PyDataBibPub: script en *Python* para automatizar la descarga de datos de bibliotecas públicas de España desarrollado con *ChatGPT 3.5*

PyDataBibPub: a *Python* script written using *ChatGPT 3.5* to automate the downloading of Spanish public libraries data

Pedro Lázaro-Rodríguez

Cómo citar este artículo:

> **Lázaro-Rodríguez, Pedro** (2024). *PyDataBibPub*: script en *Python* para automatizar la descarga de datos de bibliotecas públicas de España desarrollado con *ChatGPT 3.5* [*PyDataBibPub*: a *Python* script written using *ChatGPT 3.5* to automate the downloading of Spanish public libraries data]". *Infonomy*, 2(3) e24042. *https://doi.org/10.3145/infonomy.24.042*

Pedro Lázaro-Rodríguez
https://orcid.org/0000-0002-8756-0507
https://directorioexit.info/ficha6911
Universidad Complutense de Madrid
Facultad de Ciencias de la Documentación
Departamento de Biblioteconomía y Documentación
Grupo de Investigación INFOBISOC
Santísima Trinidad, 37
28010 Madrid, España
pedrolr@ucm.es

Resumen

Se presenta *PyDataBibPub*, script en *Python* para automatizar la descarga de datos de bibliotecas públicas de España. El script surge de la necesidad de agilizar el proceso de consulta y descarga de datos disponibles en la web *Bibliotecas públicas españolas en cifras*. *PyDataBibPub* ha sido implementado utilizando *ChatGPT 3.5* y se da a conocer el script exponiendo sus partes esenciales, su creación y su funcionamiento. *PyDataBibPub* y su código están disponibles en un proyecto en *Codeberg* con licencia *GNU Affero General Public License v3.0*, de tal forma que se puede reutilizar, adaptar y desarrollar implementando mejoras. Por último, se reflexiona sobre posibles futuras líneas de trabajo planteando la instalación de alternativas a *ChatGPT 3.5* que funcionan en local, privadas y que no dependen de grandes empresas comerciales (por ejemplo, *PrivateGPT*), con la finalidad de desarrollar más aplicaciones útiles para la disciplina de la Biblioteconomía y la Documentación.

Palabras clave
PyDataBibPub; Datos; Estadísticas; Bibliotecas; Bibliotecas públicas; España; *Python*; Script; *ChatGPT*; Inteligencia artificial; Inteligencia artificial generativa; *IAG*; Grandes modelos de lenguaje; *LLMs*.

Abstract
This paper presents *PyDataBibPub*, a *Python* script to automate the downloading of Spanish public libraries data available on the webpage *Bibliotecas públicas españolas en cifras*. The script was written using *ChatGPT 3.5* and the paper includes an exposition of its parts and how it works. *PyDataBibPub* and its code are available in a project in *Codeberg* licensed under the *GNU Affero General Public License v3.0*, so that it can be reused, adapted and improved. For the future, it is proposed the installation of alternatives to *ChatGPT 3.5*, such as *PrivateGPT*, which works locally, privately and not depending on commercial companies, with the aim of developing more applications useful for the discipline of Library and Information Science.

Keywords
PyDataBibPub; Data; Statistics; Libraries; Public Libraries; Spain; *Python*; Script; *ChatGPT*; Artificial Intelligence; Generative Artificial Intelligence; GAI; Large Language Models; LLMs.

1. Introducción

El proyecto *ChatGPT Impact* contabilizó en su actualización a inicios de diciembre de 2023 un total de 6.750 documentos científicos sobre *ChatGPT,* incluyendo artículos científicos y otros tipos de documentos como preprints y editoriales (*ChatGPT Impact*, 2024). Tratar de hacer una revisión de la bibliografía de todo lo publicado sería un trabajo en sí mismo más que una sección de otros. Esos números se refieren solo a *ChatGPT*: los resultados serían mayores si las búsquedas incluyesen conceptos referidos a Grandes Modelos de Lenguaje (*LLMs*, por sus siglas en inglés), a la Inteligencia Artificial Generativa (IAG) y a alternativas como *Phind*, *Perplexity*, etc.

En el ámbito de la biblioteconomía y la documentación desde España se pueden destacar algunos trabajos como el de **Lopezosa** (2023a), analizando la experiencia de búsqueda en *Bing Chat*; o el de **Codina** (2023) centrado en buscadores alternativos a *Google* basados en *IAG* como *You.com*, *Perplexity* y de nuevo *Bing Chat*. Ambos autores analizan también *Bard* (**Lopezosa**; **Codina**, 2023). Otros trabajos se centran en los grandes modelos de lenguaje como oportunidad para la profesión bibliotecaria (**Franganillo**, 2023); en las posibilidades de la inteligencia artificial para los procesos editoriales en revistas académicas (**Lopezosa**, 2023b); y en una reflexión sobre el contexto disruptivo en el acceso a la información (**González-Alcaide**, 2024). También hay trabajos que tratan en particular sobre *ChatGPT*, con un manual sobre sus posibles aplicaciones en investigación y educación universitaria (**Torres-Salinas**; **Arroyo-Machado**, 2023) y un compendio o corpus de aplicaciones (**Torres-Salinas**; **Thewall**; **Arroyo-Machado**, 2024). Por último y en el sentido del modelo de *ChatGPT* y la investigación en sí, cabe destacar la revisión de la bibliografía realizada por **Goyanes** y **Lopezosa** (2024) en investigación cualitativa y cuantitativa.

Siguiendo ideas planteadas en **Torres-Salinas** y **Arroyo-Machado** (2023) y **Torres-Salinas, Thelwall** y **Arroyo-Machado** (2024), una de las aplicaciones de *ChatGPT* es la de la programación en distintos lenguajes incluido *Python*. Profundizando en este posible uso, **Adamson** y **Bägerfeldt** (2023) analizan la eficacia de *ChatGPT* en generar código en

Python, hallando diferencias significativas de calidad con las soluciones escritas por personas, pero con similitudes generales en cuanto a precisión y legibilidad. **Diehl** *et al.***,** (2024) evalúan la generación de código en hasta 9 lenguajes, incluido *Python*, concluyendo que *ChatGPT* crea con éxito código en gran parte de los lenguajes analizados. **Wuisang** *et al.* (2023) analizan la capacidad de *ChatGPT* de corregir *bugs* (fallos en el código) en 40 casos de códigos en *Python*, corrigiendo con precisión 30 de esos 40 casos. Por último, **Lladós-Clos** (2024) concluye que trabajar con *ChatGPT* para desarrollar una aplicación con *Python* resulta útil y eficiente, interpretando bastante bien las instrucciones dadas por la persona al otro lado del chat, aunque detecta dificultades en las respuestas de *ChatGPT* a medida que el chat se extiende y el código se va haciendo más extenso y complejo.

En este artículo se presenta el script *PyDataBibPub* para automatizar la descarga de datos de bibliotecas públicas de España, desarrollado en *Python* y generado con la ayuda de *ChatGPT 3.5*. En la siguiente sección se presenta el script en sí, hablando de su necesidad, antecedentes, la razón para crearlo, del cómo se elaboró y qué hace exactamente, de sus posibilidades de futuro y limitaciones, con un último punto de consideraciones legales a tener en cuenta de la fuente de los datos cuya descarga automatiza el script. Todo ello se remata con una tercera sección a modo de consideraciones finales.

En este artículo se presenta el script *PyDataBibPub* para automatizar la descarga de datos de bibliotecas públicas de España, desarrollado en *Python* y generado con la ayuda de *ChatGPT 3.5*

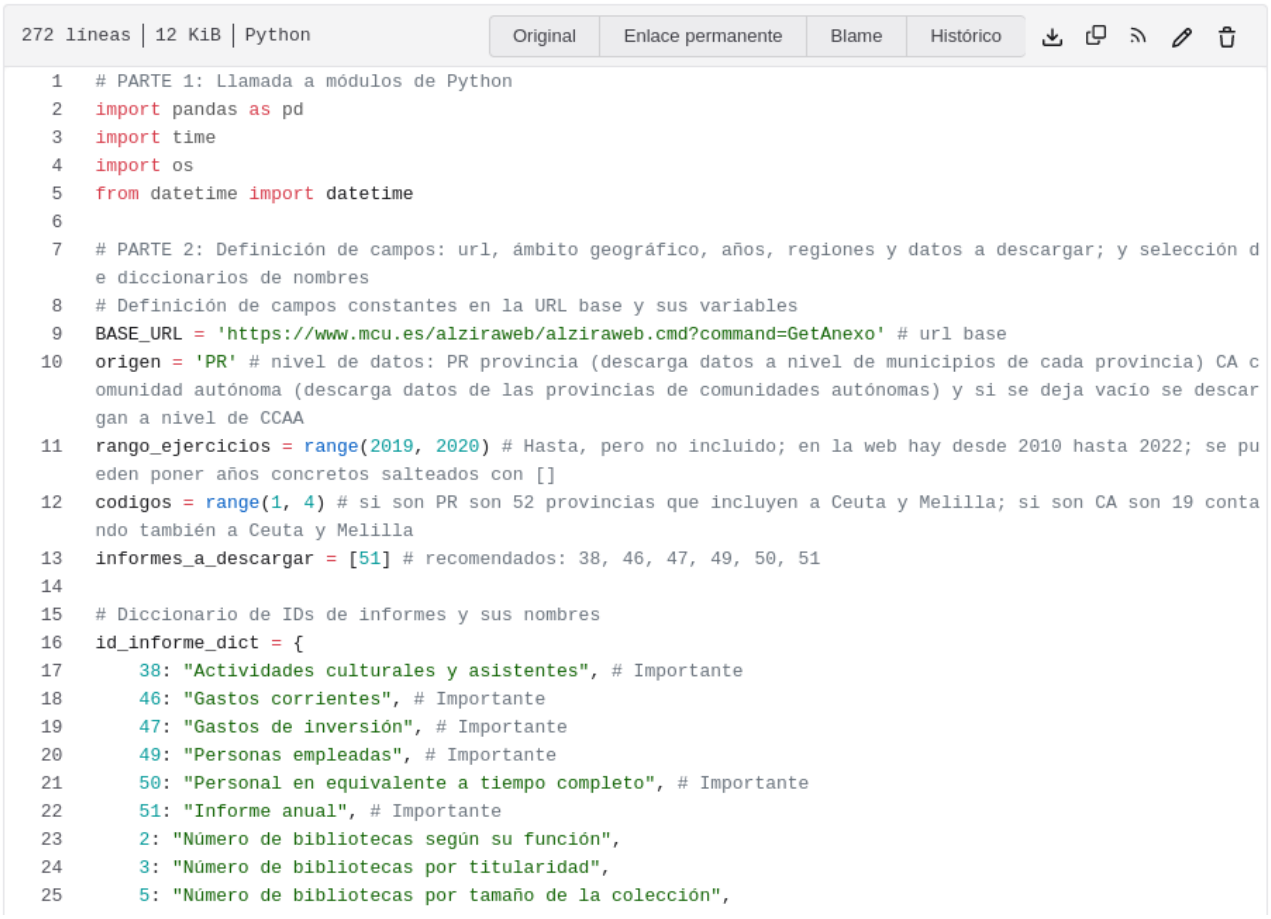
2. *PyDataBibPub*: script en *Python* para automatizar la descarga de datos de bibliotecas públicas de España

2.1. El script

PyDataBibPub está disponible en un proyecto en *Codeberg*, una plataforma para alojar proyectos de software y código, no comercial y con valores de relaciones horizontales propios de la filosofía del software libre, y alternativa a servicios como *GitHub*. El proyecto *PyDataBibPub* tiene 5 archivos:

- *readme.md*: con la presentación del script, unas instrucciones para la configuración previa de *Python* y el entorno para ejecutarlo y la explicación de la parte variable del script según los datos de bibliotecas públicas que se quieran descargar (años, tipo de dato, a nivel de España, comunidades autónomas, provincias, municipios, etc.).
- *LICENSE*: *PyDataBibPub* tiene licencia *GNU Affero General Public License v3.0*.
- *pydatabibpub.py*: el script en sí, con 272 líneas de código en *Python* (figura 1).
- *CHANGELOG.md*: se utilizará para ir avisando de cambios y actualizaciones importantes que puedan surgir para el script.
- *CITATION.md:* la cita recomendada para citar el script por si se usa en futuros trabajos e investigaciones.

El proyecto en *Codeberg* está disponible en este enlace:
https://codeberg.org/plr/PyDataBibPub/




```python
# PARTE 1: Llamada a módulos de Python
import pandas as pd
import time
import os
from datetime import datetime

# PARTE 2: Definición de campos: url, ámbito geográfico, años, regiones y datos a descargar; y selección de diccionarios de nombres
# Definición de campos constantes en la URL base y sus variables
BASE_URL = 'https://www.mcu.es/alziraweb/alziraweb.cmd?command=GetAnexo' # url base
origen = 'PR' # nivel de datos: PR provincia (descarga datos a nivel de municipios de cada provincia) CA comunidad autónoma (descarga datos de las provincias de comunidades autónomas) y si se deja vacío se descargan a nivel de CCAA
rango_ejercicios = range(2019, 2020) # Hasta, pero no incluido; en la web hay desde 2010 hasta 2022; se pueden poner años concretos salteados con []
codigos = range(1, 4) # si son PR son 52 provincias que incluyen a Ceuta y Melilla; si son CA son 19 contando también a Ceuta y Melilla
informes_a_descargar = [51] # recomendados: 38, 46, 47, 49, 50, 51

# Diccionario de IDs de informes y sus nombres
id_informe_dict = {
    38: "Actividades culturales y asistentes", # Importante
    46: "Gastos corrientes", # Importante
    47: "Gastos de inversión", # Importante
    49: "Personas empleadas", # Importante
    50: "Personal en equivalente a tiempo completo", # Importante
    51: "Informe anual", # Importante
    2: "Número de bibliotecas según su función",
    3: "Número de bibliotecas por titularidad",
    5: "Número de bibliotecas por tamaño de la colección",
```

Figura 1. Primeras 25 líneas del archivo *pydatabibpub.py* en *Codeberg*

Por último, la estructura del script tiene 4 partes bien diferenciadas:

- Parte 1: Carga de las bibliotecas (*libraries*) y módulos de *Python* necesarios para ejecutar el script.
- Parte 2: definición de campos variables según las posibilidades y estructura de la página web de los datos.
- Parte 3: funciones de *Python* para descargar y almacenar las tablas de los datos.
- Parte 4: creación de un directorio para guardar y disponer de los *csv* generados.

2.2. Necesidad

El sistema de datos de bibliotecas públicas de España desde la web de *Bibliotecas públicas españolas en cifras* (*Ministerio de Cultura*, 2024a) es de difícil consulta por su estructura y la forma en que se pueden exportar los datos. Estos están disponibles en la sección de “Anexos” (*Ministerio de Cultura*, 2024b) y se clasifican en 14 categorías, por ejemplo: informe anual, bibliotecas (a su vez con 9 tablas de datos), gastos (con 6 tablas), colección (también con 6 tablas), etc. En total, las 14 categorías desplegadas suman 54 tablas de datos (figura 2).

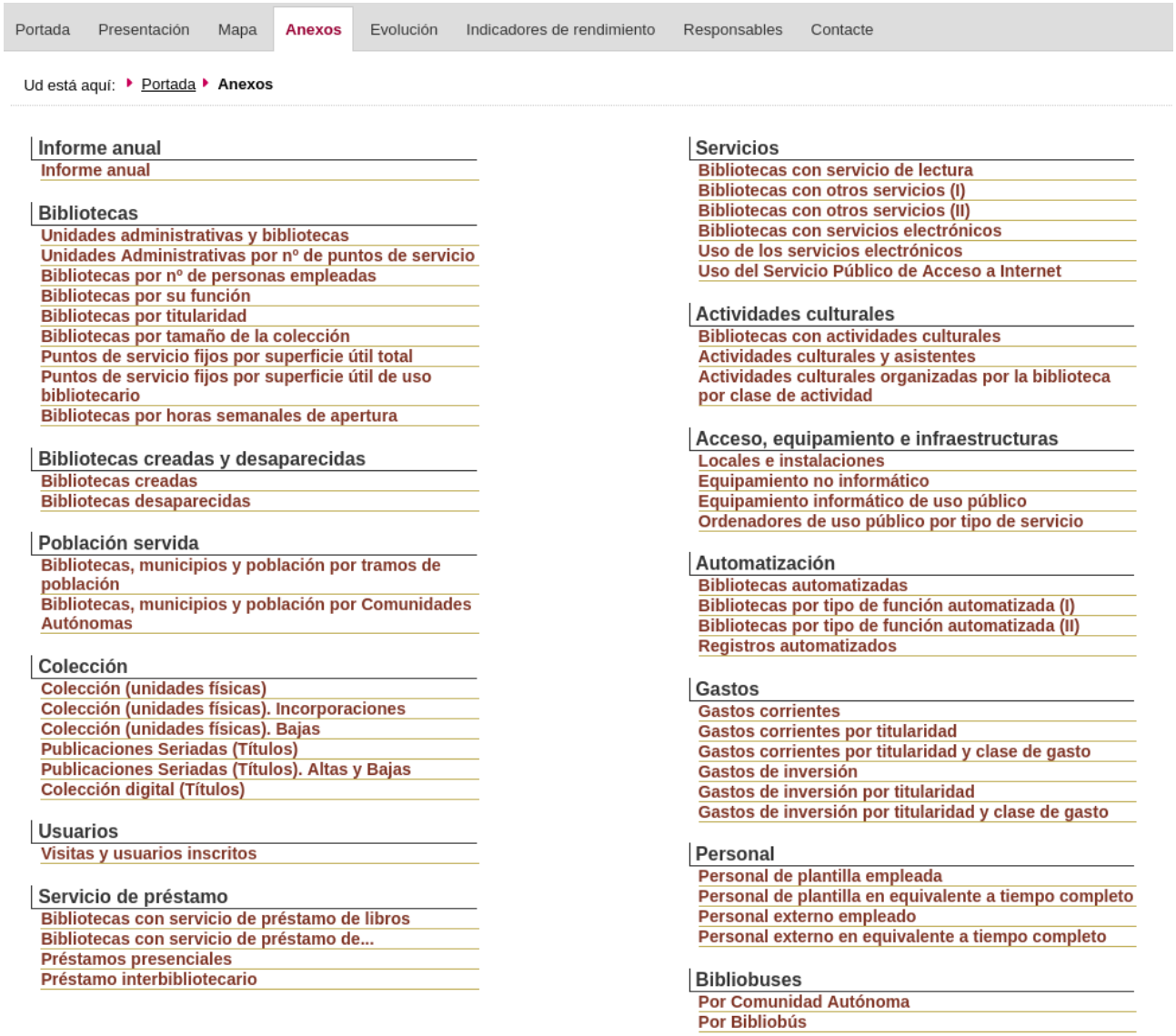


Figura 2. Captura de las categorías y tablas en la web de los datos (*Ministerio de Cultura*, 2024b)

Una vez se accede a una de las 54 tablas, por ejemplo “Gastos corrientes” en la categoría de “Gastos”, se llega a la tabla con los gastos corrientes desplegados (gasto en personal, para adquisiciones, mantenimiento de la colección, etc.), de las 17 comunidades autónomas, las dos ciudades autónomas y el total de España. Encima de la tabla aparece un menú para seleccionar el año de entre los últimos cinco disponibles. Si se quiere acceder a los resultados por provincias, hay que pinchar en cada comunidad autónoma. Y si se quieren tener los datos a nivel de municipios de una provincia, hay que acceder clicando en cada provincia (y previamente en cada comunidad autónoma).

Todo ello genera dificultades y un número muy elevado de clics para, por ejemplo, consultar unos mismos datos de 2 provincias de comunidades autónomas diferentes; o para consultar todas las tablas de todas las categorías de una comunidad autónoma, provincia o municipio. También surgen dificultades para consultar los datos de dos o más municipios de provincias de comunidades diferentes; y más aún si se quieren analizar evoluciones temporales contemplando diferentes años. El número de clics o acciones de volver atrás desde el nivel de municipios al nivel de provincias, y al nivel de comunidades autó-

nomas, y el hecho de tener que acceder a diferentes tablas según las categorías de los datos, dificulta su consulta y la posibilidad de trabajar con ellos de manera eficiente.

Por todo lo expuesto surge la necesidad de disponer y poder descargar de una manera más sencilla, eficiente y directa los datos de bibliotecas públicas de España.

2.3. Antecedentes

En un trabajo de 2022 publicado como preprint (**Lázaro-Rodríguez**, 2022), se presentó un método de descarga de datos a nivel de municipios con el software libre *GNU Wget (GNU Wget, 2024)*. Los archivos de datos a nivel de municipios en formato *Excel* (el único disponible en aquel momento) eran en realidad una url. Con *GNU Wget* se disponía de un archivo *txt* con las urls a descargar, y se descargaban todas volcándolas a una hoja de cálculo en formato *ods*. Todo ello se presentó con un vídeo en *TubEdu*.
https://tubedu.org/w/6vySSWdrFkRdWPvd8yBUah
Aunque el método era eficaz, había que generar las urls en el documento en *txt* con una url por línea.

2.4. ¿Por qué y para qué el script *PyDataBibPub*?

La razón y la finalidad del script se entiende desde una razón expresada en primera persona: por un lado, mi investigación en gran parte gira en torno a la evaluación y calidad de bibliotecas; por otro lado, en los últimos tres cursos he sido docente de la asignatura "Calidad y evaluación de los servicios de documentación" del Grado en Información y Documentación de la *Universidad Complutense de Madrid*. Por todo ello, preciso que la descarga de datos sea lo más directa y sencilla posible. En definitiva, la razón y finalidad del script *PyDataBibPub* es conseguir automatizar la descarga de los datos de una manera eficaz, eficiente y efectiva.

2.5. Creación del script en *Python* mediante *ChatGPT 3.5*

El script se elaboró con la ayuda de *ChatGPT 3.5* empezando todo desde cero, incluso con la instalación de *Python* y los módulos que *ChatGPT 3.5* iba indicando. El proceso se basó en el desarrollo y adaptación de un código a las necesidades que iban surgiendo y las particularidades de la web donde se encuentran los datos de bibliotecas públicas de España.

Todo el proceso está explicado y expuesto en un vídeo que forma parte de una charla de *ConocimIA* (*ConocimIA*, 2024a), un seminario creado en el marco del *Departamento de Biblioteconomía y Documentación* de la *Universidad Complutense de Madrid*. El objetivo de *ConocimIA* es el seguimiento y realización de actividades relacionadas con la documentación, la inteligencia artificial y la recuperación de información (*ConocimIA*, 2024b). El 15 de diciembre de 2023 se llevaron a cabo dos talleres de data-mining con *ChatGPT* como actividades de *ConocimIA*, uno sobre data-mining de *PARES* (*Portal de Archivos Españoles*), y otro sobre data-mining de bibliotecas públicas. En este segundo taller se presentó todo el proceso de creación y desarrollo de *PyDataBibPub*. La exposición está disponible en una lista de reproducción con 7 vídeos en *YouTube*:
https://www.youtube.com/playlist?list=PLv7qdgOim8tLCCM7vYxY5bCKnSbG7XIcX

2.6. ¿Qué hace exactamente el script y cómo lo hace?

El script descarga datos de bibliotecas públicas de España según unos criterios específicos. Los datos y sus posibilidades parten de una url de base sobre la que se pueden matizar variables. Se propone un análisis de la siguiente url:

https://www.mcu.es/alziraweb/alziraweb.cmd?command=GetAnexo&origen=PR&codigo=44&id=46&porcentaje=&ejercicio=2020

Esta url es en realidad la tabla de "Gastos corrientes" para el año 2020 de los municipios de la provincia de Teruel (perteneciente a la comunidad autónoma de Aragón). Los campos variables son los siguientes:

- *origen=PR*: se refiere al nivel de los datos. En este caso *PR* es a nivel de municipios de una provincia.
- *codigo=44*: se refiere a la provincia concreta. La 44 es Teruel.
- *id=46*: se refiere a la tabla de datos concreta. En este caso, el *id* 46 es la tabla de datos de los gastos corrientes de las bibliotecas.
- *porcentaje=*: en este caso no aporta nada porque no hay nada después del signo igual. De hecho se puede omitir de la url esta parte (*&porcentaje=*) y se llega a la misma tabla.
- *ejercicio=2020*: se refiere al año de los datos. Llama la atención que aunque en el menú seleccionable de años solo aparecen los últimos 5 (a fecha de redacción de este artículo desde 2018 a 2022), si en la url se escribe el año 2012, aparecen los datos de 2012, y así están disponibles desde el año 2010.

Estos criterios considerados como variables en el script han de ser especificados en la llamada "parte 2" del *código* y dependen de lo que se quiera descargar. Las posibilidades en el script son las siguientes:

- Los datos se pueden descargar tanto a nivel de municipios, como de provincias o a nivel de comunidades autónomas. Es lo que en el código se delimita como la variable *origen*.
- Cabe la posibilidad de definir el año o rango de años para los datos, en este caso en el script es la variable llamada *rango_ejercicios*.
- Se pueden definir comunidades autónomas o provincias concretas para descargar datos de las provincias de una comunidad o de los municipios de una provincia; y también definir rangos o descargar todas las comunidades autónomas, provincias y todos los municipios. En este caso, la variable en el script se llama *codigos*.
- Por último, se pueden descargar tablas de datos determinadas. La variable se llama *informes_a_descargar*.

En la llamada "parte 2" del script donde hay que definir estos criterios, se han añadido comentarios que aclaran las diversas posibilidades (figura 3). Además, tras estas líneas de código, se añaden diccionarios para entender las opciones de cada variable. Es importante dejar claro que los significados de estas opciones de los diccionarios de variables vienen determinados por la web donde se alojan los datos, y se han podido definir tras un análisis exhaustivo de la forma en que se estructura la web de los datos y las tablas que los contienen con sus diferentes opciones.

Figura 3. Parte 2 del script donde se definen las variables

Una vez con todo lo anterior definido, se ejecuta el script, y se empiezan a descargar datos. En el lugar donde se ha abierto el entorno de *Python* se genera una carpeta con un nombre según esta estructura:

descarga_tablasdescargadas_años_fechadedescarga_horadedescarga

Dentro de esta carpeta el script genera archivos en formato csv para cada tabla de datos descargada. Lo bueno del script tal y como está diseñado es que unifica en un único csv los datos de todas las provincias, municipios (según sea el nivel definido) y todos los años seleccionados por tabla de datos. Por ejemplo, si se define descargar los gastos corrientes de todos los municipios de España para 5 años, el script crea en la carpeta un único csv con los datos referidos a gastos corrientes de todos los municipios de España y para todos los años definidos; no crea un csv por año, o por provincia, etc. Esto facilita el trabajo posterior con los datos.

En la figura 4 puede verse un ejemplo de funcionamiento del script y de los mensajes que va generando según se van descargando las tablas y unificándose en el archivo en csv. En este caso, se han descargado las tablas de los informes anuales para los municipios de Álava, Albacete y Alicante, y para los años 2021 y 2022, generando un único csv dentro de la carpeta también generada.

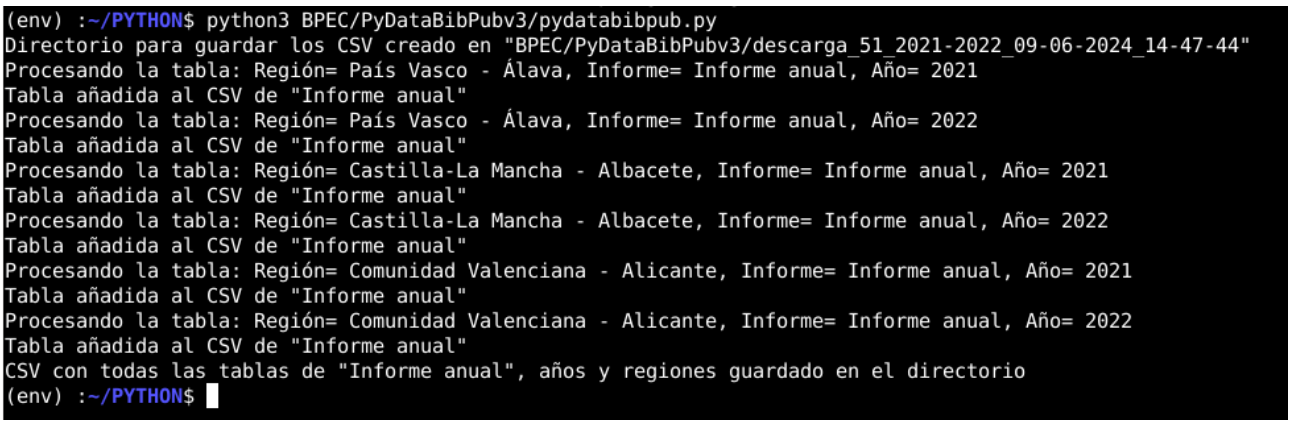


Figura 4. Ejemplo de ejecución e interfaz del script

2.7. Limitaciones y posibilidades del script

El script satisface unas necesidades concretas en el contexto de la investigación en biblioteconomía y documentación. Todo el script depende del sistema o forma de las urls de la página de los datos. También, depende de que esa web o servidor acepte peticiones. Hasta ahora no ha habido ningún problema y se entiende que no habría de haberlos, porque el script simplemente automatiza clics en la web.

Sí que hay que decir que a finales del año 2023 las urls de la web de los datos cambiaron, y hubo que actualizar el script para solventarlo. Estos cambios en páginas web que dependen de ministerios concretos pueden deberse a los cambios de Gobierno y su organización. En ese sentido, que el script esté alojado en *Codeberg* facilita que se vaya actualizando según surjan necesidades o se produzcan cambios, y para ello se ha habilitado un archivo *CHANGELOG.md* para grabar todos los cambios y actualizaciones que se vayan haciendo.

En cuanto a las posibilidades, por un lado, la licencia *GNU Affero General Public License v3.0* marcada para el script permite, de manera resumida, acceder al código del script y define su libertad de uso. Así mismo, permite desarrollar e implementar posibles modificaciones y mejoras del mismo, con colaboraciones incluso en el mismo proyecto del script en *Codeberg*.

2.8. Licencia de reutilización de los datos que descarga el script

El aviso legal del *Ministerio de Cultura* del *Gobierno de España* y que afecta al contenido de la web que utiliza el script para la descarga de datos fue actualizado en abril del año 2024 (*Ministerio de Cultura*, 2024c). **Lázaro-Rodríguez** (2024) analiza con detalle esta actualización, concluyendo que el nuevo aviso legal es más claro y tiene implicaciones positivas para la reutilización de los datos de bibliotecas públicas.

Es importante considerar que el script no desnaturaliza el sentido de la información de la fuente de los datos. Se considera que simplemente automatiza la descarga de datos, descarga que habría de hacerse con un elevado número de clics en la web de origen. También y más importante aún, hay que tener en cuenta que si se usa el script y se reutilizan los datos descargados en alguna investigación u otro tipo de publicación, es necesario añadir una cita y referencia a la fuente de los datos descargados, todo en base al contenido de la

actualización del aviso legal del *Ministerio de Cultura*. Con todo, se recomienda añadir una referencia a la fuente según esta posible forma:

> Ministerio de Cultura [año]. Bibliotecas públicas españolas en cifras (BPEC). Portada. Ministerio de Cultura.
> *https://www.cultura.gob.es/cultura/areas/bibliotecas/mc/ebp/portada.html*

Por último, la referencia recomendada para el script se ha definido en el archivo del proyecto en *Codeberg* llamado *CITATION.md* y la referencia queda así:

> Lázaro-Rodríguez, Pedro (2024). "*PyDataBibPub*: script en *Python* para extraer datos de bibliotecas públicas de España".
> *https://codeberg.org/plr/PyDataBibPub*

3. Consideraciones finales

En la introducción de este artículo se mencionaron trabajos que plantean la posibilidad de usar *ChatGPT* para tareas de programación en distintos lenguajes (**Torres-Salinas**; **Arroyo-Machado**, 2023; **Torres-Salinas**; **Thewall**; **Arroyo-Machado**, 2024), y se revisaron otros que analizan la eficacia y uso de *ChatGPT* para generar código concretamente en *Python* (**Adamson**; **Bägerfeldt**, 2023; **Diehl** *et al.*, 2024; **Wuisang** *et al.*, 2023; **Lladós-Clos**, 2024). El script *PyDataBibPub* presentado en este trabajo es un ejemplo más de lo que se puede conseguir con herramientas basadas en LLMs (*Large Language Models*) y en IAG (Inteligencia Artificial Generativa) para la generación de código y desarrollo de aplicaciones y software para fines específicos.

Aunque el uso de *ChatGPT 3.5* se valora como positivo en la creación del *PyDataBibPub* por su efectividad en la automatización de la descarga de datos, es justo también reflexionar sobre el uso de programas basados en LLMs y en IAG desde un punto de vista ético y moral, teniendo en cuenta los recursos que emplean y las posibles dependencias que generan con terceras partes y grandes empresas. En ese sentido, uno de los próximos pasos y retos a explorar como superación de estos posibles problemas es la instalación de programas de IAG y LLMs que funcionan en local, garantizando la privacidad y la independencia con terceros.

Una línea de trabajo futura es la instalación de *PrivateGPT* o similar que funciona en local con independencia de grandes empresas, para trabajar con lenguajes de programación desarrollando aplicaciones para la evaluación de bibliotecas en particular y para la biblioteconomía y la documentación en general

Un ejemplo es la implementación por **Blázquez-Ochando** (2024a) de lo que dio en llamar primera inteligencia artificial en documentación, montada sobre una instalación en local de *PrivateGPT* (**Martínez-Toro**; **Gallego-Vico**; **Orgaz**, 2023). La instalación de **Blázquez-Ochando** (2024a) con una demostración de su funcionamiento y uso se presentó en una charla del seminario *ConocimIA* en abril de 2024 y está disponible en un vídeo en *YouTube* (**Blázquez-Ochando**, 2024b).

4. Referencias

Adamson, Victor; **Bägerfeldt, Johan** (2023). *Assessing the effectiveness of ChatGPT in generating Python code* [Student thesis].
https://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-22860

Blázquez-Ochando, Manuel (2024a). *La IA en tus manos - Primera IA en Documentación*.
https://mblazquezbis.es/conocimia/wp-content/uploads/conocimIA-2024-04-26-primera-ia-documentacion.pptx

Blázquez-Ochando, Manuel (2024b). *Conferencia ConocimIA - La IA en tus manos – 2024/04/26*.
https://www.youtube.com/watch?v=cb2n_lQ4-XE

ChatGPT Impact (2024). *Scholarly Publications ChatGPT – Scientific publications by day*.
https://www.chatgptimpact.com/scholarly-publications/

Codina, Lluís (2023). "Buscadores alternativos a Google con IA generativa: análisis de You.com, Perplexity AI y Bing Chat". *Infonomy*, v. 1, n. 1, e23002.
https://doi.org/10.3145/infonomy.23.002

ConocimIA (2024a). *ConocimIA: La iniciativa ConocimIA - Seminario de IA en Documentación*.
https://conocimia.digital/

ConocimIA (2024b). *Qué es ConocimIA*.
https://mblazquezbis.es/conocimia/que-es-conocimia/

Diehl, Patrick; **Nader, Noujoud**; **Brandt**, **Steve**; **Kaiser, Hartmut** (2024). Evaluating AI-generated code for C++, Fortran, Go, Java, Julia, Matlab, Python, R, and Rust. *Arxiv.org*.
https://doi.org/10.48550/arXiv.2405.13101

Franganillo, Jorge (2023). Los grandes modelos de lenguaje: una oportunidad para la profesión bibliotecaria. *Anuario ThinkEPI*, v. 17, e17a28.
https://doi.org/10.3145/thinkepi.2023.e17a28

González-Alcaide, Gregorio (2024). Inteligencia artificial generativa: Un contexto disruptivo en el acceso a la información. *Infonomy*, v. 2, n. 1, e24013.
https://doi.org/10.3145/infonomy.24.013

Goyanes, Manuel; **Lopezosa, Carlos** (2024). ChatGPT en Ciencias Sociales: revisión de la bibliografía sobre el uso de inteligencia artificial (IA) de OpenAI en investigación cualitativa y cuantitativa. *Anuario ThinkEPI*, v. 18, e18e04.
https://doi.org/10.3145/thinkepi.2024.e18a04

Lázaro-Rodríguez, Pedro (2022). A vueltas con los datos: ¿inconsistencias en las estadísticas de bibliotecas públicas de España 2019? Recomendaciones para la mejora. *OSF Preprints*.
https://doi.org/10.31219/osf.io/8a9dq

Lázaro-Rodríguez, Pedro (2024). Cambios en el aviso legal del Ministerio de Cultura: implicaciones positivas y más claras para los datos de bibliotecas públicas de España y una propuesta con mayor claridad. *Anuario ThinkEPI*, v. 18, e18e10.
https://doi.org/10.3145/thinkepi.2024.e18a10

Lladós-Clos, Jordi (2024). Analysis of the utility of ChatGPT in the development of a Python application for environmental data processing [Trabajo final de grado]. Universitat Politècnica de Catalunya. *http://hdl.handle.net/2117/400670*

Lopezosa, Carlos (2023a). Bing chat: hacia una nueva forma de entender las búsquedas. *Anuario ThinkEPI*, v. 17, e17a04.
https://doi.org/10.3145/thinkepi.2023.e17a04

Lopezosa, Carlos (2023b). La inteligencia artificial en los procesos editoriales de las revistas académicas: propuestas prácticas. *Infonomy*, v. 1, n. 1, e23009.
https://doi.org/10.3145/infonomy.23.009

Lopezosa, Carlos; **Codina, Lluís** (2023). Probando Bard: así funciona la Inteligencia Artificial Generativa de Google. *Anuario ThinkEPI*, v. 17, e17a25.
https://doi.org/10.3145/thinkepi.2023.e17a25

Martínez-Toro, Iván; **Gallego-Vico, Daniel**; **Orgaz, Pablo** (2023). *PrivateGPT*.
https://github.com/imartinez/privateGPT

Ministerio de Cultura (2024a). *Bibliotecas públicas españolas en cifras (BPEC). Portada. Ministerio de Cultura*.
https://www.cultura.gob.es/cultura/areas/bibliotecas/mc/ebp/portada.html

Ministerio de Cultura (2024b). *Bibliotecas públicas españolas en cifras (BPEC). Anexos. Ministerio de Cultura*.
https://www.mcu.es/alziraweb/alziraweb.cmd?command=GetAnexos

Ministerio de Cultura (2024c). *Aviso legal - | Ministerio de Cultura*.
https://www.cultura.gob.es/cultura/areas/bibliotecas/mc/ebp/comunes/aviso-legal.html

Torres-Salinas, Daniel; **Arroyo-Machado, Wenceslao** (2023). *Manual de ChatGPT: aplicaciones en investigación y educación universitaria 2.0* [Computer Software]. InfluScience Ediciones. *https://doi.org/10.5281/zenodo.10390816*

Torres-Salinas, Daniel; **Thelwall, Mike**; **Arroyo-Machado, Wenceslao** (2024). ChatGPT for Bibliometrics: A comprehensive corpus of applications. *Zenodo*.
https://doi.org/10.5281/zenodo.11103551

Wuisang, Marchel-Christhoper; **Kurniawan, Marcel**; **Wira-Santosa, Komang-Andika**; **Santoso-Gunawan, Alexander-Agung**; **Saputra, Karen-Etania** (2023). An evaluation of the effectiveness of OpenAI's ChatGPT for automated Python program bug fixing using QuixBugs. En: *2023 International Seminar on Application for Technology of Information and Communication (iSemantic),* 295-300.
https://doi.org/10.1109/iSemantic59612.2023.10295323