

ETL Cuentas Cooperativas CV

Guía de uso

Daniel Chorro Juan - Febrero 2023

Fuentes

Todo el código puede descargarse desde: [URL](#)

Entorno y Librerías

Este proyecto se ha desarrollado utilizando Anaconda, un entorno que intenta simplificar el manejo de los paquetes. El script se podrá ejecutar simplemente instalando el entorno adjunto (conda env create -f practicas_ade.yml).

Por otra parte, se muestra a continuación un listado con todas las librerías y las versiones instaladas en este entorno, por si el usuario no quisiera hacerlo desde una instalación limpia de Anaconda.

| # | Name | Version | Build | Channel |
|---|--------------------|--------------|--------------------|-------------|
| | asgiref | 3.6.0 | pyhd8ed1ab_0 | conda-forge |
| | backports.zoneinfo | 0.2.1 | py39hcbf5309_7 | conda-forge |
| | bzip2 | 1.0.8 | h8ffe710_4 | conda-forge |
| | ca-certificates | 2022.12.7 | h5b45459_0 | conda-forge |
| | django | 4.1.5 | pyhd8ed1ab_0 | conda-forge |
| | libffi | 3.4.2 | h8ffe710_5 | conda-forge |
| | libsqlite | 3.40.0 | hcfcfb64_0 | conda-forge |
| | libzlib | 1.2.13 | hcfcfb64_4 | conda-forge |
| | openssl | 3.0.7 | hcfcfb64_2 | conda-forge |
| | pip | 22.3.1 | pyhd8ed1ab_0 | conda-forge |
| | python | 3.9.15 | h4de0772_0_cpython | conda-forge |
| | python_abi | 3.9 | 3_cp39 | conda-forge |
| | setuptools | 66.1.1 | pyhd8ed1ab_0 | conda-forge |
| | sqlparse | 0.4.3 | pyhd8ed1ab_0 | conda-forge |
| | tk | 8.6.12 | h8ffe710_0 | conda-forge |
| | typing_extensions | 4.4.0 | pyha770c72_0 | conda-forge |
| | tzdata | 2022g | h191b570_0 | conda-forge |
| | ucrt | 10.0.22621.0 | h57928b3_0 | conda-forge |
| | vc | 14.3 | hb6edc58_10 | conda-forge |
| | vs2015_runtime | 14.34.31931 | h4c5c07a_10 | conda-forge |
| | wheel | 0.38.4 | pyhd8ed1ab_0 | conda-forge |
| | xlsxwriter | 3.0.7 | pyhd8ed1ab_0 | conda-forge |
| | xz | 5.2.6 | h8d14728_0 | conda-forge |

Estructura de ficheros

El script trabaja con los archivos .XBRL de cada empresa, por lo tanto, en adelante cuando mencionamos las cuentas, realmente nos estamos refiriendo a estos archivos.

Las cuentas de las cooperativas de Valencia deben estar en la carpeta datos/Coop_Agroalimentarias, y es recomendable guardarlas en RMValencia y el año correspondiente. Si el año no está disponible, se puede crear una carpeta nueva sin problemas. Las cuentas de las cooperativas de Castellón deben estar en la carpeta datos/RM Castellón. Las cuentas de las cooperativas de Alicante deben estar en la carpeta datos/RM_Alicante y opcionalmente en la carpeta correspondiente al año. Aunque no es estrictamente necesario, es recomendable hacerlo para un mejor funcionamiento del programa. La siguiente figura representa la estructura recomendada de ficheros.

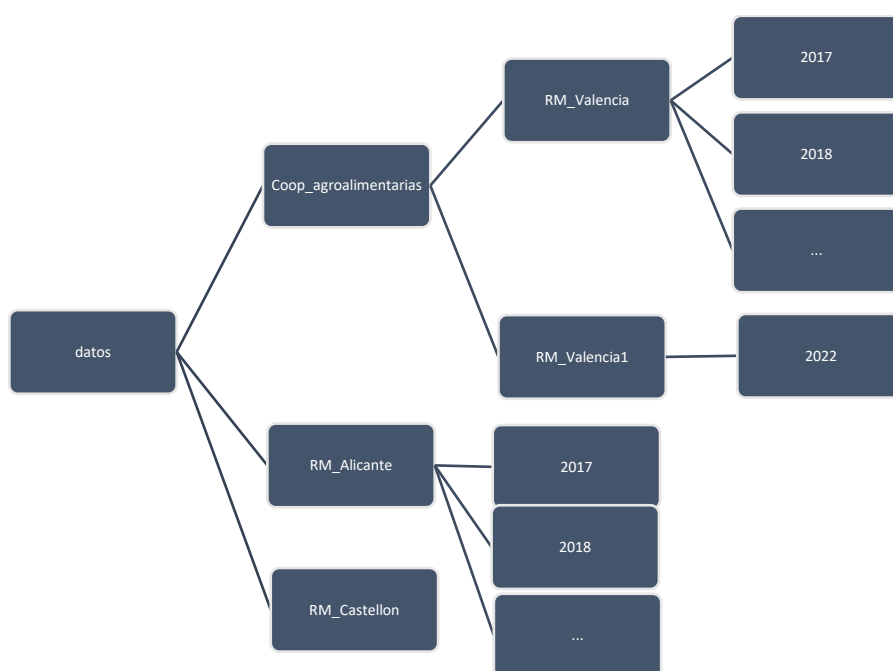


Ilustración 1 - Estructura de ficheros con la que se ha realizado el script

Nota: En este caso, las carpetas dentro de RM_Valencia1 no contienen archivos .xbrl, es decir, este script no nos sirve para extraer esas cuentas.

Nos podemos encontrar con que las cuentas de alguna provincia vienen comprimidas dentro de archivos zip, como explicaremos a continuación, de esto también puede encargarse el script (parámetro -z en la línea de comandos).

Nota: Si el usuario elige la opción de descomprimir en la carpeta de Valencia, se encontrarán archivos comprimidos con XBRL. Idealmente, se le notificará al usuario para que decida qué hacer con cada archivo (reemplazar, omitir, etc.). Si al descomprimir no se reemplazan los archivos previos (e.g., versiones previas de los ficheros de datos), podría darse el caso de haber dos archivos XBRL por cada año, lo que podría causar problemas de duplicidad de datos, coexistiendo versiones nuevas con antiguas.

Uso del script

Se recomienda usar el script desde la carpeta `.. \datos\`, es decir, desde la carpeta anterior a la carpeta datos. En esta carpeta es donde el script guardará el archivo Excel que creará.

Para ejecutar el script usaremos la siguiente instrucción:

```
python file_parser.py <carpetacuentas> <nombre_excel> <-z> (opcional)
```

Como ejemplo de uso, considerando que tenemos una estructura de carpetas y ficheros donde las cuentas se encuentran dentro de la carpeta datos, y concretamente las cuentas de Valencia dentro de la carpeta `datos\ Coop_agroalimentarias`, para crear el Excel con estas cuentas, ejecutaríamos el script de la siguiente manera:

```
python file_parser.py datos\Coop_agroalimentarias Cuentas_Valencia
```

De esta manera crearíamos un archivo Excel llamado `Cuentas_Valencia.xlsx` con los archivos XBRL de la carpeta `datos\Coop_agroalimentarios` y no extraeríamos los archivos .zip que encontraríamos.

```
python file_parser.py "datos\RM Castellón" Cuentas_Castellon -z
```

La anterior instrucción descomprimiría los archivos zip de la carpeta `"datos\RM Castellón"` (las comillas son porque RM Castellón tiene un espacio) y crearía un archivo Excel llamado `Cuentas_Castellon` con los archivos XBRL de la carpeta.

Nota: Actualmente, todos los archivos XBRL de las carpetas se convierten a ficheros de Excel.