

Implementación del comando `biblio` para la consulta del catálogo de libros y préstamos del CRyA utilizando la información de la base de datos de la aplicación ALEPH.

Leonardo Arroyo¹

¹Instituto de Radioastronomía y Astrofísica, UNAM, Campus Morelia.

Resumen: En este documento se describe el uso del comando `biblio` y la manera como se implementó para ser ejecutado en terminales del sistema operativo UNIX (linux, en la mayoría de los casos) en el Centro de Radioastronomía y Astrofísica de la UNAM, en el campus Morelia. Este comando permite consultar el catálogo local de libros y préstamos de manera ágil y sencilla, haciendo uso de la información actualizada contenida en la base de datos de la aplicación para bibliotecas ALEPH¹. A diferencia de ALEPH, la información mostrada por el comando `biblio` es breve y de acceso rápido, no requiere del uso de un navegador, ni de recordar el URL del catálogo de libros, lo cual convierte al comando `biblio` en una herramienta sumamente práctica. En el presente reporte técnico se explica el funcionamiento del comando `biblio`, así como la manera de implementarlo dentro del CRyA y, en principio, puede implementarse en cualquier centro, instituto o facultad de la UNAM. La utilidad en cada caso dependerá de qué tan intenso sea el uso del sistema operativo UNIX o sus derivados (linux, mac OS) entre los usuarios. El comando está implementado en el lenguaje de programación python.

1 INTRODUCCIÓN

La mayoría de los investigadores, alumnos y técnicos académicos del Centro de Radioastronomía y Astrofísica (CRyA) utilizan el sistema operativo Linux como herramienta de trabajo cotidiana. Este sistema operativo permite la interacción del usuario con la computadora mediante una serie de comandos, la mayoría de los cuales pueden ejecutarse en una terminal, que permiten realizar funciones específicas para interactuar con los recursos de la computadora.

Por su parte, el acceso a la base de datos de las bibliotecas y acervos bibliográficos de la UNAM se da a través del portal web de cada instituto, centro o facultad. Para consultar la existencia de algún título es

¹ Sistema Bibliotecario utilizado por la DGB para la catalogación y registro de circulación del material bibliográfico de la UNAM.

entonces necesario abrir el navegador, ir a la página del acervo bibliográfico en cuestión, ingresar el o los campos de interés, y esperar la respuesta del servidor de ALEPH para ver el estado de un libro. Este proceso puede tardar entre decenas de segundos y algunos minutos, típicamente.

Dado que la población en el CRyA suele utilizar en estaciones de trabajo que tienen variantes del sistema operativo UNIX (linux, y en menor porcentaje, macOS), en donde a través de una terminal se pueden ejecutar comandos, creímos conveniente desarrollar una herramienta que permitiera una consulta más ágil y rápida al catálogo local de libros y préstamos, sin la necesidad de tener que acceder a un navegador para realizar una consulta.

Así, con el comando `biblio` tan solo es necesario escribir el nombre del comando en la terminal, seguido del término de búsqueda, el cual puede ser una palabra que forme parte del título, del nombre del autor o bien del usuario que tiene en préstamo un libro. El comando `biblio` accede a los registros actualizados de nuestro acervo, filtra la información y se la muestra al usuario en la terminal donde se ejecutó el comando, ordenada alfabéticamente por título del libro.

2 OBJETIVO

Crear un comando de terminal Unix/Linux para consultar el catálogo local de libros del CRyA. Este comando tendrá acceso directamente a la base de datos de la aplicación ALEPH, permitiendo una consulta rápida y sintetizada del material bibliográfico.

3 ARQUITECTURA DE LA APLICACIÓN

Para la implementación del comando `biblio` en el Centro de Radioastronomía y Astrofísica deben considerarse los siguientes factores:

- 1) La base de datos del sistema ALEPH está implementada en Oracle.
- 2) Esta base de datos se localiza en un servidor dentro de la red del Centro de Matemáticas en cual se localizan otras bases de datos, como las del Centro de Matemáticas, Centro de Investigaciones en Ecosistemas, el Instituto de Geofísica.
- 3) El CRyA no cuenta con libre acceso a la base de datos, cualquier cambio o implementación debe ser efectuada por un administrador de la Dirección General de Bibliotecas (DGB).

Por esto, el comando `biblio` en realidad es solamente la tercera de tres partes independientes. La primera consiste en generar diariamente el archivo de salida de SQL con la información de todos los títulos que tenemos. Esto se hace diariamente, y es un script que se les pidió a los administradores del sistema de la DGB. La segunda consiste en traer dicho archivo generado al CRyA, leerlo y cambiarle el formato para que la búsqueda sea más ágil. La tercera, finalmente, es el comando `biblio`, que lee el archivo generado en el segundo paso, y realizar la búsqueda. La figura 3.1 muestra un diagrama con las tres partes descritas y su interacción.

Proceso 1: Consulta SQL diaria en base de datos ALEPH, se genera un archivo de texto con el resultado.

Proceso 2: Lectura de archivo remoto y procesamiento del mismo para formato de lectura del script *biblio*

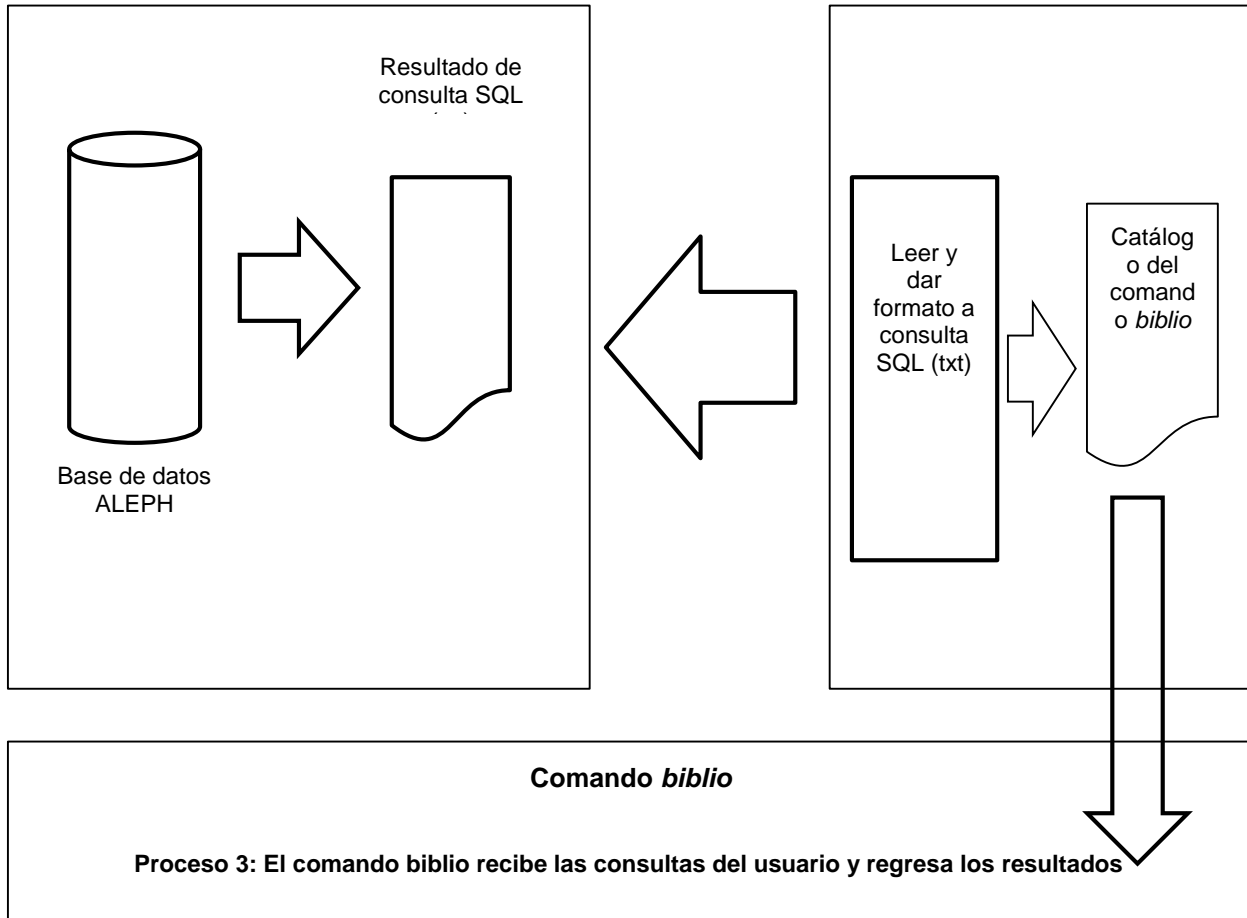


figura 3.1. Arquitectura del comando biblio.

4 IMPLEMENTACIÓN

La implementación de este comando se realizó en el lenguaje python 2.7, el cual brinda características de flexibilidad y manejo de datos de una manera simplificada, el cual interactúa con una amplia variedad de aplicaciones y sistemas operativos. Para la consulta de los datos de ALEPH, se implementan consultas en el lenguaje SQL, el cual es ejecutado sobre la herramienta SQL*Plus², incluida con el manejador de bases de datos ORACLE.

² Es un programa de línea de comandos de Oracle que puede ejecutar comandos SQL y PL/SQL de forma interactiva o mediante un script.

Para el desarrollo se siguió el diseño conceptual que se muestra en la figura 1.1. Desarrollando uno a uno los procesos descritos en el. Los requerimientos mínimos para la implementación son los siguientes:

Servidor

- Sistema operativo Unix/Linux
- 256 MB RAM
- Paquete SSH instalado
- Interprete python 2.7 +
- Cliente Oracle

A continuación se hace una descripción detallada de cada uno de los procesos que involucran el desarrollo y funcionamiento del comando `biblio`.

4.1 Obtención de datos mediante la consulta SQL en la base de datos de ALEPH.

Debido a las restricciones de acceso y localización del servidor ALEPH, se escribió un pequeño *script* en el lenguaje SQL, el cual se ejecuta todos los días a las 5:00 A.M. Mediante esta consulta se genera el archivo de texto `material.lst`. El listado de este script se encuentra en el apéndice A, y los datos que arroja están organizados por:

```
AUTOR,          CLASIFICACION,  TITULOS,          NUM_SISTE,        NUM_ADQ,        NOMBRE,          FECHA_PR,        PRESTADO
```

La consulta a la base de datos se programó como un evento `cron` del sistema operativo, el cual se ejecuta todos los días a las 5:00 hrs. Dado que nosotros no tenemos acceso al sistema ALEPH, el comando `cron` debe ser programado por gente del departamento de sistemas de la DGB.

Una vez generando diariamente el archivo de texto `material.lst` con la información del catálogo, el archivo es traído al CRyA para poder ser utilizado mediante el comando `biblio`.

4.2 Manejo y preparación de los datos

Los datos obtenidos por la consulta periódica a la base de datos Oracle (es decir, el archivo `material.lst`) se localizan en un servidor fuera de la red del CRyA, por lo que es necesario traerlos para trabajar de manera local y así evitar cualquier problema de comunicación que nos deje sin el acceso a estos.

Para obtener una copia de los datos se genera un proceso en el `cron` del servidor local para que cada día, unos minutos después de haberse generado la lista con los datos de ALEPH, se cree una copia local. Esto se efectúa con el siguiente comando:

```
sftp larroyo@132.248.196.36:/directorio_en_DGB/material.lst /directoro_en_el_CRyA/
```

Una vez traído el archivo `material.lst`, éste es leído por un script en lenguaje python, a fin de transformar la información del archivo original en un arreglo de tuplas. El arreglo tiene una estructura lógica similar a la figura 2.1

n	AUTHOR	CLASIFICA	TITULOS	NUM SIST	NUM ADO	NOMBRE	PRESTADO
1	AUTHOR	CLASIFICA	TITULOS	NUM SIST	NUM ADO	NOMBRE	PRESTADO
2	AUTHOR	CLASIFICA	TITULOS	NUM SIST	NUM ADO	NOMBRE	PRESTADO
3	AUTHOR	CLASIFICA	TITULOS	NUM SIST	NUM ADO	NOMBRE	PRESTADO
4	AUTHOR	CLASIFICA	TITULOS	NUM SIST	NUM ADO	NOMBRE	PRESTADO
n	AUTHOR	CLASIFICA	TITULOS	NUM SIST	NUM ADO	NOMBRE	PRESTADO

figura 2.1 Esquema lógico del arreglo de tuplas de material.lst

Este arreglo se escribe en un archivo nuevo, el cual será finalmente consultado por el comando `biblio`. El listado de este nuevo script se encuentra en el apéndice B.

4.3 Consultas con el comando `biblio`.

Para realizar una consulta mediante el comando `biblio` se obtiene la cadena de consulta proporcionada por el usuario cuando ingresa en la terminal el comando como se muestra en el siguiente ejemplo:

```
usuario@server$ biblio <cadena de búsqueda>
```

Como ya se había mencionado antes la cadena de búsqueda puede contener palabras que hagan referencia a:

- 1) Parte del título del libro que se busca.
- 2) Parte o nombre del autor.
- 3) La colocación de un libro.
- 4) Número de adquisición o sistema (si se conoce).
- 5) Parte o nombre completo de un usuario para conocer cuáles libros se tienen en préstamo.
- 6) Una consulta en blanco, la cual devolverá el listado completo del material en existencia.

La consulta se lleva a cabo comparando la cadena de búsqueda con cada uno de los campos en cada una de las tuplas del arreglo, cuando hay alguna coincidencia, la tupla completa se envía a impresión en pantalla. En caso de no encontrar coincidencias el comando enviará un mensaje indicando que la búsqueda no dió resultados. El script que lleva a cabo esta búsqueda se presenta en el Apéndice C.

5 RESULTADOS Y PRUEBAS

El usuario puede acceder al comando mediante una terminal de comandos, teclear la cadena de búsqueda, que puede corresponder a un autor, título o parte del título de un libro, clasificación e incluso número de adquisición.

Con el fin de mostrar los resultados de este comando se presentan a continuación los siguientes ejemplos de uso:

Ejemplo 1:

Búsqueda de libros que en alguno de sus campos contenga la palabra “astrometry”

```

larroyo@acervo: ~
[larroyo@fenix ~]$ biblio astrometry
-----
1
Titulo   : (1017) A giant step, from milli- to micro-arcsecond astrometry :
Autor    : International Astronomical Union. Symposium (248 : 2007 : Shanghai, China)
Editorial : ----
ISBN     : ----
Colocaci+n: QB807 I57 2007
Usuario  : Disponible

2
Titulo   : (2112) Astrometry for astrophysics :
Autor    :
Editorial : ----
ISBN     : ----
Colocaci+n: QB807 A76
Usuario  : Disponible

3
Titulo   : (462) Astrometry in the age of the next generation of large telescopes
Autor    :
Editorial : ----
ISBN     : ----
Colocaci+n: QB807 A77
Usuario  : Disponible

4
Titulo   : (422) Astrometry of fundamental catalogues :
Autor    : Walter, Hans George, 1928-
Editorial : ----
ISBN     : ----
Colocaci+n: QB807 W35
Usuario  : Disponible

5
Titulo   : (403) Fundamentals of astrometry
Autor    : Kovalevsky, Jean autor
Editorial : ----
ISBN     : ----
Colocaci+n: QB807 K686
Usuario  : Disponible

6
Titulo   : (1825) Vectorial astrometry
Autor    : Murray, C. Andrew
Editorial : ----
ISBN     : ----
Colocaci+n: QB807 M87
Usuario  : Disponible
-----
[larroyo@fenix ~]$

```

Imagen 5.1. Ejemplo de uso del comando biblio en una terminal de Linux “ubuntu”

Ejemplo 2:

Búsqueda de libros que tenga en préstamo el usuario de apellido “gazol”

```

larroyo@acervo: ~
[larroyo@fenix ~]$ biblio gazol
-----
1
Titulo   : (1735) Accretion processes in star formation
Autor    : Hartmann, Lee, 1950-
Editorial : ----
ISBN     : ----
Colocaci+n: QB806 H37 2009
Usuario  : Gazol, Adriana - 20141128

2
Titulo   : (1895) Accretion processes in star formation
Autor    : Hartmann, Lee, 1950-
Editorial : ----
ISBN     : ----
Colocaci+n: QB806 H37 2009
Usuario  : Gazol, Adriana - 20140319

3
Titulo   : (1733) Physical processes in the interstellar medium
Autor    : Spitzer, Lyman, 1914-1997 autor
Editorial : ----
ISBN     : ----
Colocaci+n: QB790 S67 2004
Usuario  : Gazol, Adriana - 20141128

4
Titulo   : (471) The black hole at the center of the Milky Way
Autor    : Eckart, Andreas
Editorial : ----
ISBN     : ----
Colocaci+n: QB843.B55 E35
Usuario  : Gazol, Adriana - 20130425

5
Titulo   : (1829) Theoretical astrophysics :
Autor    : Padmanabhan, T. (Thanu), 1957
Editorial : ----
ISBN     : ----
Colocaci+n: QB461 P33
Usuario  : Gazol, Adriana - 20140319
-----
[larroyo@fenix ~]$

```

Imagen 5.2. Ejemplo de búsqueda con el comando biblio en una terminal de Linux “ubuntu”

6 CONCLUSIONES

La estructura del comando `biblio` permite mostrar de manera resumida un listado de libros y préstamos del catálogo local de la institución. Para trabajos futuros, la búsqueda se podría hacer más precisa e incluir parámetros al comando para especificar el campo de búsqueda. De ser necesario podría aumentarse el número de campos a la consulta periódica de la base de datos, de tal forma que cualquier dato almacenado en el sistema ALEPH podría consultarse mediante esta herramienta o alguna similar.

Si bien la información que el comando `biblio` proporciona es mínima, comparado con una consulta al OPAC de ALEPH, esta consulta es bien recibida por los usuarios ya que proporciona de manera ágil información concisa de lo que se busca. Dado el entorno de uso, por el momento esta información reducida es suficiente para satisfacer las necesidades de los usuarios.

Esta implementación podría utilizarse en otros centros haciendo las modificaciones correspondientes a la consulta de SQL para Oracle. Ya que el script de python lee un archivo con ciertas características, las cuales son definidas al momento de crear el archivo con la consulta SQL generada, el script utilizado en el CRYA funcionaría en algún otro centro.

7 REFERENCIAS

[1] Russel, J. (2003). *PL/SQL User's Guide and Reference*, 10g. Estados Unidos. Oracle Corporation.

[2] *Documentación oficial Python*. (2014). Python.org. <https://docs.python.org/2/index.html>

APÉNDICE A. SCRIPT EN SQL QUE SE EJECUTA EN EL SERVIDOR ALEPH PARA OBTENER EL LISTADO DE LOS DATOS PARA EL COMANDO `BIBLIO`.

```

SET ECHO OFF
SET FEEDBACK OFF
SET PAUSE OFF
SET PAGESIZE 10000
SET NEWPAGE NONE
SET LINESIZE 200
SET TERMOUT OFF
TITLE 'LISTADO MATERIAL PRESTADO'

rem SET VERIFY OFF
column c1 format a30 HEADING "AUTOR"
column c2 format a15 HEADING "CLASIFICACI\323N"
column c3 format a30 HEADING "TITULOS"
column c4 format a09 HEADING "NUM_SISTEMA"
column c5 format a09 HEADING "NUM_ADQ"
column c6 format a15 HEADING "NOMBRE"

```



```
column c7 format a08 HEADING "FECHA_PRESTA"
```

```
Spool /opt/larroyo/material.lst
select L3901.Z13.Z13_AUTHOR c1, L3901.Z13.Z13_CALL_NO c2, L3901.Z13.Z13_TITLE c3, z13_rec_key c4,
L3950.Z30.Z30_BARCODE c5, substr(L1050.z303.Z303_NAME,1,30) c6 , to_char(Z36_LOAN_DATE) c7,
case when L3950.Z36.Z36_REC_KEY is not null then 'Prestado' end as Prestados from
L3901.z13, l3950.z30 left outer join L3950.z36 on L3950.Z30.Z30_REC_KEY=L3950.Z36.Z36_REC_KEY
LEFT OUTER JOIN L3950.Z303 ON(L3950.z303.Z303_REC_KEY = L3950.z36.Z36_ID)
where
L3901.Z13.Z13_REC_KEY=substr(L3950.Z30.Z30_REC_KEY,1,9);
```

```
Spool off;
quit;
```

APÉNDICE B. SCRIPT DE CONVERSIÓN DE DATOS DE ALEPH, A FORMATO LEGIBLE POR EL COMANDO BIBLIO.

```
# -*- coding: utf-8 -*-
import sys;
import csv
reload(sys);
sys.setdefaultencoding("utf8")

archivo = "/home/larroyo/reporte/material.lst"
archi=open(archivo,'r')
lineas=archi.readlines()[2:]
archi.close()

autor = clasif = titulo = n_sist = n_adq = nombre = fecha = observacion = ''
data = libros = []

def eliminaEspacios(cadena):
    cadena = cadena.rstrip()
    cadena = cadena.lstrip()
    return cadena

def eliminaTextoAutor(cadena):
    cadena = cadena.replace(', autor', '')
    cadena = cadena.replace(',autor', '')
    cadena = cadena.replace(', -autor', '')
    cadena = cadena.replace(', - autor', '')
    cadena = cadena.replace('-autor', '')
    cadena = cadena.replace('- autor', '')
    cadena = cadena.replace('-',autor', '')
    cadena = cadena.replace('-', autor', '')
    return cadena

def eliminaTextoAdd(cadena):
    cadena = cadena.replace('/', '')
    return eliminaEspacios(cadena)

for linea in lineas:
    if not 'NUM_ADQ' in linea and not '-----' in linea:
        if '::::' in linea:
            data = linea.split('::::')
            autor += eliminaEspacios(data[0])
            clasif += eliminaEspacios(data[1])
            titulo += eliminaEspacios(data[2])
            n_sist += eliminaEspacios(data[3])
            n_adq += eliminaEspacios(data[4])
```

```

        nombre += eliminaEspacios(data[5])
        fecha += eliminaEspacios(data[6])
        observacion += eliminaEspacios(data[7])
    else:
        f = open('/home/larroyo/reporte/material.observaciones', 'rb')
        reader = csv.reader(f)
        reader.next()
        for row in reader:
            if int(row[0]) == int(n_adq):
                observacion = row[1]
        f.close()
        libros.append((eliminaTextoAutor(autor.encode('utf-8')), clasif, eliminaTextoAdd(titulo),
n_sist, int(n_adq), nombre, fecha, observacion))
        autor = clasif = titulo = n_sist = n_adq = nombre = fecha = observacion = ''

libros.sort(key=lambda x: x[2]) #Ordenar por título
f = open('/home/larroyo/reporte/nuevo', 'w')
for libro in libros:
    f.write('('+str(libro[4])+') '+libro[2]+'\\n')
    f.write(libro[0]+'\\n')
    f.write('----\\n')
    f.write('----\\n')
    f.write(libro[1]+'\\n')
    if libro[6]:
        f.write(libro[5]+' - '+libro[6]+'\\n\\n')
    else:
        if libro[7]:
            f.write(libro[7]+'\\n\\n')
        else:
            f.write('Disponible\\n\\n')
f.close()

libros.sort(key=lambda x: x[4]) #Ordenar por NA
f = open('/home/larroyo/reporte/material', 'w')
for libro in libros:
    f.write(str(libro[4])+'\\t'+libro[3]+'\\t'+libro[1]+'\\t'+libro[2]+'\\t'+libro[0]+'\\t'+libro[6]+'\\t'+libro[5]
]+'\\t'+libro[7]+'\\n')
f.close()

f = open('/home/larroyo/reporte/material.prestamos', 'w')
for libro in libros:
    if libro[7] != '':
        f.write(str(libro[4])+'\\t'+libro[6]+'\\t'+libro[5]+'\\t'+libro[7]+'\\n')
f.close()

```

APÉNDICE C. SCRIPT DE BÚSQUEDA BIBLIO.

```

#!/usr/bin/perl
open ARCHIVO, "/home/acervo/perl-prog/nuevo";

$j=0;
$h=0;
while($linea = <ARCHIVO>)
{
    chop($linea);
    $libro[$j][1]=$linea;
    $linea=<ARCHIVO>;
    chop($linea);
    $libro[$j][2]=$linea;
    $linea=<ARCHIVO>;
}

```

```

chop($linea);
$libro[$j][3]=$linea;
$linea=<ARCHIVO>;
chop($linea);
$libro[$j][4]=$linea;
$linea=<ARCHIVO>;
chop($linea);
$libro[$j][5]=$linea;
$linea=<ARCHIVO>;
chop($linea);
$libro[$j][6]=$linea;
$linea=<ARCHIVO>;
chop($linea);
$libro[$j][7]=$linea;
$linea=<ARCHIVO>;
chop($linea);
$libro[$j][8]=$linea;
$j++;
}
close ARCHIVO;

$cadena = $ARGV[0];

$global=0;
print"-----\n";
for($i=0;$i<$j;$i++)
{
    $si=0;
    for($k=1;$k<=8;$k++)
    { $_ = $libro[$i][$k]; $si=1 if /$cadena/i ;
    }

    if($si)
    {
        $h++;
        $global=1;
        print $h."\n";
        print "Titulo : ".$libro[$i][1]."\n";
        print "Autor   : ".$libro[$i][2]."\n";
        print "Colocación: ".$libro[$i][5]."\n";
        print "Usuario  : ".$libro[$i][6]."\n";
        print "\n";
    }
}
print"-----\n";
if(!$global){printf "No se encontró\n";}

```