



Catalogación de libros y CDs con códigos de barra y Perl

BARRAS

Los códigos de barras aceleran enormemente las colas en las cajas de los supermercados, pero esta tecnología también es útil en aplicaciones totalmente diferentes. Un barato escáner de código de barras puede ayudarnos a organizar nuestra biblioteca privada con la colección de CDs y DVDs. **POR MICHAEL SCHILLI**

Dealxtrême.com, una empresa ubicada en Hong Kong, ofrece toda una gama de gadgets a un precio muy razonable. Los clientes pueden pagar con Paypal, y el envío es gratuito. ¿Está interesado en un puntero láser por menos de dos dólares o en un adaptador SATA/IDE por sólo ocho dólares? Si no le importa esperar unas dos semanas de entrega, encontrar una ganga está garantizado con Dealxtrême.

Desde hace un tiempo le había echado el ojo a un escáner de código de barras basado en CCD que costaba unos 42 \$ (véase la Figura 1, es uno de los productos más caros de Dealxtrême [2]), hasta que finalmente pulsé el botón de comprar.

Un Paquete de Hong Kong

Cuando el cartero finalmente llegó con el paquete, ya no podía esperar más para comenzar. La opción obvia era escribir una aplicación para que escaneara los códigos

de barras de mi extensa colección de libros técnicos y que guardara los resultados en una base de datos. En función de la procedencia del libro, los códigos de barras estaban impresos bien en formato UPC (Universal Product Number) o en EAN (European Article Number). Amazon.com ofrece un servicio Web gratuito que proporciona información detallada de un producto si se suministra el código de barras en uno de estos dos formatos. Esto significa que un script Perl puede identificar el título y autor de un libro o el artista de un CD con escanear el código. La información suministrada por este servicio incluye imágenes de la portada del disco o del libro. Si añadimos una interfaz gráfica de usuario a la aplicación, podríamos mostrar la portada del libro o CD en pantalla y a todo color tras haber escaneado el código.

El lector tiene una conexión USB, y Linux lo identifica inmediatamente como un teclado secundario. Si mantenemos el

sensor sobre el código de barras de un libro, CD o DVD y presionamos el botón, el escáner enciende la luz roja (véase la Figura 2) y habilita el sensor CCD. A continuación, la CPU interna comienza a analizar las barras de diferente grosor para descubrir el número codificado.

Este escáner de código de barras es muy fiable, genera un pitido cuando ha terminado y envía los números al puerto USB del ordenador de la misma forma que si el usuario los hubiese presionado en un teclado y hubiese pulsado *Return*.

Si el escáner falla al identificar el código de barras, cosa que no ha pasado en ninguno de mis experimentos, el usuario aún puede teclear el número en el cuadro de entrada de la aplicación que se presenta en este artículo, pues el efecto es el mismo.

Imagen a Todo Color

El script, *upscan* (véase el Listado 1), usa una interfaz gráfica basada en la plata-

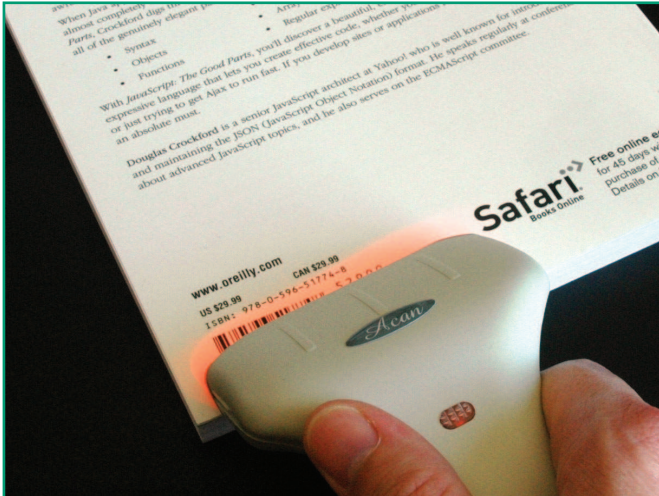


Figura 1: El escáner de código de barras es capaz de leer el código UPC o EAN de un libro.



Figura 2: El escáner tiene un sensor CCD que enciende sus diodos rojos con la pulsación de un botón.

forma Tk. El cuadro de entrada de texto de la interfaz gráfica toma el foco del teclado inmediatamente tras iniciarse. Una vez que el escáner de código de barras ha identificado un código, los números aparecen en el cuadro de entrada. El escáner envía un código de retorno de carro para terminar, y la interfaz gráfica responde con la función `callback scan_done()`. La función envía el código de barras a los Amazon Web Services (AWS), y, tras un retardo cercano al segundo, recibe el título y autor del libro, CD o DVD más una URL que apunta a la imagen JPEG de la portada del libro o CD.

La Figura 3 muestra la aplicación justo después de escanear el código de barras impreso en la contraportada de un libro de JavaScript. Los campos de información se han completado correctamente, y el programa ha recibido la portada correcta desde Amazon. La Figura 4 muestra el resultado tras escanear un CD del vocalista de los Beach Boys, Brian Wilson. En ambos casos el script coloca la información recuperada en una base de datos local SQLite, por la cual podemos navegar a nuestro libre albedrío, así como escribir aplicaciones para ella (véase la Figura 5).

Manténgase a la Espera

Gracias al paquete Tk de CPAN, las interfaces gráficas de usuario no son un pro-

blema para los scripts Perl. Desafortunadamente, apareció un contratiempo con la aplicación que tenía en mente: las operaciones largas, como las peticiones a una Web, provocaban que la interfaz se colgara. Hacer la petición con el código de barras a Amazon podría llevar un par de segundos, y la interfaz se colgaba en este intervalo.

El módulo POE, también de CPAN, me proporcionó una solución: permite a la interfaz gráfica de usuario que se ejecute en el espacio de usuario orientado a eventos del “kernel”. Sin embargo, éste no debe confundirse con el kernel de Linux. En POE, el “kernel” es sólo un extravagante nombre para un bucle de evento. Proporciona mecanismos para la multitarea cooperativa.

El script no maneja las peticiones Web de manera síncrona en este entorno. En su lugar, envía una petición al servidor Web y entonces devuelve el control inmediatamente al kernel POE, y por tanto, al bucle de la interfaz gráfica. Cuando llega la respuesta desde Internet, el kernel despierta a la tarea que esperaba y le pasa la información.

El módulo `Net::Amazon` de CPAN controla las comunicaciones con Amazon y soporta toda una variedad de peticiones

a los servicios Web de este gigante de Internet. Internamente, no hace uso del módulo POE asíncrono para hacer la petición a la base de datos de Amazon. En su lugar, usa `LWP::UserAgent`, que es síncrono. Podemos utilizar el parámetro `ua` para indicarle al módulo que trabaje con un agente de usuario que le podemos pasar.

CPAN dispone de `LWP::UserAgent::POE`, un agente con interfaz LWP que respeta las necesidades especiales asíncronas del kernel POE. Mientras que el módulo realiza las peticiones Web, y parece que espera de forma síncrona los resultados, un poco de magia negra dentro del módulo permite al kernel POE mantenerse a la espera, dando paso a otras tareas.

Bajar a la Base

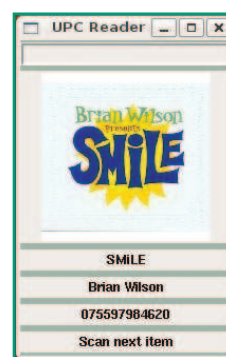


Figura 4: El escáner identifica los CDs y DVDs para interpretar el código UPC y recuperar la información del producto y la portada desde Amazon.com.

El programa `upscan` usa el wrapper de bases de datos `Rose::DB` de CPAN para identificar el esquema de la base de datos e insertar nuevos registros en la tabla `articles` (véase la Figura 6). La línea 19 identifica el archivo de base de datos `articles.dat` en el directorio actual como una base de datos SQLite, y las subsiguientes opciones `AutoCommit` y `RaiseError` aseguran que las nuevas entradas se escriben en la base de datos y sin necesidad de un comando explícito `commit`, y que cualquier error que

pueda ocurrir genere inmediatamente una excepción.

El método *make_classes()* de la línea 23 importa los objetos de la base de datos al código script, y debido a esto, se puede indicar simplemente *Article->new()* posteriormente para preparar una nueva entrada en la tabla de la base de datos *articles*.

Widgets en el Flujo POE

La interfaz gráfica se está ejecutando en la ventana principal, *\$top*, cuya línea 25 acepta desde la variable *\$poe_main_window*, que se exporta desde POE. El bucle de evento de la interfaz gráfica no se deja

a sus propios dispositivos en el script, sino que necesita cooperar con el kernel de POE.

Si *use Tk* aparece antes de *use POE* en el código, POE sabe que tiene que preparar bucle de evento para la interfaz gráfica de Tk. Inicializa la ventana principal y crea un puntero a éste en *\$poe_main_window*.

El comando *configure()* de la línea 26 guarda la cadena del título *UPC Reader* en la cabecera de la ventana y fija el color de fondo de la interfaz gráfica a *#a2b2a3* (un verde oliva claro).

En la parte superior de la ventana podemos ver el widget de entrada *\$entry*, que acepta secuencias de números desde el

escáner y los guarda en la variable global *\$UPC_VAR*. Más abajo se encuentra un widget para la imagen de las portadas de libros y CDs. Por motivos de organización, está empotrado en un widget de etiqueta. Le siguen cuatro widgets de tipo *Label* que guardan el título (*\$PRODUCT*), el autor o artista (*\$BYWHO*), el código UPC o EAN escaneado (*\$UPC*) y un mensaje de estado (*\$FOOTER*).

El bucle *for* de la línea 42 y siguientes sitúa los widgets, de arriba a abajo, en la ventana principal y realiza la llamadas *-fill => "x"* y *-expand => 1* para asegurarse de que las etiquetas cubren el espacio horizontal hasta el borde y escalan auto-

Listado 1: upcscan

```

001 #!/usr/local/bin/perl -w
002 use strict;
003 use Tk;
004 use Tk::JPEG;
005 use POE;
006 use LWP::UserAgent::POE;
007 use Net::Amazon;
008 use Net::Amazon::Request::UPC;
009 use MIME::Base64;
010 use Rose::DB::Object::Loader;
011 use Log::Log4perl qw(:easy);
012
013 my @MODES = qw(books music dvd);
014
015 my $UA = LWP::UserAgent::POE->new();
016
017 my $loader = Rose::DB::Object::Loader->new(
018     db_dsn =>
019         "dbi:SQLite:dbname=articles.dat",
020     db_options => {
021         AutoCommit => 1, RaiseError => 1 },
022 );
023 $loader->make_classes();
024
025 my $top = $poe_main_window;
026 $top->configure(-title => "UPC Reader",
027     -background=> "#a2b2a3");
028 $top->geometry("200x300");
029
030 my $FOOTER = $top->Label();
031 $FOOTER->configure(-text => "Escanear siguiente
032     artículo");
033
034 my $BYWHO = $top->Label();
035 my $UPC = $top->Label();
036 my $PHOTO = $top->Photo(-format => 'jpeg');
037 my $photolabel = $top->Label(-image => $PHOTO);
038 my $entry = $top->Entry(-textvariable => \my
039     $UPC_VAR);
040
041 my $PRODUCT = $top->Label();
042
043 for my $w ($entry, $photolabel, $PRODUCT,
044     $BYWHO, $UPC, $FOOTER) {
045     $w->pack(-side => 'top', -expand => 1, -fill =>
046         "x" );
047 }
048
049 my $session = POE::Session->create(
050     inline_states => {
051         _start => sub{
052             $poe_kernel->delay("_start", 60);
053         }
054     });
055 POE::Kernel->run();
056
057 #####
058 sub scan_done {
059     #####
060     $PHOTO->blank();
061     $PRODUCT->configure(-text => "");
062     $FOOTER->configure(-text => "Procesando ...");
063     $BYWHO->configure(-text => "");
064     $UPC->configure(-text => $UPC_VAR);
065     resp_process(
066         amzn_fetch( $UPC_VAR ) );
067     $UPC_VAR = "";
068 }
069
070 #####
071 sub amzn_fetch {
072     my($upc) = @_;
073     #####
074     my $resp;

```

máticamente cuando se expande la ventana principal.

El comando *bind()* de la línea 47 juega un importante papel. El widget de entrada ignora el carácter retorno de carro enviado por el escáner debido a que el campo de entrada es un campo de una sola línea. El método *bind* enlaza el código de teclado de la tecla Return a la función *scan_done()* definida en la línea 59 y siguientes para desencadenar el procesamiento del código leído por el escáner.

En primer lugar, se llama al método del objeto fotografía *blank()* para eliminar la anterior imagen de portada. A continuación el título en los nombres de autor o

artista se reemplazan con cadenas en blanco. Aparece un mensaje “Processing...” en *\$FOOTER*, y se envía la petición a Amazon mediante la llamada a *amzn_fetch()*. Para prepararse para el siguiente escaneo, la línea 68 borra inmediatamente el código leído por el escáner desde el widget de entrada. El código de barras del producto vigente se guarda en el widget *\$UPC*.

Tras completar estas preparaciones, la línea 49 define una sesión POE, y la línea 56 inicia el kernel POE, que controla el programa desde este punto hasta que finalice. El kernel POE acepta la entrada de usuario, pulsaciones de ratón o entradas

desde el teclado o escáner y se asegura de que se asigna un segmento de tiempo a cada manejador desencadenado por un evento. POE es muy estricto con las sesiones que controla. Si no tienen una tarea para completar, se eliminan. POE no entiende que una aplicación Tk pueda estar esperando la entrada de usuario. Para evitar que elimine a la interfaz gráfica de usuario, las líneas 49 a 54 definen una sesión que salta al evento *_start* cada 60 segundos.

Pregunta en Amazon

Una vez que el escáner ha devuelto el código UPC o EAN, la función

Listado 1: upcscan (Contin.)

```

077                                     117 my $a = Article->new();
078 my $amzn = Net::Amazon->new(        118 $a->upc($upc);
079     token => 'XXXXXXXXXXXXXXXXXXXX', 119 $a->type($mode);
080     ua    => $UA,                    120 $a->title( $property->Title() );
081 );                                    121
082                                     122 if($mode eq "books") {
083 for my $mode (@MODES) {             123     $a->bywho( $property->author() );
084                                     124 } elsif( $mode eq "music") {
085     my $req =                        125     $a->bywho( $property->artist() );
086     Net::Amazon::Request::UPC->new( 126 } else {
087         upc => $upc,                 127     $a->bywho( "" );
088         mode => $mode,               128 }
089     );                                129
090                                     130 $BYWHO->configure(-text => $a->bywho() );
091 $resp = $amzn->request($req);        131 $PRODUCT->configure(-text => $a->title() );
092                                     132
093 if($resp->is_success()) {            133 if($a->load( speculative => 1 )) {
094     return($resp, $mode, $upc);      134     $PRODUCT->configure(
095     last;                             135         -text => "YA EXISTE");
096 }                                     136 } else {
097                                     137     $a->save();
098     WARN "Nothing found in mode '$mode'; 138 }
099 }                                     139
100 return $resp;                       140 $FOOTER->configure(-text => "Escanear siguiente
101 }                                     artículo");
102                                     141 return 1;
103 #####                                142 }
104 sub resp_process {                  143
105 #####                                144 #####
106 my($resp, $mode, $upc) = @_;        145 sub img_display {
107                                     146 #####
108 if($resp->is_error()) {              147 my($imgurl) = @_;
109     $PRODUCT->configure(-text => "NOT FOUND"); 148
110     return 0;                        149 my $imgresp = $UA->get( $imgurl );
111 }                                     150
112                                     151 if($imgresp->is_success()) {
113 my ($property) = $resp->properties(); 152     $PHOTO->configure( -data => encode_base64(
114 my $imgurl = $property->ImageUrlMedium(); $imgresp->content() );
115 img_display( $imgurl );              153 }
116                                     154 }

```

```
$ sqlite3 articles.dat
SQLite version 3.4.2
Enter ".help" for instructions
sqlite> .width 14 40 20
sqlite> .mode col
sqlite> .headers on
sqlite> select upc, title, bywho from articles;
upc          title          bywho
-----
724356061125  Minimum-Maximum  Kraftwerk
9780596009762  SQL Cookbook (Cookbooks (O'Reilly))  Anthony Molinaro
9780596517748  JavaScript: The Good Parts  Douglas Crockford
9781416553649  Born Standing Up: A Comic's Life  Steve Martin
51075597984620  SMiLE  Brian Wilson
sqlite>
$
```

Figura 5: La información del artículo escaneado se envía a la base de datos SQLite.

`amzn_fetch()` de la línea 72 y siguientes crea una instancia del objeto `Net::Amazon` y la pasa tanto al token del desarrollador (tendrá que solicitarla a Amazon si no tiene una, véase la sección “Instalación” más adelante), como al agente especial global `LWP::UserAgent::POE`, que no sólo recupera peticiones Web, como su clase de base `LWP::UserAgent`, sino que también colabora con POE.

Un objeto de petición tipo `Net::Amazon::Request::UPC` se comunica con el Web Service de Amazon que proporciona la funcionalidad de búsqueda UPC/EAN. La petición debe especificar si se quiere buscar el código UPC/EAN en las secciones de libros, música o DVDs.

Como el escáner no sabe si está escaneando un libro o un CD, el bucle `for` que comienza en la línea 83, simplemente prueba con las tres secciones soportadas y finaliza tan pronto como Amazon reporta una búsqueda satisfactoria. La respuesta devuelta ofrece el método `is_success()`, que indica si se ha encontrado el código coincidente.

La función `amzn_fetch()` devuelve tres parámetros: el objeto respuesta `$resp`, la sección en la cual se encontró la coincidencia (libros, música o DVD), y el código UPC/EAN escaneado.

La función `resp_process()` definida en la línea 104 y siguientes, recupera los resultados y los usa para actualizar los campos de la interfaz gráfica de usuario. El método `ImageUrlMedium()` del artículo recuperado en `$property` incluye una URL a una imagen JPEG de tamaño medio en el servidor de Amazon que muestra la portada del producto.

Leer JPEGs

Para lograr que el widget de foto de la plataforma Tk pueda leer y mostrar

imágenes JPEG, la línea 4 carga el módulo `Tk::JPEG`, que también es parte de la distribución de Tk. La función `img_display()` definida en las líneas 145 y siguientes aguarda la URL de una imagen almacenada en Amazon.com y a continuación pide al agente de usuario POE que la recupere.

Debido a que el widget de foto de Tk insiste en trabajar con información codificada en Base-64 (al menos para los JPEGs), la función `encode_base64()` del módulo `MIME::Base64` convierte la información JPEG extraída mediante `content()` a un formato que el widget puede manejar. Tras esto, el método `configure()` del widget de foto fija la opción `-data`, que indica al widget que analice la información, la convierta a formato interno Tk y muestre la imagen decodificada en la interfaz gráfica de usuario.

Base de Datos con Wrappers

Volvemos ahora a la función `resp_process()`: no sólo muestra la información del producto, sino que también la guarda en la base de datos. Para permitir esto, en la línea 117 del wrapper `Rose::DB` crea un nuevo objeto de tipo `Article` y configura los campos de los objetos `upc`, `type`, `title` y `bywho`, que corresponden a las columnas de la base de datos con los mismos nombres.

El método `load()` con el atributo `speculative` intenta encontrar entonces una entrada coincidente en la base de datos. Si la búsqueda tiene éxito, el script envía un mensaje “ALREADY EXISTS” a la interfaz gráfica para indicarle al operador que ese producto ya se ha escaneado. Si `load()` falla, `save()` en la línea 137 guarda el artículo recién introducido en la base de datos.

```
CREATE TABLE 'articles' (
  id INTEGER PRIMARY KEY,
  upc TEXT,
  type TEXT,
  title TEXT,
  bywho TEXT,
  UNIQUE(upc, type)
);
--
"schema.sql" 8L. 132C 1.1      All
```

Figura 6: Esquema de bases de datos SQLite.

Instalación

Antes de que el script pueda usar AWS, debemos solicitar un token de desarrollador de Amazon e introducirlo en la línea 79. El proceso es rápido y lo podemos desencadenar [3] introduciendo una dirección de email válida a la que se enviará en token. Sin un token es válido, el script nos indicará `NOT FOUND` una y otra vez. La base de datos se crea por el cliente SQLite `sqlite3` cuando se le pasa el archivo de esquema (véase la Figura 6): el cliente crea entonces la base de datos `articles.dat` y una base de datos vacía, `articles`, que contiene las columnas `id`, `upc` (el código UPC/EAN), `type` (libro, música o DVD), `title` (el título de libro o CD/DVD), y `bywho` (el autor o artista).

El comando `UNIQUE` del SQL de la base de datos usa el tipo de artículo y el número UPC como claves primarias para permitir al wrapper de bases de datos Rose que ejecute `load()` rápidamente y verifique si el producto con un UPC y tipo dados han sido ya escaneados.

Conclusión

Todos los módulos usados aquí están disponibles en CPAN y pueden instalarse desde una shell de CPAN. La base de datos de artículos creada por el script puede usarse de diferentes maneras: como ayuda a la hora de comprar (¿tengo ya este libro en casa?), como sistema de biblioteca de CD o (si incluimos detalles de la ubicación, como “habitación 1, estantería 4, compartimento 3”) como un catálogo en línea para bibliotecarios despistados. ■

RECURSOS

- [1] Listados de este artículo: http://www.linux-magazine.com/resources/article_code
- [2] Dealextrème: <http://www.dealextrème.com/details.dx/sku.12559>
- [3] Amazon Web Services: <http://amazon.com/soap>