
Caso práctico: acceso a recursos web

PID_00270629

Gerard Farràs Ballabriga

Tiempo mínimo de dedicación recomendado: 1 hora



**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
© Gerard Farràs Ballabriga
Todos los derechos reservados
© de esta edición, FUOC, 2020
Avda. Tibidabo, 39-43, 08035 Barcelona
Realización editorial: FUOC

Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea este eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares de los derechos.

Índice

1. Introducción.....	5
2. Primer caso.....	6
3. Segundo caso.....	11
Bibliografía.....	17

1. Introducción

Este caso práctico son en realidad dos casos. Hemos decidido ponerlos juntos porque tienen similitudes: ambos casos obtienen datos de internet (permiten descargar los datos directamente, sin la necesidad de emplear una API ni ningún otro código), tratan los datos descargados y les hacen una pequeña transformación.

El objetivo del primer subcaso (caso A) consiste en cruzar datos *opendata* del Ayuntamiento de Barcelona con la Viquipèdia en catalán. Se extraerá primero el nombre de los museos que aparecen en el fichero «Datos de museos y exposiciones de la ciudad de Barcelona» y, después, se hará una búsqueda en la Viquipèdia para obtener la descripción del museo, y también la URL con la entrada en la Viquipèdia, todo ello de manera automatizada.

Respecto al caso B, trata de descargar datos de «Equipamientos de Cataluña» del portal de datos abiertos de Cataluña, filtrarlos por un campo concreto y transformarlos generando un fichero HTML sencillo.

Estos casos servirán a los estudiantes como ejemplo para tratar ficheros con formatos diversos (concretamente, ficheros csv, json y xml), hacer un tratamiento sencillo y obtener un fichero resultante de dicha transformación.

Palabras clave

csv, json, xml, html, http requests, xmlstarlet, jq.

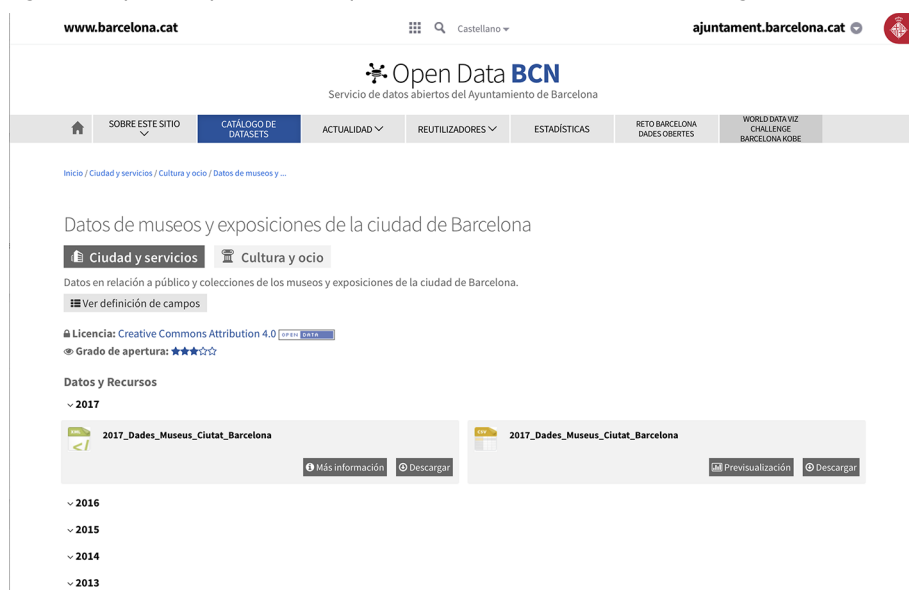
2. Primer caso

El primer paso consiste en acceder al sitio web:

opendata-ajuntament.barcelona.cat

Luego hay que descargar el fichero «Datos de museos y exposiciones de la ciudad de Barcelona» en formato .csv (obtendremos un fichero «2017_dades_museus.csv»).

Figura 1. Captura de pantalla en la que se ve el sitio web desde donde descargamos los datos



Con el comando siguiente podremos ver las dos primeras líneas del fichero. Observamos que la primera línea contiene la cabecera donde se indican los campos que hay presentes:

```
usuario@nombreMaquina:$ head -2 2017_dades_museus.csv
Any,Indicador,Equipament,Valor,Notes_Dades,Notes_Equipament,Codi_Districte,Nom_Districte,
Codi_Barri,Nom_Barri,Titularitat,TipusGeneral,TipusEquipament,Ambit,Latitud,Longitud
2017,Visitants,"Arts Santa Mònica. Centre de la Creativitat",94806,,01,"Ciutat Vella",01,
"el Raval","Pública no municipal","Museus i espais d'exposicions","Centres d'exposicions",
"Arts visuals",41.377227254,2.175597264
```

Este fichero separa sus campos con comas. Sin embargo, en el interior de algunos campos de texto también hay comas, por lo que no podemos separar los campos de este fichero empleando este delimitador. Se trata de un problema de separadores. Solucionaremos este asunto sustituyendo todas las comas que no están en medio de comillas:

```
usuario@nombreMaquina:$ cat 2017_dades_museus.csv | sed 's/,\[^\"]*\)/;/\1/g' >
2017_dades_museus-2.csv
```

El resultado del comando anterior se almacena en un segundo fichero («2017_dades_museus-2.csv»):

```
usuari@nombreMaquina:~$ head -2 2017_dades_museus-2.csv
Any;Indicador;Equipament;Valor;Notes_Dades;Notes_Equipament;Codi_Districte;Nom_Districte;
Codi_Barri;Nom_Barri;Titularitat;TipusGeneral;TipusEquipament;Àmbit;Latitud;Longitud
2017;Visitants;"Arts Santa Mònica. Centre de la Creativitat";94806;;;01;"Ciutat Vella";01;
"el Raval";"Pública no municipal";"Museus i espais d'exposicions";"Centres d'exposicions";
"Arts visuals";41.377227254;2.175597264
```

Ahora ya es posible procesar los campos de este fichero csv con comandos similares al siguiente:

```
usuari@nombreMaquina:~$ cut -d ";" -f3 2017_dades_museus-2.csv | uniq | sort | sed 's/"//g' >
museusbcn.txt
```

Cabe recordar que el comando `cut` extrae el campo indicado por el parámetro `-f` (en este caso, el tercer parámetro), en el que el carácter `;` actúa como delimitador. En la cabecera del fichero podemos ver que este tercer campo se refiere al nombre del equipamiento. El resultado de la salida se hace pasar por un *pipe* que eliminará los posibles valores repetidos (comando `uniq`), los ordenará (comando `sort`) y también eliminará las comillas (el comando `sed` sustituye las comillas por el carácter nulo). Todo ello se redirige a un tercer fichero que se denomina «museusbcn.txt».

```
usuari@nombreMaquina:~$ head -6 museusbcn.txt
Arts Santa Mònica. Centre de la Creativitat
Arxiu Fotogràfic de Barcelona
Arxiu Històric de la Ciutat de Barcelona
Basílica de la Sagrada Família
CaixaForum
Can Framis. Fundació Vila Casas
```

Ahora, a partir de este fichero, haremos una búsqueda en la Viquipèdia para cada museo. Para hacerlo, utilizaremos el comando `curl` seguido de la dirección. Mostramos previamente un par de ejemplos:

Nota

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

```
usuari@nombreMaquina:~$ curl "https://ca.wikipedia.org/w/api.php?action=opensearch&search=
caixaforum&limit=1&format=txt"
<!DOCTYPE html>
<html class="client-nojs" lang="ca" dir="ltr">
<head>
<meta charset="UTF-8"/>
<title>MediaWiki API result - Viquipèdia, l'enciclopèdia lliure</title>
<script>document.documentElement.className="client-js";RLCONF={"wgCanonicalNamespace":
"Special","wgCanonicalSpecialPageName":"ApiHelp","wgNamespaceNumber":-1,"wgPageName":
"Especial:ApiHelp","wgTitle":"ApiHelp","wgCurRevisionId":0,"wgRevisionId":0,
"wgArticleId":0,"wgIsArticle":!1,"wgIsRedirect":!1,"wgAction":"view","wgUserName":null,
"wgUserGroups":["*"],"wgCategories":[],"wgBreakFrames":!1,"wgPageContentLanguage":"ca",
"wgPageContentModel":"wikitext","wgSeparatorTransformTable":["\t","\t"],
"wgDigitTransformTable":["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":["","gener",
"febrer","març","abril","maig","juny","juliol","agost","setembre","octubre","novembre",
"desembre"],"wgMonthNamesShort":["","gen","feb","març","abr","maig","juny","jul","ago",
"set","oct","nov","des"],"wgRelevantPageName":"Especial:ApiHelp","wgRelevantArticleId":0,
"wgRequestId":"XY@D-QpAIDQAAGTDNmUAAACX","wgCSPNonce":!1,"wgIsProbablyEditable":!1,
"wgRelevantPageIsProbablyEditable":!1,
```

Con el comando anterior llamamos a la API de la Viquipèdia haciendo una búsqueda mediante el término «caixaforum» y limitando la salida a solamente una entrada. El formato resultante será el código html de la página web, código poco tratable. De hecho, se ha truncado el resultado del comando anterior para que el texto sea más legible.

En cambio, si lo especificamos como formato json, se obtiene información mucho más tratable. Por ejemplo:

```
usuario@nombreMaquina:~$ curl "https://ca.wikipedia.org/w/api.php?action=opensearch&search=caixaforum&limit=1&format=json"
["caixaforum",["CaixaForum Barcelona"],["CaixaForum Barcelona \u00e9s un centre cultural gestionat per \u201dla Caixa\u201d dins la seva Obra Social de l\u2019entitat."],["https://ca.wikipedia.org/wiki/CaixaForum_Barcelona"]]
```

Dado que deseamos hacer una búsqueda para cada uno de los nombres de los equipamientos, podemos automatizar la tarea utilizando un *script*. En el momento de escribir este texto, el número total de líneas del fichero anterior es de:

```
usuario@nombreMaquina:~$ wc -l museusbcn.txt
56 museusbcn.txt
```

Hemos nombrado al fichero que contiene el código con el guion que hará la búsqueda en la Viquipèdia «llamadaWiki.sh» y lo escribimos a continuación:

```
#!/bin/bash
# Script que, a partir del fichero $input
# hará una búsqueda en la Viquipèdia y obtendrá
# un fichero json con el resultado de la búsqueda.

input="museusbcn.txt"
i=1

# Este while tratará línea a línea
while IFS= read -r line
do
  #Reemplazamos los espacios por %20, el espacio en formato html.
  cerca=echo $line | sed 's/ /%20/g'

  #Se realiza la llamada a la API de la Viquipèdia.
  #El resultado se almacenará en un fichero json.
  curl -a -s "https://ca.wikipedia.org/w/api.php?action=opensearch&search=\"$cerca\"&limit=1&format=json" -o $i.json
  let i++
done < "$input"
```

Recordemos darle permiso de ejecución al *script* anterior con el comando:

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

y ejecutar el *script* indicando su ruta (dado que la carpeta actual no reside en la variable de entorno *PATH*, la *shell* no sabría dónde buscar el comando).

```
usuario@nombreMaquina:~$ ./llamadaWiki.sh
```


Figura 2. Captura de pantalla de un terminal en la que se observa el contenido del *script* (con el comando `more`), cómo se otorga el permiso de ejecución (con el comando `chmod`) y, finalmente, cómo se ejecuta



```

gerard@Zeus: ~/Proyectos/uocDataScience/casoPractico2
Archivo Editar Ver Buscar Terminal Ayuda
gerard@Zeus:~/Proyectos/uocDataScience/casoPractico2$ more llanadaWiki.sh
#!/bin/bash
input="museusbcn.txt"
i=1
while IFS= read -r line
do
    cerca="echo $line | sed 's/ /%20/g'"
    curl -a -s "https://ca.wikipedia.org/w/api.php?action=opensearch&search=$cerca&limit=1&format=json" -o $i.json
    let i++
done < "$input"
gerard@Zeus:~/Proyectos/uocDataScience/casoPractico2$ chmod +x llanadaWiki.sh
gerard@Zeus:~/Proyectos/uocDataScience/casoPractico2$ ./llanadaWiki.sh
gerard@Zeus:~/Proyectos/uocDataScience/casoPractico2$

```

En este paso, se dispondrá de un montón de ficheros `.json` (idealmente, uno para cada equipamiento), con el resultado de la búsqueda en la Viquipèdia. Per ejemplo:

```

usuario@nombreMaquina:~$ more 4.json
["Bas\u00edlica de la Sagrada Fam\u00edlia",["Bas\u00edlica de la Sagrada Fam\u00edlia"],["https://ca.wikipedia.org/wiki/Bas%C3%ADlica_de_la_Sagrada_Fam%C3%ADlia"]]

```

O:

```

usuario@nombreMaquina:~$ more 7.json
["Casa Batll\u00f3",["Casa Batll\u00f3"],["La Casa Batll\u00f3 \u00e9s un edifici dissenyat per l'arquitecte Antoni Gaud\u00ed, m\u00e9s oxim representant del modernisme catal\u00e0, entre 1904 i 1907. Est\u00e0 situat al n\u00famero 43 del passeig de Gr\u00e0cia de Barcelona, l'ampla avinguda que travessa el barri modernista de l'Eixample."],["https://ca.wikipedia.org/wiki/Casa_Batll%C3%B3"]]

```

Estos ficheros en formato json se pueden procesar usando el comando `jq`. Este comando no acostumbra a estar en las instalaciones por defecto, así que es posible que haya que instalarlo utilizando:

```

usuario@nombreMaquina:~$ sudo apt-get install jq

```

Con `jq` es posible extraer los campos del fichero json que se deseen. Por ejemplo:

```

usuario@nombreMaquina:~$ more 5.json
["CaixaForum",["CaixaForum"],["CaixaForum \u00e9s el nom comercial dels diferents centres culturals que la Fundaci\u00f3 \"La Caixa\" t\u00e9 a diverses ciutats de l'estat espanyol."],["https://ca.wikipedia.org/wiki/CaixaForum"]]

```

Con el comando `jq` podemos indicar exactamente qué campo deseamos ver. En este caso, el tercero contiene el enlace:

```

usuario@nombreMaquina:~$ jq .[3] 5.json
[
  "https://ca.wikipedia.org/wiki/CaixaForum"
]

```

Utilizando el parámetro `-c`, se muestra un resultado de salida más compacto:

```

usuario@nombreMaquina:~$ jq -c .[3] 5.json
https://ca.wikipedia.org/wiki/CaixaForum

```

```
["https://ca.wikipedia.org/wiki/CaixaForum"]
```

Realizaremos esta tarea de manera masiva en todos los ficheros json con el *script* siguiente, al cual hemos titulado «*extrae.sh*»:

```
#!/bin/bash
for f in *.json
do
  jq -c .[3,2] $f
done
```

Recordemos nuevamente otorgar permisos de ejecución y ejecutarlos indicando la ruta:

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

Tras su ejecución, aparecerá algo similar a lo siguiente:

```
["https://ca.wikipedia.org/wiki/Casa_Batl1%C3%B3"]
["La Casa Batlló és un edifici dissenyat per l'arquitecte Antoni Gaudí, màxim representant del modernisme català, entre 1904 i 1907. Està situat al número 43 del passeig de Gràcia de Barcelona, l'ampla avinguda que travessa el barri modernista de l'Eixample."]

["https://ca.wikipedia.org/wiki/Casa_Museu_Gaud%C3%AD"]
["La Casa Museu Gaudí, situada dintre del recinte del Parc Güell, a Barcelona, va ser la residència d'Antoni Gaudí durant gairebé 20 anys, de 1906 al final de 1925. El 28 de setembre de 1963 es va inaugurar com a museu i actualment acull una col·lecció de mobles i objectes dissenyats per l'arquitecte."]

["https://ca.wikipedia.org/wiki/Castell_de_Montju%C3%AFc"]
["El Castell de Montjuïc de Barcelona fou una fortalesa militar i, després de la guerra civil, va ser un museu militar."]
```

En este caso se ha ilustrado, por un lado, el tratamiento de un fichero csv, luego se han realizado búsquedas en la Viquipèdia usando su API y obteniendo ficheros json, que también se han tratado por medio del terminal. El resultado consiste en un enlace de datos entre el fichero de equipamientos del portal de datos abiertos del Ayuntamiento de Barcelona con las búsquedas en la Viquipèdia.

3. Segundo caso

Recordemos que el objetivo de este segundo caso consiste en descargar del portal de datos abiertos el listado de equipamientos de Cataluña, filtrar los datos para un campo concreto y transformarlos en un fichero html sencillo.

«[Esta] base de datos de equipamientos integra, en una única fuente, los datos descriptivos de equipamientos de Cataluña, tanto de titularidad pública como privada: sedes de los ayuntamientos, centros educativos, sociosanitarios, culturales, instalaciones deportivas, de transporte, etc. La solución consolida en una estructura común y normalizada los datos que están en diferentes bases de datos y que son gestionadas por los diferentes departamentos y organismos de la Generalitat de Catalunya.»

Fuente: «Equipaments de Catalunya» [en línea]. Datos abiertos Cataluña. Disponible en: analisi.transparenciacatalunya.cat/es/Urbanisme-infraestructures/Equipaments-de-Catalunya/8gmd-gz7i

El primer paso consiste en descargar el fichero con el listado de equipamientos en formato xml del sitio web. Esta operación se podría realizar con el navegador de internet o, si lo preferimos, desde el terminal con los comandos `wget` o `curl`.

```
usuario@nombreMaquina:~$ wget -O equipaments.xml https://analisi.transparenciacatalunya.cat/api/views/8gmd-gz7i/rows.xml?accessType=DOWNLOAD
```

En el caso de que el comando anterior no funcione adecuadamente, es mejor descargar los datos directamente mediante el navegador visitando la dirección analisi.transparenciacatalunya.cat/es/ y buscar el fichero «Equipaments de Catalunya».

Figura 3. Captura de pantalla del portal de «dades obertes Catalunya»

The screenshot shows the search results for 'Equipaments de Catalunya' on the 'dades obertes Catalunya' portal. The search bar contains the text 'Equipaments de Catalunya'. The results are displayed in a list format, with each result showing a title, a brief description, the date it was last updated, the number of views, and a link to the API documents. The results are ordered by 'Más relevante'.

Result Title	Category	Updated	Views
Equipaments de Catalunya	Urbanisme e infraestructuras	27 de enero de 2020	5.095
Fets succeïts l'1 d'octubre de 2017	Memoria	18 de diciembre de 2019	372
Registre de llicències comercials	Comerç	13 de diciembre de 2019	1.049
Indicadors territorials de l'Observatori del Territori	Urbanisme e infraestructuras	28 de octubre de 2019	673

Podemos obtener más información de este fichero con los comandos siguientes:

```

usuario@nombreMaquina:~$ file equipaments.xml
equipaments.xml: UTF-8 Unicode text, with very long lines

usuario@nombreMaquina:~$ wc -l equipaments.xml
1451 equipaments.xml

```

Recordemos que este fichero está en formato xml, así que para tratarlo, usaremos el comando `xmlstarlet`.

Este fichero de equipamientos contiene, para cada fila, campos diversos (en concreto, 23 atributos). Entre otros, un identificador, un alias, un nombre, una categoría, la dirección (con un campo `poblacio` y otro `comarca`), teléfonos, sitio web, etc.

Es posible filtrar la información utilizando diversos atributos. Por ejemplo, el comando siguiente filtra por el campo `comarca` y muestra solo el campo `nom` de los registros que están en la comarca de Osona. Mostramos a continuación solo cinco registros para no ocupar mucho espacio:

```

usuario@nombreMaquina:~$ xmlstarlet sel -T -t -m "/response/row/row[comarca='Osona']" -v "nom"
-n equipaments.xml | head -5
Jutjat de Pau de Torelló
Punt TIC Centelles
Museu Claretia
Museu del Vidre Bac de Roda
Biblioteca de Lluçà

```

A continuación, mostramos los campos `nom` y `via`, separados por el caracter `;` como delimitador y filtramos para los equipamientos que estén en la ciudad de Solsona:

```
usuario@nombreMaquina:~$ xmlstarlet sel -T -t -m "/response/row/row[poblacio='Solsona']" -v
"concat(nom,',';','via)" -n equipaments.xml > equipSolsona.txt
```

Este fichero contiene solo los campos `nom` y `via` del xml original separados por un `;`:

```
usuario@nombreMaquina:~$ head -10 equipSolsona.txt
CFA Solsonès; c. Dominics, 12
Oficina Comarcal del Solsonès; Avinguda de la Mare de Déu del Claustre, 18
Jutjat de la Instància i Instrucció núm. 1; Llobera, 1
Punt TIC - Biblioteca Carles Morató;de Mirabalda
Consulta Perifèrica Solsona; PL GUITART, 1
DEIXALLERIA DE SOLSONA; PTDA SANT HONORAT S/N
Col·lecció Antic Museu del Ganivet; C. de Bassella, 1. Palau Llobera
Servei d'informació i atenció a les dones (SIAD) del Solsonès; Dominics
SERVEI COMARCAL DE JOVENTUT DEL SOLSONÈS;dels Dominics
OFICINA DE TURISME DEL SOLSONÈS; Ctra. Bassella , N° 1
```

A continuación, generaremos una página web en formato HTML sencilla a partir de este fichero.

Recordemos que un fichero html consiste en un fichero de texto plano con la estructura siguiente:

```
<html>
<head>Cabecera del sitio web</head>
<body>
Contenido del sitio web.
</body>
</html>
```

La acción que realizará el *script* consiste en generar por pantalla estos marcadores en html y agregar una fila en una tabla html para cada equipamiento. Hemos comentado el código del *script* con los comentarios precedidos por el caracter almohadilla (`#`). El *script* se llama `generaweb.sh`:

```
#!/bin/bash
# Este script generará un HTML
# que contendrá una tabla en la que cada fila contendrá un equipamiento.

# Mostramos los marcadores HTML iniciales.
echo "<html>"
echo "<body>"
echo "<table border='1'>"

# Lectura del fichero obtenido anteriormente.
# Cada fila contiene un equipamiento. Los campos están separados por ;
# El primer campo contiene el nombre. El segundo la vía.

input="equipSolsona.txt"
i=1
while IFS= read -r line
do
  nom='echo $line | cut -d ";" -f1'
  via='echo $line | cut -d ";" -f2'
  echo "<tr><td>$nom</td><td>$via</td></tr>"
done < "$input"
```

```
# Mostramos los marcadores HTML finales.
echo "</table>"
echo "</body>"
echo "</html>"
```

Recordemos establecer los permisos de ejecución en este *script*:

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

Figura 4. Captura de pantalla en la que se ve el código del *script* (comando `more`), cómo se otorgan los permisos de ejecución (comando `chmod +x`) y cómo se ejecuta el *script*, redirigiendo la salida a un fichero html

```
gerard@Zeus: ~/Proyectos/uocDataScience/casoPractico2
Archivo Editar Ver Buscar Terminal Ayuda
gerard@Zeus:~/Proyectos/uocDataScience/casoPractico2$ more generaWeb.sh
#!/bin/bash
echo "<html>"
echo "<body>"
echo "<table border='1'>"
input="equipSolsona.txt"
i=1
while IFS= read -r line
do
  nom="echo $line | cut -d ";" -f1`
  via="echo $line | cut -d ";" -f2`
  echo "<tr><td>$nom</td><td>$via</td></tr>"
done < "$input"
echo "</table>"
echo "</body>"
echo "</html>"
gerard@Zeus:~/Proyectos/uocDataScience/casoPractico2$ ./generaWeb.sh > output.html
gerard@Zeus:~/Proyectos/uocDataScience/casoPractico2$ firefox output.html
```

Y ejecutarlo con el comando redirigiendo la salida a un fichero `output.html`:

```
usuario@nombreMaquina:~$ ./generaWeb.sh > output.html
usuario@nombreMaquina:~$ head -10 output.html
<html>
<body>
<table border='1'>
<tr><td>CFA Solsonès</td><td> c. Dominics, 12</td></tr>
<tr><td>Oficina Comarcal del Solsonès</td><td> Avinguda de la Mare de Déu del Claustre, 18</td></tr>
<tr><td>Jutjat de la Instància i Instrucció núm. 1</td><td> Llobera, 1</td></tr>
<tr><td>Punt TIC - Biblioteca Carles Morató</td><td>de Mirabalda</td></tr>
<tr><td>Consulta Perifèrica Solsona</td><td> PL GUITART, 1</td></tr>
<tr><td>DEIXALLERIA DE SOLSONA</td><td> PTDA SANT HONORAT S/N</td></tr>
<tr><td>Col·lecció Antic Museu del Ganivet</td><td> C. de Bassella, 1. Palau Llobera</td></tr>
```

Podríamos visualizar esta página html usando un navegador. Por ejemplo:

```
usuario@nombreMaquina:~$ firefox output.html
```

Figura 5. Fichero html generado con el *script* `generaWeb.sh` que contiene una tabla en formato html en la que cada fila es un equipamiento de la ciudad de Solsona. Cada fila contiene dos columnas: la primera con el nombre del equipamiento y, la segunda, con su ubicación

CFA Solsonès	c. Dominics, 12
Oficina Comarcal del Solsonès	Avinguda de la Mare de Déu del Claustre, 18
Jutjat de 1a Instància i Instrucció núm. 1	Llobera, 1
Punt TIC - Biblioteca Carles Morató	de Mirabalda
Consulta Perifèrica Solsona	PL GUITART, 1
DEIXALLERIA DE SOLSONA	PTDA SANT HONORAT S/N
Col·lecció Antic Museu del Ganivet	C. de Bassella, 1. Palau Llobera
Servei d'informació i atenció a les dones (SIAD) del Solsonès	Dominics
SERVEI COMARCAL DE JOVENTUT DEL SOLSONÈS	dels Dominics
OFICINA DE TURISME DEL SOLSONÈS	Ctra. Bassella , Nº 1
Biblioteca del Museu Diocesà i Comarcal de Solsona	Pl. Palau, 1
Biblioteca Carles Morató	C/ Mirabalda, 6
Arxiu Comarcal del Solsonès. Biblioteca Auxiliar	C/ Dominics, 14
Ajut a domicili sol del solsonès	C. Josep Maria de Segarra, 3, bxs
Centre de dia Hospital Pere Màrtir Colomés	Pça. Antoni Guitart, 1
Centre de desenvolupament infantil i atenció precoç del Solsonès	C. Carles Morató, 4, bxs
Extranjeros Unidos en Solsona	C. Pedraforca, 19, 1er 1º (Residència Delta)
Programa d'atenció a la infància i adolescència en risc social	C. Castell, 20
Programa d'atenció als menors amb risc social del Consell Comarcal del Solsonès	dels Dominics
Llar residència Amisol	Ctr. de Torà, 54, bxs
Servei d'intervenció socioeducativa "El 6" Cal Porrer	C. Llobera, 14, bxs
Residència Hospital Pere Màrtir Colomés	Pça. Antoni Guitart, 1
Servei de suport a l'autonomia a la pròpia llar Amisol	de Santa Llúcia
Servei d'atenció domiciliària Hospital Pere Màrtir Colomés	Pça. Antoni Guitart, 2
PAVEL·LO MUNICIPAL D'ESPORTS	C. Salvador Espriu, 8

Bibliografía

«jq Manual (development version)». Disponible en: <https://stedolan.github.io/jq/manual/>

«XmlStarlet Command Line XML Toolkit User's Guide». Disponible en: xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html

