

Data-mining de Bibliotecas Públicas con ChatGPT



Pedro Lázaro Rodríguez

[ConocimIA](#) -15 de diciembre de 2023

Departamento de Biblioteconomía y Documentación (UCM)



Índice

Introducción.....	3
1. Necesidad.....	8
2. Antecedentes.....	18
3. Diseñando PyDataBibPub con ChatGPT 3.5.....	23
4. Límites y posibilidades de PyDataBibPub.....	43
5. Reflexiones sobre el uso de ChatGPT.....	46

Introducción

Sobre mi investigación:

- Análisis, evaluación y desarrollo de herramientas y métodos para la mejora de bibliotecas

Introducción

- PyDataBibPub: *script* en Python para descargar datos de bibliotecas públicas de España
 - ¿Qué es Python?
 - ¿Qué es un *script*?

Introducción: ¿Qué es Python?

“Si trabajas mucho con ordenadores, en algún momento encontrarás que hay alguna tarea que quieres automatizar...”

“Python es justo el lenguaje para ti”

Fuente: docs.python.org

Introducción: ¿Qué es Python?

- Lenguaje de programación
 - Automatizar la ejecución de instrucciones y procesos para alcanzar unos resultados
- Multiparadigma: integrador y dinámico
 - módulos (libraries)
- Multiplataforma: Linux, Windows, Mac, etc.

Fuente: docs.python.org

Introducción: ¿Qué es un *script*?

- En Python: `archivo.py`
- Conjunto de órdenes, funciones, instrucciones, etc., para la automatización

1. Necesidad

- Automatizar la descarga de datos de bibliotecas públicas de España
 - La web [Bibliotecas públicas españolas en cifras](#)
 - La web [Estadística de Bibliotecas. Resultados](#) de CulturaBASE
- Limitaciones

1. Necesidad: la web BPEC

GOBIERNO DE ESPAÑA MINISTERIO DE CULTURA U23

Bibliotecas públicas españolas en cifras

Portada

Presentación

Mapa

Anexos

Evolución

Indicadores de rendimiento

Responsables

Contacte

Bibliotecas públicas españolas en cifras 2021

Destacados

Últimos datos publicados: 2022

Dirección y contacto

Subdirección General de Coordinación Bibliotecaria
Plaza del Rey, 1
28071 Madrid
Tel.: 91 701 71 40
[Acceso al formulario de contacto](#)

Sistema de Información sobre bibliotecas públicas españolas gestionado por las Comunidades Autónomas y el Ministerio de Cultura y Deporte de acuerdo a la norma ISO 2789 information and documentation: international library statistics.

Captura de la web [Bibliotecas públicas españolas en cifras](#)

1. Necesidad: la web BPEC

Limitaciones:

1. Consulta de datos por categorías y no por redes

- Para obtener todos los datos de una red, en el mejor de los casos, habrían de hacerse 52 consultas
- En el mejor de los casos → por comunidades autónomas

1. Necesidad: la web BPEC

Limitaciones:

2. Nivel de los datos

- Para ver municipios, hay que acceder a cada provincia; para acceder a cada provincia, a cada comunidad autónoma
- Para descargar los datos de 1 variable a nivel de municipios:

$1 * 17 * 52 = 884$ clics (+ las descargas + volver atrás)

Todo esto para 1 variable y un solo año

1. Necesidad: la web BPEC

¿Por qué disponer de datos a nivel de municipios?

Municipio	Personal (ETC)	Gasto (en personal)	Gasto por persona trabajadora (ETC)
Tías (Las Palmas)	4	¿?	
Cocentaina (Alicante)	3	¿?	
Donostia (Gipuzkoa)	9	¿?	¿?
Hondarribia (Gipuzkoa)	2	¿?	
Elda (Alicante)	11	¿?	
Lemoiz (Bizkaia)	5	¿?	

Fuente: [Lázaro-Rodríguez y Aguilar-Soto \(2023\)](#)

1. Necesidad: la web BPEC

¿Por qué disponer de datos a nivel de municipios?

Municipio	Personal (ETC)	Gasto (en personal)	Gasto por persona trabajadora (ETC)
Tías (Las Palmas)*	4	8.498.373	2.124.593
Cocentaina (Alicante)	3	1.222.573	407.524
Donostia (Gipuzkoa)	9	3.028.349	336.483
Hondarribia (Gipuzkoa)	2	326.025	163.013
Elda (Alicante)	11	0	0
Lemoiz (Bizkaia)	5	0	0

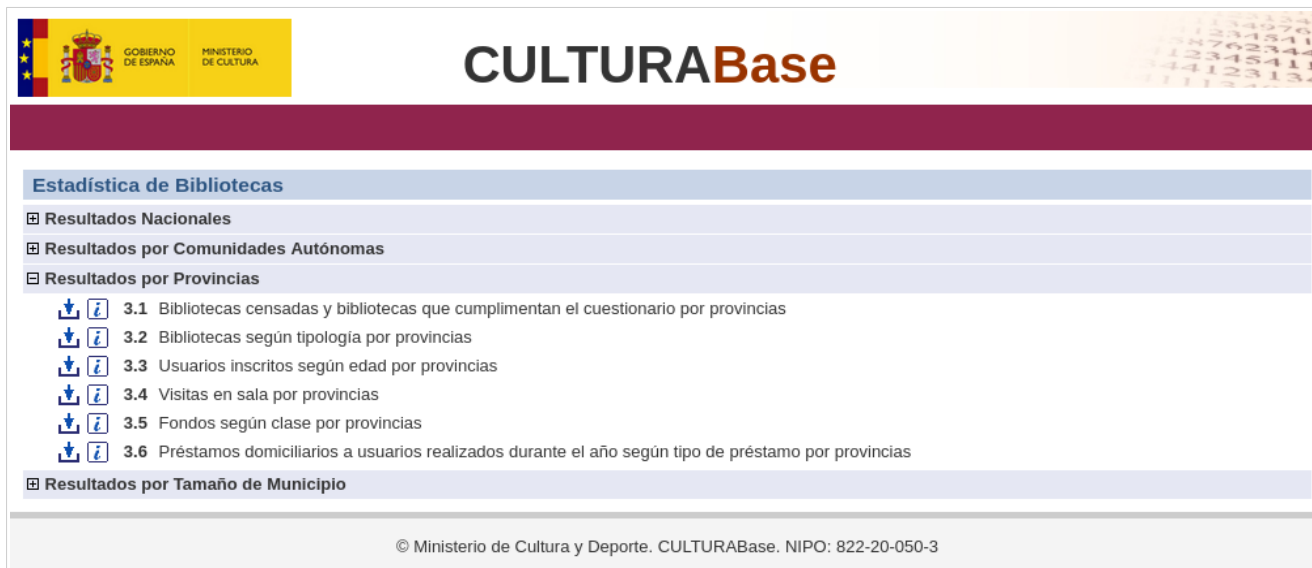
Fuente: [Lázaro-Rodríguez y Aguilar-Soto \(2023\)](#)

1. Necesidad: la web BPEC

¿Por qué disponer de datos a nivel de municipios?

- Mayor nivel de concreción
 - Resultados/interpretación/conclusiones más ajustadas a la realidad → más correctas, completas y válidas
- Identificar errores e inconsistencias en los datos que no se *perciben* en las medias por provincias y comunidades autónomas

1. Necesidad – la web de CulturaBASE



The screenshot shows the header of the CulturaBase website. On the left is the logo of the Spanish Government and the Ministry of Culture. The main title 'CULTURABase' is displayed in a large, bold font. Below the header is a dark red horizontal bar. The main content area is titled 'Estadística de Bibliotecas' and contains a list of menu items and data links. The menu items are: 'Resultados Nacionales', 'Resultados por Comunidades Autónomas', and 'Resultados por Provincias'. Under 'Resultados por Provincias', there are six numbered items (3.1 to 3.6) each with a download icon and an information icon. The items are: 3.1 Bibliotecas censadas y bibliotecas que cumplimentan el cuestionario por provincias; 3.2 Bibliotecas según tipología por provincias; 3.3 Usuarios inscritos según edad por provincias; 3.4 Visitas en sala por provincias; 3.5 Fondos según clase por provincias; 3.6 Préstamos domiciliarios a usuarios realizados durante el año según tipo de préstamo por provincias. Below the menu items is another menu item: 'Resultados por Tamaño de Municipio'. At the bottom of the page, there is a copyright notice: '© Ministerio de Cultura y Deporte. CULTURABase. NIPO: 822-20-050-3'.

GOBIERNO DE ESPAÑA **MINISTERIO DE CULTURA**

CULTURABase

Estadística de Bibliotecas

- ☒ Resultados Nacionales
- ☒ Resultados por Comunidades Autónomas
- ☒ Resultados por Provincias
 - ↓ ⓘ 3.1 Bibliotecas censadas y bibliotecas que cumplimentan el cuestionario por provincias
 - ↓ ⓘ 3.2 Bibliotecas según tipología por provincias
 - ↓ ⓘ 3.3 Usuarios inscritos según edad por provincias
 - ↓ ⓘ 3.4 Visitas en sala por provincias
 - ↓ ⓘ 3.5 Fondos según clase por provincias
 - ↓ ⓘ 3.6 Préstamos domiciliarios a usuarios realizados durante el año según tipo de préstamo por provincias
- ☒ Resultados por Tamaño de Municipio

© Ministerio de Cultura y Deporte. CULTURABase. NIPO: 822-20-050-3

Captura de la web [Estadística de Bibliotecas. Resultados](#) de CulturaBASE

1. Necesidad – la web de CulturaBASE

Limitaciones:

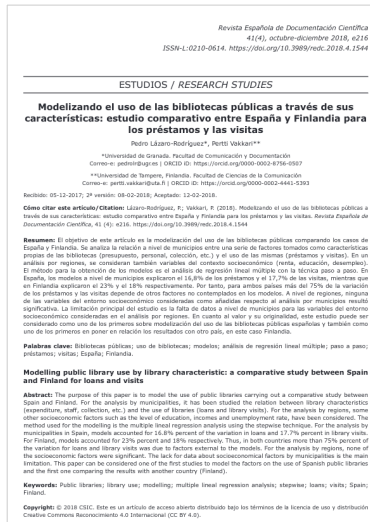
- A mayor nivel de concreción, menos datos disponibles
 - Por provincias, solo 6 variables
- No se pueden consultar datos por municipios
 - Solo por tamaño de municipios
 - Nota crítica en relación a la [Ley 7/1985, Artículo 26](#) y los intervalos del tamaño de municipios

1. Necesidad

- Automatizar la descarga de datos de bibliotecas públicas de España

2. Antecedentes

Descarga manual de todos los municipios (un año y dos variables):



Fuente: [Lázaro-Rodríguez y Vakkari \(2018\)](#)

2. Antecedentes

- Con **wget** en Linux
 - Análisis de la *url* de un caso:

```
http://www.cultura.gob.es/alziraweb/alziraweb.cmd?  
command=GetAnexo&origen=PR&codigo=04&id=51&ejercicio  
=2019&excel=true
```

→

```
codigo=[del 01 al 52]&id=[variable a descargar]&ejercicio=[año]
```

Fuente: [Lázaro-Rodríguez \(2022\)](#)

2. Antecedentes

```
wget -O nombredelarchivo.ods -i nombredelarchivo.txt
```

Creaba una hoja de cálculo *ods* con todos los datos descargados y añadidos como *urls* en el documento *txt*

Recurso: un ejemplo [en vídeo](#)

2. Antecedentes

De wget a Python gracias a ChatGPT 3.5

→

De wget a PyDataBibPub

→

3. Diseñando PyDataBibPub con ChatGPT 3.5



You

Hola me puedes ayudar con un script?



ChatGPT

iHola! Claro, estaré encantado de ayudarte con tu script. ¿En qué lenguaje de programación estás trabajando y cuál es el problema o la tarea específica que necesitas abordar?

Capturas: conversando con ChatGPT 3.5

3. Diseñando PyDataBibPub con ChatGPT 3.5



ChatGPT

Parece que tienes un script de shell (Bash) que descarga datos de una serie de URL con diferentes códigos y los agrega a un archivo de salida. ¿En qué puedo ayudarte con este script? ¿Tienes alguna pregunta específica o algún problema que necesitas resolver?

Capturas: conversando con ChatGPT 3.5

3. Diseñando PyDataBibPub con ChatGPT 3.5

 **You** Sin mucha idea de Python y descargar solo lo que sean tablas?

 **ChatGPT** Si deseas descargar solo las tablas de las páginas web y eliminar el resto del contenido, necesitarás una herramienta más avanzada que Bash, ya que Bash por sí solo no es adecuado para analizar y procesar el contenido HTML. Puedes usar Python junto con una biblioteca como BeautifulSoup para analizar las páginas y extraer tablas.

Aquí hay un ejemplo básico de cómo podrías hacerlo usando Python:

1. Asegúrate de tener Python instalado en tu sistema.
2. Instala la biblioteca BeautifulSoup si aún no la tienes instalada:

```
Copy code  
pip install beautifulsoup4
```

1. Crea un script Python para descargar y analizar las páginas web. Aquí tienes un ejemplo:

Capturas: conversando con ChatGPT 3.5

3. Diseñando PyDataBibPub con ChatGPT 3.5

```
python Copy code

import requests
from bs4 import BeautifulSoup

# URL base y rango de códigos
url_base = "http://www.cultura.gob.es/alziraweb/alziraweb."
inicio_codigo = 1
fin_codigo = 52

# Carpeta donde se guardarán los archivos
carpeta_salida = "archivos_tablas/"

# Itera a través del rango de valores de "codigo" y descarga
for codigo in range(inicio_codigo, fin_codigo + 1):
    url = url_base.format(codigo)
    response = requests.get(url)

    if response.status_code == 200:
        # Analiza la página web con BeautifulSoup
        soup = BeautifulSoup(response.text, 'html.parser')

        # Encuentra todas las tablas en la página
        tablas = soup.find_all('table')

        if tablas:
            # Crea un archivo HTML con la tabla y guárdalo
            with open(f"{carpeta_salida}tabla_{codigo}.html", "w") as archivo_salida:
                archivo_salida.write(str(tablas[0]))
        else:
            print(f"No se pudo descargar la página para el código {codigo}")

print("Descarga de tablas completada.")
```

Sin mucha idea de Python

Este script descargará las páginas web, buscará tablas en ellas y guardará cada tabla en un archivo HTML separado en la carpeta especificada. Puedes ajustar la carpeta de salida y otros detalles según tus necesidades.

Asegúrate de que tengas permisos de escritura en la carpeta de salida y que la biblioteca BeautifulSoup esté instalada antes de ejecutar el script.



You

Esto se puede instalar en debian?

Capturas: conversando con ChatGPT 3.5

3. Diseñando PyDataBibPub con ChatGPT 3.5



You

me da este error:externally-managed-environment

Sin mucha idea de
Python



You

no puedo instalar pip3 install beautifulsoup4



You

Cómo ejecuto el script en python?



You

Y se puede hacer para que las tablas vayan a un único documento?

Capturas: conversando con ChatGPT 3.5

3. Diseñando PyDataBibPub con ChatGPT 3.5



You

Y para que el archivo resultante sea en formato csv o ods



ChatGPT

Para que el archivo resultante sea en formato CSV o ODS (OpenDocument Spreadsheet), puedes utilizar las bibliotecas adecuadas para manejar estos formatos en Python. A continuación, te mostraré cómo modificar el script para guardar las tablas en estos formatos.

Para guardar las tablas en formato CSV:

Asegúrate de tener la biblioteca `pandas` instalada para facilitar la conversión de tablas en formato CSV. Puedes instalarlo con el siguiente comando:

```
bash
```

Copy code

```
pip3 install pandas
```

Luego, modifica el script de la siguiente manera:



You

Perdón, solo en formato csv

Errores míos

Capturas: conversando con ChatGPT 3.5

3. Diseñando PyDataBibPub con ChatGPT 3.5

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

# URL base y rango de códigos
url_base = "http://www.cultura.gob.es/alziraweb/
alziraweb.cmd?command=GetAnexo&id=31&origen=PR&
codigo={}&ejercicio=2021"
inicio_codigo = 1
fin_codigo = 52
```

Capturas: conversando con ChatGPT 3.5

Resolviendo casos
concretos



You

hay un problema el 1 sale en la web como 01



ChatGPT

Entiendo que el problema es que los códigos en la web están formateados como "01" en lugar de "1". Para solucionar esto y asegurarte de que el script funcione correctamente, puedes modificar la parte del script que se encarga de formatear el código para incluir un "0" adicional si el número es menor que 10. Aquí está el código corregido:

3. Diseñando PyDataBibPub con ChatGPT 3.5

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

# URL base y rango de códigos
url_base = "http://www.cultura.gob.es/alziraweb/alziraweb.cmd?command=GetAnexo&id=31&origen=PR&
codigo={024}6ejercicio=2021"
inicio_codigo = 1
fin_codigo = 52

# Nombre del archivo de salida CSV
archivo_salida_csv = "tablas combinadas.csv"

# Inicializa una lista de DataFrames para almacenar las tablas y los captions
tablas = []
captions = []

# Itera a través del rango de valores de "codigo" y descarga las páginas
for codigo in range(inicio_codigo, fin_codigo + 1):
    url = url_base.format(codigo)
    response = requests.get(url)

    if response.status_code == 200:
        # Analiza la página web con BeautifulSoup
        soup = BeautifulSoup(response.text, 'html.parser')

        # Encuentra todas las tablas en la página
        tablas_pagina = soup.find_all('table')

        if tablas_pagina:
            # Encuentra el caption con la clase 'capt_anexosc'
            caption = soup.find('caption', class_='capt_anexosc')
            caption_text = caption.get_text() if caption else ''

            # Convierte la primera tabla encontrada en un DataFrame
            tabla = pd.read_html(str(tablas_pagina[0]))[0]

            # Agrega la tabla y el caption a las listas correspondientes
            tablas.append(tabla)
            captions.append(caption_text)

# Concatena todas las tablas en un solo DataFrame
tabla_final = pd.concat(tablas, ignore_index=True)

# Agrega la columna 'caption' al DataFrame final
tabla_final['caption'] = captions

# Guarda el DataFrame en un archivo CSV
tabla_final.to_csv(archivo_salida_csv, index=False)

print(f"Tablas combinadas con captions guardadas en {archivo_salida_csv}.")
```

Corrigiendo a
ChatGPT 3.5



You
e incluye lo del 01?



ChatGPT
Mis disculpas, olvidé incluir el formateo del código en el rango de valores de "codigo"
para que tenga ceros a la izquierda si es menor que 10. Aquí está el script corregido:

Capturas: conversando con ChatGPT 3.5

3. Diseñando PyDataBibPub con ChatGPT 3.5

Mejoras... ¿Mejoras?...

 You

El caption que quería no es ese. te puedo decir el correcto?

 ChatGPT

¡Claro! Si el "caption" que deseas extraer no es el contenido de la etiqueta `<caption>` de la tabla, por favor, indícame cuál es el elemento HTML específico que contiene la información que deseas antes de cada tabla. Proporciona detalles sobre cómo se identifica ese elemento en la página web para que pueda ayudarte a modificar el script de acuerdo a tus necesidades.

 You

`<caption colspan="4" class="capt_anexosc">`

 ChatGPT

Entendido, quieres extraer el contenido de la etiqueta `<caption>` con la clase `'capt_anexosc'` antes de cada tabla. Puedes hacerlo modificando el script de la siguiente manera:

Capturas: conversando con ChatGPT 3.5

 You

sí, a ese script, me gustaría añadirle el título de cada tabla antes

 ChatGPT

Para agregar el título de cada tabla antes de los datos en el archivo CSV, puedes modificar el script de la siguiente manera:

 You

no, no son h2, los títulos están en el campo `summary` de las tablas


`<caption colspan="4" class="capt_anexosc">Informe anual. 2022. Andalucía - Almería</caption>`

 ChatGPT

Entendido, si los títulos de las tablas están en el campo `'summary'` de las tablas HTML, puedes modificar el script de la siguiente manera para agregar esos títulos al archivo CSV:

 You

Se podría hacer para que la columna variable se divida en dos, con lo que hay antes del punto, y llamarle a esa columna variable, y a lo que hay después del punto y eliminando el espacio, año

 ChatGPT

Claro, puedes modificar el script para dividir la columna 'variable' en dos columnas, una con lo que hay antes del punto y otra con lo que hay después del punto (eliminando el espacio). La primera columna se llamará 'variable' y la segunda se llamará 'año'. Aquí tienes el script modificado:

 You

Me gustaría que la columna título aparezca la primera

3. Diseñando PyDataBibPub con ChatGPT 3.5

Aprendiendo... procesos > resultados



You

codigo={:02d} por qué esto se escribe así, qué hace el {:02d}



ChatGPT

La parte ``{:02d}`` en la cadena de formato se utiliza para formatear el valor de la variable `codigo`. Aquí hay una explicación de lo que significa:

- ``{}``: Esto indica un marcador de posición en la cadena de formato donde se insertará un valor.
- ``:02d``: Esto es una especificación de formato que controla cómo se formatea el valor que se insertará. Desglosemos esto:
 - ``d``: Indica que el valor es un número entero (decimal).
 - ``2``: Indica que el número debe tener al menos 2 dígitos.
 - ``0``: Indica que, si el número tiene menos de 2 dígitos, se deben rellenar los espacios en blanco con ceros.

Capturas: conversando con ChatGPT 3.5

3. Diseñando PyDataBibPub con ChatGPT 3.5

Aprendiendo... procesos > resultados

Entonces, cuando tienes `{:02d}`, significa que el valor de `codigo` se formateará como un número entero con al menos 2 dígitos, y si tiene menos de 2 dígitos, se rellenarán con ceros a la izquierda.

Por ejemplo:

- Si `codigo` es `1`, se formateará como `"01"`.
- Si `codigo` es `12`, se mantendrá como `"12"`.

Esto a menudo se usa en programación para garantizar que los números se formateen de manera uniforme, especialmente cuando se trabajan con números que deben tener una longitud fija o cuando se desea que la salida sea más legible. En este caso, parece que se está utilizando para que `codigo` siempre tenga al menos 2 dígitos en la URL.

Capturas: conversando con ChatGPT 3.5

3. Diseñando PyDataBibPub con ChatGPT 3.5

Problemas de codificación de caracteres... de la web



You

Este script descarga unas tablas que son datos. Estos datos la mayoría son números que están en español, con la coma como decilam y el punto como millares. Se puede tener eso en cuenta a la hora de las descargas de los datos a los csv?




ChatGPT

Sí, puedes tener en cuenta el formato de los números al descargar los datos a los archivos CSV. Para hacerlo, puedes especificar el separador de miles y el separador decimal al guardar los datos en el archivo CSV utilizando el parámetro ``decimal`` y ``thousands`` de la función ``to_csv`` de pandas.

Aquí te muestro cómo puedes hacerlo en el script que proporcionaste:

3. Diseñando PyDataBibPub con ChatGPT 3.5



Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Notes and Potential Issues](#) [Validation Output](#)

Errors found while checking this document as HTML 4.0 Transitional!

Result:	8 Errors, 2 warning(s)	
Address:	<input type="text" value="https://www.mcu.es/alziraweb/alziraweb.cmd?command=GetAnexo&id=51"/>	
Encoding:	iso-8859-1	<input type="text" value="(detect automatically)"/>
Doctype:	HTML 4.0 Transitional	<input type="text" value="(detect automatically)"/>
Root Element:	html	

Notes and Potential Issues

⚠ Character Encoding mismatch!

The character encoding specified in the HTTP header (iso-8859-1) is different from the value in the <meta> element (iso-8859-15). I will use the value from the HTTP header (iso-8859-1) for this validation.

Captura de la validación del HTML del W3C y de la web de los datos

3. Diseñando PyDataBibPub con ChatGPT 3.5

... Hasta aquí:

- El primer *script* válido tenía 41 líneas
- Añadí variabilidad en *ids* (tipos de datos: gastos, préstamos, etc.), en *años* y *origen* (provincia y CCAA)
- Añadí otros matices
- 8 *scripts* válidos
- ...





→








3. Diseñando PyDataBibPub con ChatGPT 3.5

¿Qué hacía el script?

- Descarga datos de los municipios de 52 provincias. Se podrían seleccionar casos concretos
- Los datos son de variables sobre bibliotecas públicas (tipos de datos; hay 52)
- La descarga de datos es para un año o rango de años. Hay datos disponibles desde 2010 a 2021, pero algunas variables los tienen desde 2017
- El script está diseñado para unificar en un CSV cada variable con todos los años seleccionados y códigos (de las provincias con sus municipios). Si hay más de una variable, crea un CSV para cada una. Además, añade columnas con las provincias y las comunidades autónomas
- El script está diseñado para unificar en un CSV cada variable con todos los años seleccionados y códigos (de las provincias con sus municipios). Si hay más de una variable, crea un CSV para cada una. Además, añade columnas con las provincias y las comunidades autónomas
- Tras la creación de estos CSV, genera también un CSV con todas las variables juntas para todos los años y municipios
- A su vez, genera una hoja de cálculo en formato ODS con pestañas para los CSV de cada variable y el que contiene todas las variables
- Entre medias, limpia las filas que en la variable "informe anual" de datos de bibliotecas aparecen con 0 bibliotecas
- A su vez, chequea que las columnas para las variables "Municipio", "Año", "Comunidad Autónoma" y "Provincia" coinciden en todos los casos
- Entre medias, limpia las filas que en la variable "informe anual" de datos de bibliotecas aparecen con 0 bibliotecas. Esto solo ocurre con la variable "informe anual".
- A su vez, chequea que las columnas para las variables "Municipio", "Año", "Comunidad Autónoma" y "Provincia" coinciden en todos los casos (en el CSV con todas las variables)
- La carpeta que crea sigue el nombre de "archivos_ids-de-las-variables-descargadas_años_descargados_descarga-fecha-de-la-descarga-y-hora"
- Los CSV de las variables tienen el nombre de la variable y los años descargados, y al ODS con todo unido, van por pestañas con esos mismos nombres
- El ODS y el CSV con todo unido tienen estos nombres: "csv-o-ods_ids-descargados_años_descargados_descarga-fecha-de-la-descarga-y-hora"

3. Diseñando PyDataBibPub con ChatGPT 3.5

Name
 PROV_archivos_ids-51_años_2010-2021_descarga-13-10-2023_06-23-37
 CCAA_archivos_ids-51_años_2010-2021_descarga-13-10-2023_06-23-32
 archivos_ids-51_años_2010-2021_descarga-17-10-2023_10-40-07
 archivos_ids-38-46-47-50-51_años_2010-2021_descarga-03-10-2023_19-37-45

Name	Size ▲
 ods_ids-38-46-47-50-51_años_2010-2021_descarga-03-10-2023_19-37-45.ods	12,8 MiB
 csv_ids-38-46-47-50-51_años_2010-2021_descarga-03-10-2023_19-37-45.csv	12,2 MiB
 Informe_anual_2010-2021.csv	3,2 MiB
 Gastos_corrientes_2010-2021.csv	2,7 MiB
 Personal_en_equivalente_a_tiempo_completo_2010-2021.csv	2,3 MiB
 Gastos_de_inversión_2010-2021.csv	2,1 MiB
 Actividades_culturales_y_asistentes_2010-2021.csv	1,8 MiB

Captura de un directorio de descarga

3. Diseñando PyDataBibPub con ChatGPT 3.5

- PyDataBibPub V1:
 - 271 líneas de instrucciones para automatizar...
 - requiriendo 8 módulos de Python
 - ¿Es/era necesario tanto?
- A inicios de noviembre, la web cambia su *url* y PyDataBibPub V1 *se rompe* →
- De PyDataBibPub V1 a PyDataBibPub V2
 - ¿PHP? ¿Python? ...

3. Diseñando PyDataBibPub con ChatGPT 3.5

PyDataBibPub V2:

Explicando y ejecutando el *script*

3. Diseñando PyDataBibPub con ChatGPT 3.5

Partes del *script*:

1. Llamada a módulos de Python
2. Definición de campos: *url*, ámbito geográfico, años, regiones y datos a descargar; y selección de diccionarios de nombres
3. Funciones para:
 1. Generar las *urls*
 2. Obtener el nombre de regiones
 3. Descargar y almacenar las tablas según condiciones
4. Creación de un directorio y publicación de los *csv*

3. Diseñando PyDataBibPub con ChatGPT 3.5

PyDataBibPub V2:

Explicando y ejecutando el *script*

4. Límites y posibilidades de PyDataBibPub

Límites:

- *url* base
- Política y [aviso legal](#) de la fuente de datos:

“Para el resto de contenidos la licencia de uso **se limita** a la descarga por parte del usuario de dicho contenido y el uso privado del mismo, siempre que los citados contenidos permanezcan íntegros, quedando su uso, reproducción, distribución, comunicación pública, transformación o cualquier otra actividad similar o análoga, **totalmente prohibidos** salvo que medie expresa autorización del Ministerio de Cultura”

4. Límites y posibilidades de PyDataBibPub

Posibilidades:

- Scripts adicionales:
 - *CA-PR_informeanual.py* y *MUNI_informeanual.py* → limpian y mejoran el csv del informe anual
 - *CA-PR_indibibmuni.py* → [IndiBibPubli](#)
- DataBibMuni → Datos para bibliotecas públicas de España a nivel de municipios 2010-2021

4. Límites y posibilidades de PyDataBibPub

¿Posibilidades?:

- Nuevo sistema de datos de tal forma que no sean necesarias herramientas tipo PyDataBibPub
 - El sistema de datos de [bibliotecas públicas de Finlandia](#) como ejemplo o modelo de buenas prácticas

5. Reflexiones sobre el uso de ChatGPT

- Versión 3.5
- Actitud exploratoria – 10 meses después
- Uso didáctico: resultados, pero también procesos
- ¿Herramienta buena o mala por sí misma?
 - La ética y la moral son categorías de las personas
 - Tecnología buena o mala según el uso que hagamos las personas
- Para posicionarse en el sí o en el no → hay que conocer

FIN



Pedro Lázaro Rodríguez

pedrolaz@ucm.es

[ConocimIA](#) -15 de diciembre de 2023

Departamento de Biblioteconomía y Documentación (UCM)