Caso práctico: manipulación de ficheros de texto

PID_00270625

Gerard Farràs Ballabriga

Tiempo mínimo de dedicación recomendado: 1 hora

Universitat Oberta de Catalunya

Gerard Farràs Ballabriga

Ingeniero técnico en Informática de sistemas por la Universidad Autónoma de Barcelona (UAB). Ingeniero en Informática y máster en Sociedad de la información y el conocimiento por la Universitat Oberta de Catalunya (UOC). Actualmente trabaja como profesor en una escuela de secundaria y formación profesional. Anteriormente desarrolló su actividad profesional en el área de sistemas de información de un centro tecnológico y también como profesional autónomo (*freelance*) trabajando como administrador de sistemas y desarrollador web.

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por el profesor: Julià Minguillón Alfonso (2020)

Primera edición: febrero 2020

Avda. Tibidabo, 39-43, 08035 Barcelona
Realización editorial: FUOC

Índice

1. Introducción

El Servicio Catalán de la Salud genera un catálogo actualizado mensualmente con «todos los medicamentos dispensables en oficinas de farmacia y los productos sanitarios del Sistema Nacional de Salud que financia el CatSalut mediante las recetas médicas oficiales. Incluye los datos de identificación del producto, los datos económicos y de composición». Este fichero está en un formato de texto plano y es automatizable, ya que sigue un formato concreto.

Catálogo de productos farmacéuticos

La información completa de este catálogo está disponible en el enlace: catsalut.gencat.cat/ca/proveidors-professionals/registres-catalegs/catalegs/productes-farmaceutics/index.html#googtrans(cales) (pestaña «Descarga del catálogo completo»).

El objetivo de este ejercicio consiste en implementar un sistema automatizado que, el día 2 de cada mes, obtenga dos campos concretos del apartado nomenclátores medicamentos y productos sanitarios (la «Descripción producto farmacéutico» y el «precio de comercialización») y genere un fichero csv con estos dos valores.

Este primer caso tiene que servir al estudiante como muestra para tratar ficheros de texto plano, sin formatos, como, por ejemplo, csv, xml o json. Tampoco emplea ninguna API externa para la obtención de los datos. Se trata de la lectura y tratamiento de un fichero grande que contiene solamente texto.

Palabras clave

Ficheros TXT, comandos `wget`, `head`, `cut`, `tail`, `paste`.

Figura 1. Sitio web del Servicio Catalán de la Salud desde donde podéis descargar el catálogo completo que se trabajará en este caso

Fuente: catsalut.gencat.cat/ca/proveidors-professionals/registres-catalegs/catalegs/productes-farmaceutics/index.html#googtrans(cales)

2. Pasos

El primer caso consiste en descargar el catálogo y también su formato para tenerlo como referencia.

Nota

Aunque el editor de texto parte los enlaces que no caben en una sola línea, se trata solamente de un comando.

```
usuario@nombreMaquina:~$ wget -q https://catsalut.gencat.cat/web/.content/minisite/catsalut/
proveidors_professionals/registres_catalegs/documents/catalegfarmacia.zip

usuario@nombreMaquina:~$ ls -lh catalegfarmacia.zip
-rw-r--r-- 1 usuario usuario 8,4M de se 3 13:04 catalegfarmacia.zip
```

Observamos que este fichero comprimido ocupa 8,4 megabytes. Lo podemos descomprimir con el comando siguiente:

```
usuario@nombreMaquina:~$ unzip catalegfarmacia.zip
Archive: catalegfarmacia.zip
inflating: CATALEGFARMACIA20190901.TXT
```

También será útil disponer del formato del catálogo, para tenerlo como referencia:

```
usuario@nombreMaquina:$ wget -q https://catsalut.gencat.cat/web/.content/minisite/catsalut/
proveidors_professionals/registres_catalegs/documents/for_extrac_cpf.pdf
```

El fichero descomprimido es un texto plano sin elementos separadores de cada campo, puesto que cada uno de ellos se especifica en una posición concreta (se recomienda echar un vistazo a las primeras páginas del formato del catálogo). Podemos obtener información de este fichero con los comandos siguientes:

```
usuario@nombreMaquina:~$ file CATALEGFARMACIA20190901.TXT
CATALEGFARMACIA20190901.TXT: ISO-8859 text, with very long lines, with CRLF line terminators

usuario@nombreMaquina:~$ wc -l CATALEGFARMACIA20190901.TXT
511540 CATALEGFARMACIA20190901.TXT
```

Este fichero concreto tiene más de medio millón de líneas. Tal como especifica el formato del catálogo, el primer registro contiene la cabecera del fichero:

```
usuario@nombreMaquina:~$ head -1 CATALEGFARMACIA20190701.TXT
001 PFC00013S20190701 2019062810091200473879000043 2019HPIT3PFC PFC 00013 0001300130013000000
```

Y el segundo, el registro de cabecera de detalle de los nomenclátores de los medicamentos y productos sanitarios:

```
usuario@nombreMaquina:~$ head -2 CATALEGFARMACIA20190701.TXT | tail -1
100000101 2019070120190630201906281009120006311 5PFC18001SNOMENCLATOR
EF NORMALES Y EFECTOS / ACCESORIOS
```

El número de registros del grupo que dependen de la cabecera de detalle está en esta línea (se trata de una cifra de ocho caracteres; el campo concreto se denomina «Número de registros del grupo que dependen de la cabecera de detalle»). Obtenemos este valor con el comando siguiente:

```
usuario@nombreMaquina:~$ d=`head -2 CATALEGFARMACIA20190901.TXT | tail -1 | cut -c 55-62`
```

Este comando ejecuta el *head*, que muestra las dos primeras líneas del fichero, las traspasa a un *pipe*, donde el *tail* se quedará con la última línea y, finalmente, lo pasa con otro *pipe* al comando `cut`, que recortará los caracteres del 55 al 62. Todo ello se almacenará en una variable que hemos denominado `d`. Con el comando siguiente podemos observar el valor que hemos obtenido:

```
usuario@nombreMaquina:~$ echo $d
00063353
```

Hay que tratar `$d` líneas, aunque que las dos anteriores, si recordamos, contenían registros de cabecera. Por tanto:

```
n=`expr $d + 2`
```

Filtramos ahora por el campo «Descripción producto farmacéutico» (que está, según el catálogo, en los valores 22 y 121 y ocupa un total de 100 caracteres):

```
usuario@nombreMaquina:~$ head -$n CATALEGFARMACIA20190901.TXT | tail -$d | cut -c22-121
```

Hemos obtenido uno de los campos que estábamos buscando.

Este resultado contiene más de 60.000 líneas (en el momento de hacer este ejercicio, concretamente, 63.115, aunque es una cifra que puede variar). A modo de demostración, mostraremos las 15 primeras:

```
usuario@nombreMaquina:~$ head -$n CATALEGFARMACIA20190901.TXT | tail -$d | cut -c22-121 | head -15
APALOZ 5 MG COMPRIMIDOS EFG , 28 comprimidos
MEDIA LARGA (A-F) COMP NORMAL KURVAY-B (500 TALLA PEQUEÑA
SONDA VESICAL SILICONA FOLEY SILICONA 100% CH12 B10
SONDA VESICAL SILICONA FOLEY SILICONA 100% CH14 B10
SONDA VESICAL SILICONA FOLEY SILICONA 100% CH16 B10
SONDA VESICAL SILICONA FOLEY SILICONA 100% CH18 B10
SONDA VESICAL SILICONA FOLEY SILICONA 100% CH20 B10
SONDA VESICAL SILICONA FOLEY SILICONA 100% CH22 B10
SONDA VESICAL SILICONA FOLEY SILICONA 100% CH24 B10
MEDIA LARGA (A-F) COMP NORMAL KURVAY-B (500 TALLA MEDIANA
BOLSAS ILEOST RES SINT MIC FIL MODERMA FLEX ABIERTA PLANA MINI OPACA 15-55MM 30U
BOLSAS ILEOST RES SINT MIC FIL MODERMA FLEX ABIERTA CONVEXA MAXI OPACA 15-38MM 30U
BOLSAS ILEOST RES SINT MIC FIL MODERMA FLEX ABIERTA CONVEXA MAXI OPACA 15-51MM 30U
BOLSAS ILEOST RES SINT MIC FIL MODERMA FLEX ABIERTA CONVEXA MAXI TRANSPARENTE 15-38MM 30U
BOLSAS ILEOST RES SINT MIC FIL MODERMA FLEX ABIERTA CONVEXA MAXI TRANSPARENTE 15-51MM 30U
```

De manera similar, obtendremos otro campo: «Precio de comercialización», que se expresa con ocho números (seis valores enteros y dos decimales) y está entre la posición 131 y 138:

```
usuario@nombreMaquina:~$ head -$n CATALEGFARMACIA20190901.TXT | tail -$d | cut -c131-138 | head -15
00000000
00000652
00000809
00000809
```

```
00000809
00000809
00000809
00000809
00000809
00000652
00006850
00006850
00006850
00006850
00006850
```

Con objeto de generar un fichero .csv con la información de los dos campos, los mezclaremos con el comando `paste`. Para hacerlo, antes generaremos un par de ficheros auxiliares con la información de cada campo:

```
usuario@nombreMaquina:~$ head -$n CATALEGFARMACIA20190901.TXT | tail -$d | cut -c22-121
> descripcion.txt
usuario@nombreMaquina:~$ head -$n CATALEGFARMACIA20190901.TXT | tail -$d | cut -c131-138
> coste.txt
```

Con el comando siguiente es posible mezclar el contenido de los dos ficheros (aquí solamente se muestra la primera línea, a modo de ejemplo):

```
usuario@nombreMaquina:~$ paste -d ';' descripcion.txt coste.txt | head -1
APALOZ 5 MG COMPRIMIDOS EFG , 28 comprimidos ;00000000
```

El parámetro `-d` indica el separador que deseamos.

Si se quisiera proteger con comillas ("), en los dos campos se podrían hacer sustituciones similares a las siguientes:

```
usuario@nombreMaquina: $ paste -d ';' descripcion.txt coste.txt |
sed s/^/\"/g | sed s/$/\"/g | sed s/\;/\"\;\"/g > datos.csv
```

En el primer `sed`, se sustituye el inicio de cada línea agregando unas comillas (recordemos que se indica el inicio de cada línea usando el carácter especial `^`), en el segundo lo mismo, pero al final de línea (que se indica con el carácter `$`) y, en el último, se sustituye el carácter separador que habíamos agregado `; ,` por `;`. Recordemos que las `\` sirven para proteger caracteres especiales.

Quedaría solo automatizar este proceso para que solamente se ejecutara el día 2 de cada mes a una hora específica.

Primero habría que agregar todos los pasos en un solo *script*. Este guion suprime primero los posibles ficheros resultantes de ejecuciones anteriores. El nombre del fichero también varía cada mes (aunque empieza siempre por «CATALEGFARMACIA» y finaliza con un «.TXT», se cambian las cifras que indican la fecha). Por tanto, habrá que obtener el nombre del fichero en concreto. Recordemos que las líneas que empiezan por una almohadilla `#` actúan como comentarios y se pueden obviar si se escribe el *script*.

Hemos denominado este *script* «obtieneDatos.sh» y se muestra el código a continuación:

```
#!/bin/bash
# Script para tratar datos.

# Accedemos primero a la carpeta donde deseamos realizar todo el tratamiento.
cd /home/usuario/

# Suprimimos posibles ficheros anteriores.
# El parámetro -f fuerza a no pedir nada al usuario.
rm -f catalegfarmacia.zip
rm -f CATALEGFARMACIA*.TXT
rm -f descripcion.txt
rm -f coste.txt
rm -f datos.csv

# Obtenemos el fichero y lo descomprimimos.
wget https://catsalut.gencat.cat/web/.content/minisite/catsalut/proveidors_professionals/
registres_catalegs/documents/catalegfarmacia.zip

unzip catalegfarmacia.zip

# Obtenemos el nombre del fichero concreto, puesto que varía cada mes.
# Se ubica en una variable de entorno.
nombrefichero=`ls CATALEGFARMACIA*.TXT`


# Obtenemos el número de registros que se van a tratar.
d=`head -2 $nombrefichero | tail -1 | cut -c 55-62`
n=`expr $d + 2`

# Generamos un fichero diferente para cada campo.
head -$n $nombrefichero | tail -$d | cut -c22-121 > descripcion.txt
head -$n $nombrefichero | tail -$d | cut -c131-138 > coste.txt

# Unimos ambos ficheros y generamos un fichero .csv
paste -d ';' descripcion.txt coste.txt | sed s/^/\"/g | sed s/$/\"/g | sed s/\;/\"\;\"/g > datos.csv
```

Recordemos que hay que agregar permisos de ejecución a este fichero con el comando `chmod`:

```
usuario@nombreMaquina:~$ chmod +x obtieneDatos.sh

usuario@nombreMaquina:~$ ls -l obtieneDatos.sh
-rwxr-xr-x 1 usuario usuario 1101 de se 27 06:32 obtieneDatos.sh
```

Finalmente, habría que agregar una línea al fichero «/etc/crontab» que automatizaría la ejecución:

```
30 7    2 * *   usuario    /home/usuario/obtieneDatos.sh
```

A las 7:30 del día 2 de cada mes, en esta carpeta habrá un fichero «datos.csv» con los dos campos del catálogo. Por ejemplo:

```
usuario@nombreMaquina:~$ head -10 datos.csv
"APALOZ 5 MG COMPRIMIDOS EFG , 28 comprimidos ";"00000000"
"MEDIA LARGA (A-F) COMP NORMAL KURVAY-B (500 TALLA PEQUE#A ";"00000652"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH12 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH14 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH16 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH18 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH20 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH22 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH24 B10 ";"00000809"
"MEDIA LARGA (A-F) COMP NORMAL KURVAY-B (500 TALLA MEDIANA ";"00000652"
```

Observad que es posible que haya algún problema con la codificación de caracteres. Concretamente, no se lee adecuadamente el carácter «Ñ» (segunda línea, donde muestra «TALLA PEQUEÑA»). Habría que cambiar la codificación de los caracteres empleando el comando siguiente:

```
usuario@nombreMaquina:~$ iconv -t UTF-8 -f ISO-8859-1 datos.csv | head -10
"APALOZ 5 MG COMPRIMIDOS EFG , 28 comprimidos ";"00000000"
"MEDIA LARGA (A-F) COMP NORMAL KURVAY-B (500 TALLA PEQUEÑA ";"00000652"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH12 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH14 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH16 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH18 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH20 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH22 B10 ";"00000809"
"SONDA VESICAL SILICONA FOLEY SILICONA 100% CH24 B10 ";"00000809"
"MEDIA LARGA (A-F) COMP NORMAL KURVAY-B (500 TALLA MEDIANA ";"00000652"
```