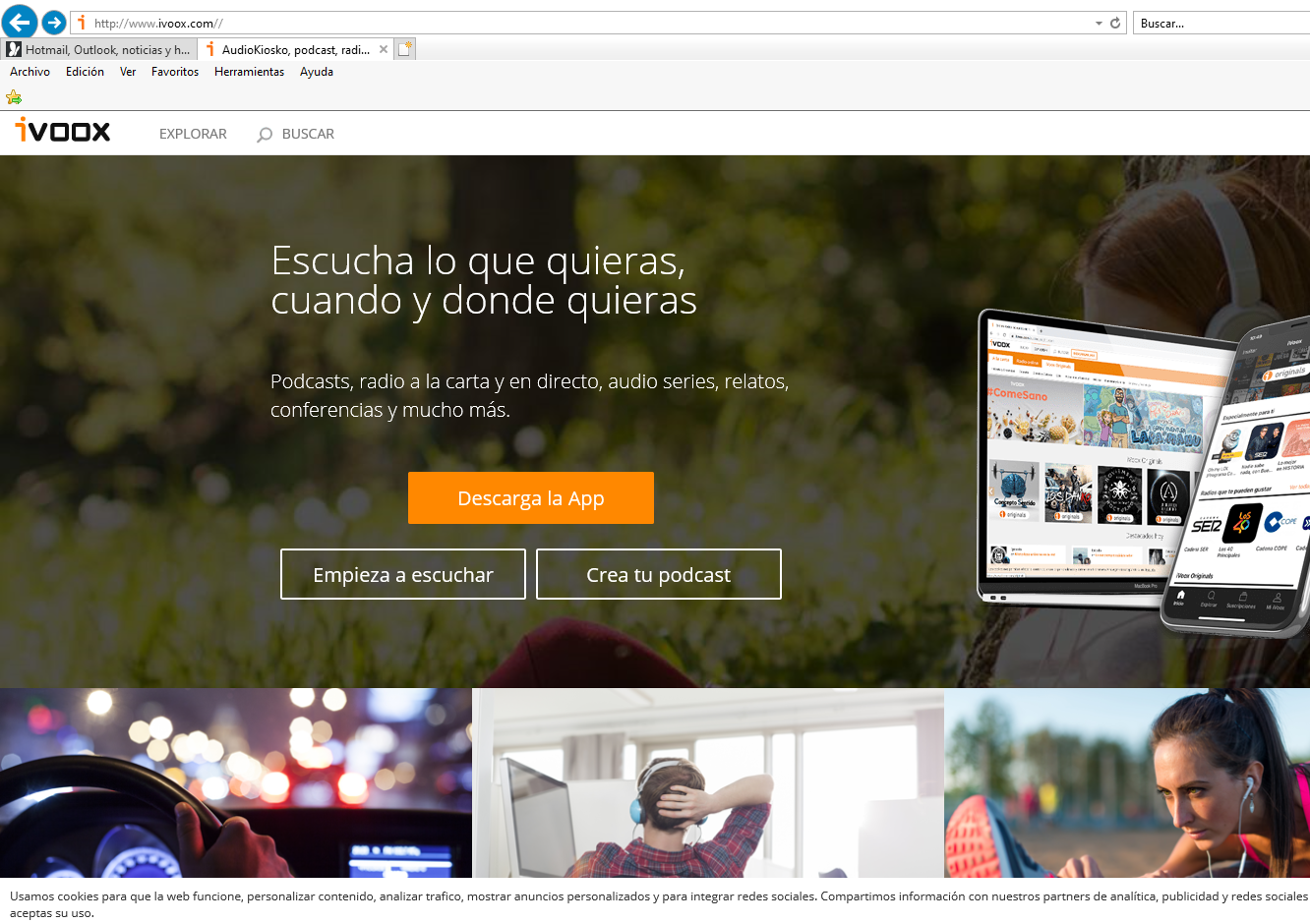
**PRÁCTICA 1**

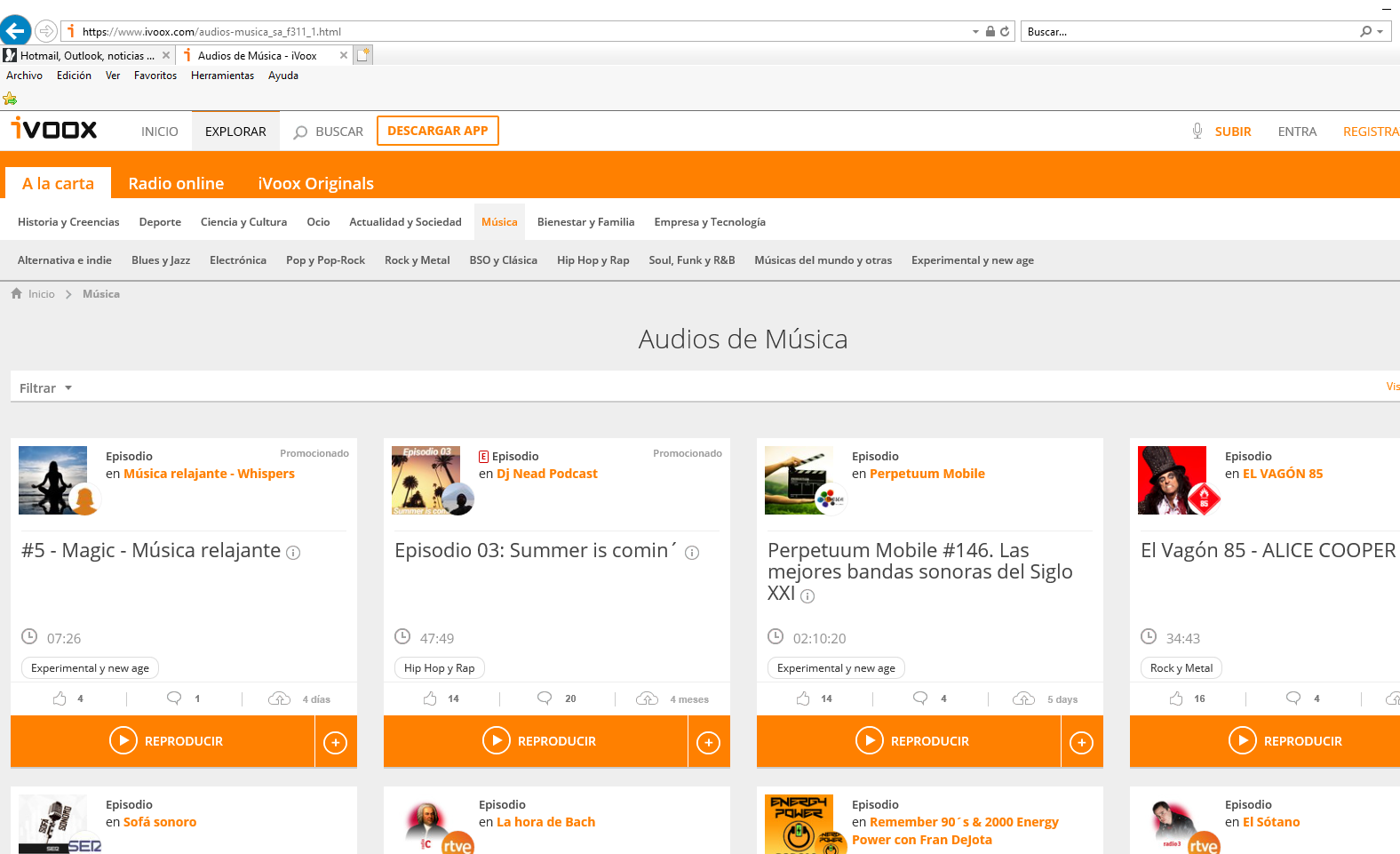
# Descripción de la PRÁCTICA

## El objetivo de esta actividad será la creación de un dataset a partir de los datos contenidos en una web. Para su realización, se deben cumplir los siguientes puntos:

## 1. Contexto. Explicar en qué contexto se ha recolectado la información. Explique por qué el sitio web elegido proporciona dicha información.

El objetivo de eta actividad es la creación de un dataset a partir de los contenidos en la web <http://www.ivoox.com/> en concreto al ser una web de Podcasts, radio a la carta y en directo, audio series, relatos, conferencias y mucho más he hecho un filtro y he comenzado desde la opción de **Empieza a escuchar**, es decir , y una vez allí he filtrado por música.





El

## 2. Definir un título para el dataset. Elegir un título que sea descriptivo.

Dataset: “Música a la carta desde ivoox”

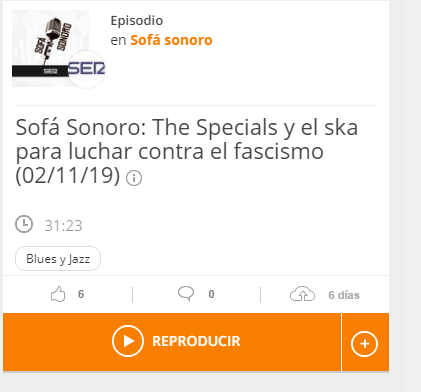
Nombre del fichero de CSV: **salida\_muusica\_tipo\_a**

En Python: **Marco**

## 3. Descripción del dataset. Desarrollar una descripción breve del conjunto de datos que se ha extraído (es necesario que esta descripción tenga sentido con el título elegido).

“Música a la carta desde ivoox” como se basa en marcos o cuadros, en los que elige en la web, he llamado a si al dataset en el programa.

Marco



El conjunto de los datos que se extrae en el caso mi caso esta filtrado previamente por la elección del tipo de música.



Es un filtro que podemos elegir en la web en un menú a tercera línea de página que implementado en Python y es usando la para la ejecución la opción un parámetro –tipo X, donde X puede ser:

**"all"**:**"Todo tipo de múscia"**,  
**"a"** :**"Alternativa e indie"**,  
**"b"** :**"Blues y Jazz"**,  
**"e"** :**"Electrónica"**,  
**"p"** :**"Pop y Pop-Rock"**,  
**"r"** :**"Rock y Metal"**,  
**"bs"** :**"BSO y Clásica"**,  
**"h"** :**"Hip Hop y Rap"**,  
**"s"** :**"Soul, Funk y R&B"**,  
**"m"** :**"Músicas del mundo y otras"**,  
**"ex"**:**"Experimental y new age"**}

Existen ayuda en el programa de Python para ver estoy también en caso de error lo indica

**Ejemplos:**

Python muusica-vox.py –tipo a :**"Alternativa e indie"**

Python muusica-vox.py –tipo b :**"Blues y Jazz"**

Python muusica-vox.py –tipo all: **TOda**

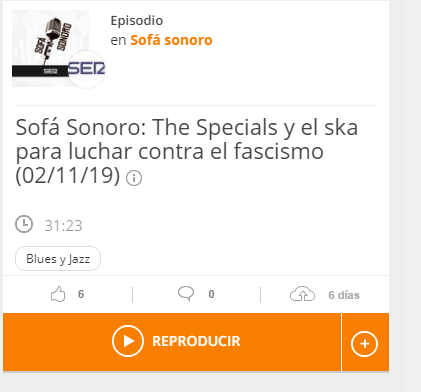
Mejoras

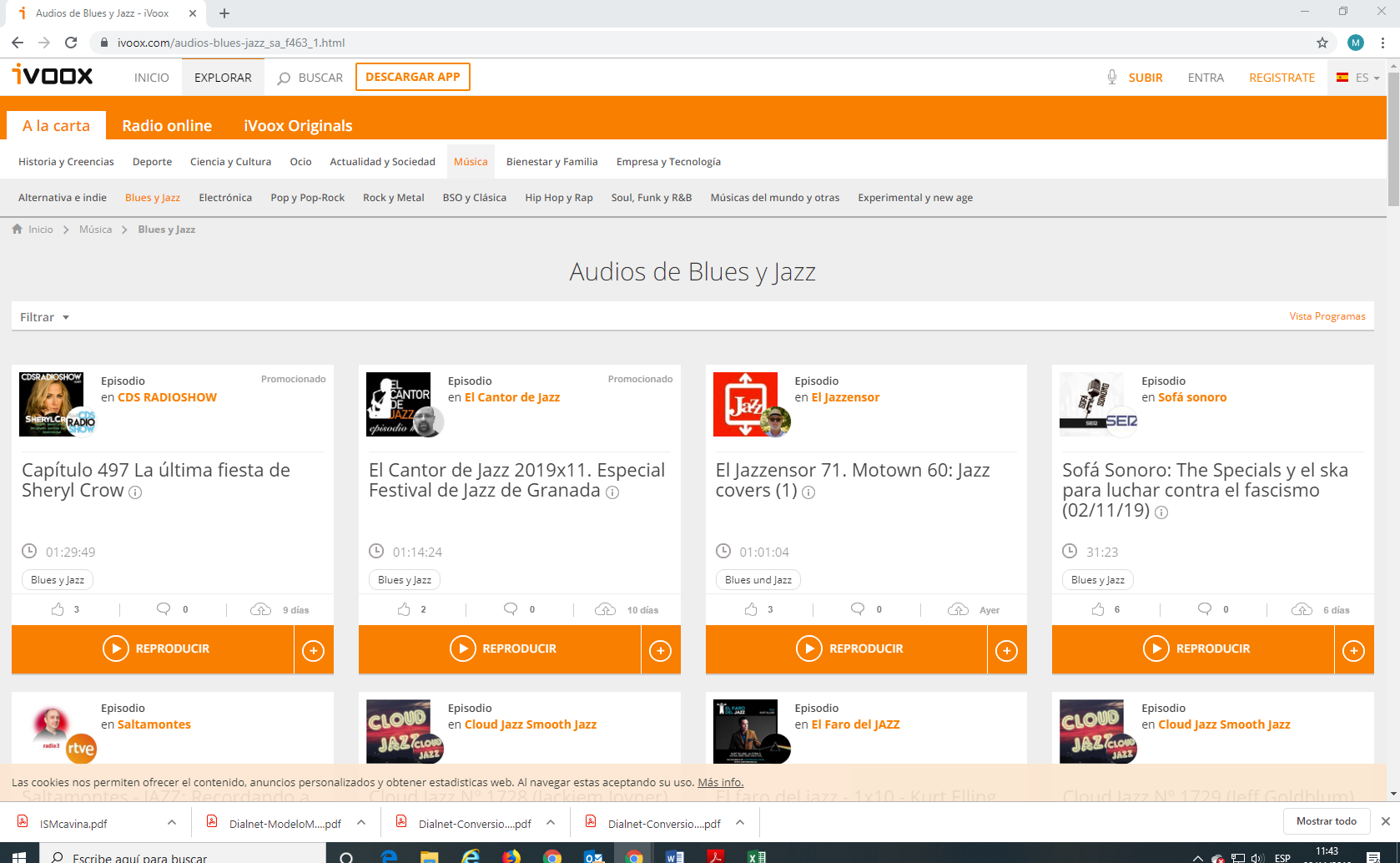
Visto lo que hecho modificando muy poco el programa serviría para todas las opciones no solo la música.

Pero es un problema por el diseño de ivoox, si en vez de empezar en música, empiezo en el menú también funcionaria, pero hay muchas URL



## 4. Representación gráfica. Presentar una imagen o esquema que identifique el dataset visualmente





## 5. Contenido. Explicar los campos que incluye el dataset, el periodo de tiempo de los datos y cómo se ha recogido.

Los campos son:

|  |  |  |
| --- | --- | --- |
| Caratula | https://static-2.ivoox.com/canales/7/1/4/0/8021554710417\_SM.jpg |  |
| Programa | Sofá sonoro |  |
| Emisión | Sofa Sonoro: The Specials y el ska para luchar contra el fascismo (02/11/19) |  |
| Tiempo | 31:23:00 |  |
| Tipo\_musica | Blues y Jazz |  |
| Likes | 6 |  |
| Comentarios | 0 |  |
| Antiguedad | 6 |  |
| Reproducir | https://www.ivoox.com/sofa-sonoro-the-specials-ska-para-audios-mp3\_rf\_43827223\_1.html?autoplay=true |  |

Los campos son:

|  |  |  |
| --- | --- | --- |
| Caratula | URL | Url de la caratula de programa |
| Programa | Carácter | Nombre del programa |
| Emisión | Carácter | Numero de Emision del programa anterior |
| Tiempo | Fecha/Hora | Duracion tiempo |
| Tipo\_musica | Carácter | Tipo de música elegida por mi |
| Likes | Numérico | Gente que hadado me gusta |
| Comentarios | Numérico | Numero de comentarios que hay |
| Antiguedad | Carácter | Días que llega el programa emitido |
| Reproducir | URL | Url de la MP3 de programa |

## 6. Agradecimientos. Presentar al propietario del conjunto de datos. Es necesario incluir citas de investigación o análisis anteriores (si los hay).

No existe agradecimientos al propietario, pues no pude ponerme en contacto con él.

Por otra parte, los datos de investigación cambian día, casi, pues al ser activa la web o datos inyectados a diario el fichero de Excel que se genere cambia.

## 7. Inspiración. Explique por qué es interesante este conjunto de datos y qué preguntas se pretenden responder.

El motivo de elegir este sitio es personal para poder saber qué tipo de música existe en este sitio y así poder más información cuando la uso, y elegir con más comodidad.

También es porque en la página web se puede escuchar Podcasts, radio a la carta y audio de series, relatos, conferencias, y música, me gusta poder elegir programas de radio o series que por su horario no puedo oír, y escuchar más cuando voy en el coche o en el tren o estoy en cas en la terraza.

## 8. Licencia. Seleccione una de estas licencias para su dataset y explique el motivo de su selección:

## ○ Released Under CC0: Public Domain License

## ○ Released Under CC BY-NC-SA 4.0 License

## ○ Released Under CC BY-SA 4.0 License

## ○ Database released under Open Database License, individual contents under Database Contents License

## ○ Other (specified above)

## ○ Unknown License

NO he usado ningún tipo de licencia o más bien seria Unknown License, no he visto ninguno que pudiera usar

## 9. Código. Adjuntar el código con el que se ha generado el dataset, preferiblemente en Python o, alternativamente, en R.

**import** requests  
**import** lxml  
**import** bs4  
**import** argparse  
**import** os  
**import** csv  
**import** sys  
**import** pathlib  
  
  
  
tipos\_muusica={  
**"all"**:**"Todo tipo de múscia"**,  
**"a"** :**"Alternativa e indie"**,  
**"b"** :**"Blues y Jazz"**,  
**"e"** :**"Electrónica"**,  
**"p"** :**"Pop y Pop-Rock"**,  
**"r"** :**"Rock y Metal"**,  
**"bs"** :**"BSO y Clásica"**,  
**"h"** :**"Hip Hop y Rap"**,  
**"s"** :**"Soul, Funk y R&B"**,  
**"m"** :**"Músicas del mundo y otras"**,  
**"ex"**:**"Experimental y new age"**}  
  
tipos\_url\_base={  
**"all"**:**"https://www.ivoox.com/audios-musica\_sa\_f311\_1.html"**,  
**"a"**:**"https://www.ivoox.com/audios-alternativa-e-indie\_sa\_f462\_1.html"**,  
**"b"** :**"https://www.ivoox.com/audios-blues-jazz\_sa\_f463\_1.html"**,  
**"e"** :**"https://www.ivoox.com/audios-electronica\_sa\_f464\_1.html"**,  
**"p"** : **"https://www.ivoox.com/audios-pop-pop-rock\_sa\_f465\_1.html"**,  
**"r"** : **"https://www.ivoox.com/audios-rock-metal\_sa\_f466\_1.html"**,  
**"bs"**: **"https://www.ivoox.com/audios-bso-clasica\_sa\_f467\_1.html"**,  
**"h"** :**"https://www.ivoox.com/audios-hip-hop-rap\_sa\_f468\_1.html"**,  
**"s"** :**"https://www.ivoox.com/audios-soul-funk-r-b\_sa\_f470\_1.html"**,  
**"m"** :**"https://www.ivoox.com/audios-musicas-del-mundo-otras\_sa\_f471\_1.html"**,  
**"ex"**:**"https://www.ivoox.com/audios-experimental-new-age\_sa\_f472\_1.html"**}  
  
  
*#Parse command line arguments*parser = argparse.ArgumentParser()  
parser.add\_argument(**"--tipo"**, help=**'''Elige el tipo de música:  
"a":"Alternativa e indie",  
"b":"Blues y Jazz",  
"e":"Electrónica",  
"p":"Pop y Pop-Rock",  
"r":"Rock y Metal",  
"bs":"BSO y Clásica",  
"h":"Hip Hop y Rap",  
"s":"Soul, Funk y R&B",  
"m":"Músicas del mundo y otras",  
"ex":"Experimental y new age"  
  
 '''**)  
args = parser.parse\_args()  
  
t=args.tipo  
  
**if not** t:  
 **print**(**"Debes elegir tipo de música: ejemplo: >python muusica-ivox.py --tipo a"**)  
 exit()  
  
**if** t **in** tipos\_muusica:  
 **print**(**"Has elegido: {}"**.format(tipos\_muusica[t]))  
**else**:  
 **print**(**"La opción elegeda no es correcta"**)  
 exit()  
  
  
  
  
  
*#función para generar las url correspondientes con la paginación del sitio web:***def** genera\_paag(url, j):  
 partes=url.split(**'\_'**)  
 s=**""  
 for** i **in** range(len(partes)-1):  
 s=s+partes[i]+**"\_"** s=s+str(j)+**".html"  
 return** s  
  
  
*#FUNCIONES DE OBTENCÍON DEL DATASET A PARTIR DEL DIV CONTENDDOR, LLAMADO 'marco'***def** obtener\_caraatula(marco):*#cada marco tiene 2 imágenes, y la carátula es la primera* imgs=marco.select(**'img'**) *#obtiene 2 imágenes* im=imgs[0] *#guarda la primera* **return**(im[**'src'**]) *#saca el vínculo***def** obtener\_programa (marco):*#es un tag 'a' dentro del div con class='wrapper'* divs=marco.select(**'div'**)  
 **for** d **in** divs:  
 **if** d.has\_attr(**'class'**) **and** d[**'class'**][0]==**'wrapper'**:  
 a=d.select(**'a'**)  
 tag=a[0]  
 **return**(tag[**'title'**])  
  
**def** obtener\_emisioon(marco):*#está dentro de un 'p' con class="title-wrapper text-ellipsis-multiple"* ps=marco.select(**'p'**)  
 **for** p **in** ps:  
 **if** p.has\_attr(**'class'**):  
 c=p[**'class'**]  
 **if** len(c)==2 **and 'title-wrapper' in** c **and 'text-ellipsis-multiple' in** c:  
 aas=p.select(**'a'**)  
 tag=aas[0]  
 **return** (tag[**'title'**])  
  
**def** obtener\_tiempo(marco):*# p.time* ps=marco.select(**'p'**)  
 **for** p **in** ps:  
 **if** p.has\_attr(**'class'**) **and 'time' in** p[**'class'**]:  
 **return** (p.getText())  
  
**def** obtener\_tipo\_muusica(marco):*# busco a.rounded-label* aa=marco.select(**'a'**)  
 **for** a **in** aa:  
 **if** a.has\_attr(**'class'**) **and 'rounded-label' in** a[**'class'**]:  
 **return** (a.getText()).strip()  
  
**def** obtener\_likes(marco):*# busco li.likes y dentro un a con 'title'="r'\d' Likes"* lis=marco.select(**'li'**)  
 **for** li **in** lis:  
 **if** li.has\_attr(**'class'**) **and 'likes' in** li[**'class'**]:  
 texto=li.a[**'title'**]  
 texto=texto.split(**" "**)[0]  
 **return** texto  
  
  
**def** obtener\_comentarios(marco): *# busco li.comments y dentro un a con 'title'="r'\d' Comentarios"* lis=marco.select(**'li'**)  
 **for** li **in** lis:  
 **if** li.has\_attr(**'class'**) **and 'comments' in** li[**'class'**]:  
 texto=li.a[**'title'**]  
 texto=texto.split(**" "**)[0]  
 **return** texto  
  
**def** obtener\_diias(marco):*#li.date getText()* lis=marco.select(**'li'**)  
 **for** li **in** lis:  
 **if** li.has\_attr(**'class'**) **and 'date' in** li[**'class'**]:  
 texto=li.getText()  
 texto=texto.strip()  
 **return** texto  
  
**def** obtener\_reproducir(marco):*# busco div.play , dentro el 'a' con rel="nofollow" y onclick="...."* divs=marco.select(**'div'**)  
 **for** div **in** divs:  
 **if** div.has\_attr(**"class"**) **and 'play' in** div[**"class"**]:  
 aas=div.select(**'a'**)  
 **for** a **in** aas:  
 **if** a.has\_attr(**"rel"**) **and** a.has\_attr(**"onclick"**) **and 'nofollow' in** a[**"rel"**]:  
 link=a[**"onclick"**]  
 link=link.split(**'"'**)[1]  
 **return** link  
  
  
url\_base=tipos\_url\_base[t]  
  
  
lista=[]  
encabezado= [**'Carátula'**, **'Programa'**, **'Emisión'**, **'Tiempo'**, **'Tipo de música'**, **'Likes'**, **'Comentarios'**,**'Antigüedad'**, **'Reproducir'**]  
lista.append(encabezado)  
  
  
npags=5 *#se recomienda poner 20 como máximo***for** i **in** range(npags):  
 url=genera\_paag(url\_base,1+i)  
  
 *#proceso la página* res=requests.get(url)  
 soup=bs4.BeautifulSoup(res.text,**'lxml'**)  
 div=soup.select(**'div'**)  
  
 marcos=[]  
 *#filtro los div que tienen class="front modulo-view modulo-type-episodio"* **for** d **in** div:  
 **if** d.has\_attr(**'class'**):  
 c= (d[**'class'**])  
 **if** len(c)==3 **and 'front' in** c **and 'modulo-view' in** c **and 'modulo-type-episodio' in** c:  
 marcos.append(d)  
  
  
  
  
 *#por cada marco saco una fila del dataset* **for** m **in** marcos:  
 fila=[]  
 fila.append(obtener\_caraatula(m))  
 fila.append(obtener\_programa(m))  
 fila.append(obtener\_emisioon(m))  
 fila.append(obtener\_tiempo(m))  
 fila.append(obtener\_tipo\_muusica(m))  
 fila.append(obtener\_likes(m))  
 fila.append(obtener\_comentarios(m))  
 fila.append(obtener\_diias(m))  
 fila.append(obtener\_reproducir(m))  
  
 lista.append(fila)  
  
 **print**(**"Procesada la página {} de {}... "**.format(1+i,npags))  
  
  
  
  
  
*#escritura*currentDir = pathlib.Path(\_\_file\_\_).parent  
filename = **" "**filePath = os.path.join(currentDir, filename)  
  
**with** open(filePath,**'w'**,newline=**''**) **as** csvFile:  
 writer=csv.writer(csvFile)  
 **for** l **in** lista:  
 writer.writerow(l)  
  
**print**(**"Se ha creado el fichero {} en la ruta: {}"**.format(filename,filePath))

## 10. Dataset. Presentar el dataset en formato CSV

Preento dos fcheros **csv** para que se vean 2 casos.

salida\_muusica\_tipo\_a.csv

salida\_muusica\_tipo\_b.csv

11.- Nombre de los componentes del grupo

Maria José Morte Ruiz

[mjmorteruiz@uoc.edu](mailto:mjmorteruiz@uoc.edu)