

NOMBRES de los AUTORES: Verónica Lencinas, [REDACTED] ; Roberto Cattaneo, [REDACTED]

INSTITUCIÓN: Escuela de Bibliotecología de la Universidad Nacional de Córdoba ;  
Biblioteca Observatorio Astronómico de Córdoba

DIRECCIÓN: Pab. Casa Verde, Ciudad Universitaria ; Laprida 854, 5000 Córdoba

EMAIL: [verolencinas@yahoo.com.ar](mailto:verolencinas@yahoo.com.ar); [neolibris@gmail.com](mailto:neolibris@gmail.com)

TÍTULO: Migración de registros bibliográficos a Marc21. Problemáticas, técnicas y experiencias en torno a la implementación del sistema Digibepé (Koha) de Conabip.

### **Resumen:**

A partir de la experiencia de la Red Puntobiblio en la construcción de un catálogo colectivo de bibliotecas de diferentes tipologías: populares, escolares, especializadas, etc. la Comisión Nacional Protectora de Bibliotecas Populares solicitó a la responsable técnica del proyecto Puntobiblio, Verónica Lencinas, y a un informático especializado, Pablo Bianchi, la presentación de un proyecto que permitiera implementar un sistema de gestión bibliotecario actualizado en las bibliotecas populares argentinas con el objeto de reemplazar el sistema Sigebi que había quedado obsoleto. Una vez aprobado el proyecto que e iniciado la implementación del sistema Digibepé (Koha), fue necesario migrar los datos, sobre todo los datos de catálogos de bibliotecas al formato Marc21. Se migraron no solamente catálogos en Sigebi sino también aquellos provenientes de Aguapey, Winisis, Troya, Pérgamo, Catalis, OpenMarcoPolo y sistemas propios realizados mayoritariamente en MS Excell, Clipper, FoxPro y MS Access. En el presente aporte se describen estrategias y técnicas utilizadas en la migración de datos diversos al formato Marc21 tal como lo implementa el sistema Digibepé (Koha) y la capacitación realizada a los nodos regionales de asistencia técnica, "Nodos Digibepés", a fin de capacitar a bibliotecarios en la migraciones de datos.

### ***Introducción***

El proyecto Redbipa nace en 2005 a partir del trabajo conjunto de la Presidente de la Biblioteca Popular República Argentina de la Ciudad de Córdoba, Sonia Yulán, y la docente de la Escuela de Bibliotecología de la Universidad Nacional de Córdoba, Verónica Lencinas, usuaria de la Biblioteca. En Septiembre de 2005 se subió la página Web de la Biblioteca con su catálogo en línea. Una de las funcionalidades que ofrecía

el catálogo de la Biblioteca en Internet era la posibilidad de “bajar” registros y con ello facilitar la catalogación por copia.

A partir de la experiencia positiva en cuanto a difusión y visibilidad de la Biblioteca y la creciente utilización del catálogo en línea que se plasmaba en consultas bibliográficas vía correo electrónico a la Biblioteca, se propuso a la Federación de Bibliotecas Populares de Córdoba (Febipo) ampliar la experiencia a otras bibliotecas y crear un catálogo colectivo provincial. La propuesta fue bien recibida, pero no tuvo consecuencias inmediatas.

A fines del 2006 el servidor que hospedaba la página Web de la Biblioteca Popular República Argentina suspendió este servicio por superar el tráfico permitido. Un análisis detallado permitió identificar al catálogo en línea como receptor del 88% de las consultas al sitio que superaban las 40.000 páginas servidas por mes. Con este éxito y ante la falta de respuesta de la Federación de Bibliotecas Populares de Córdoba, se decidió ampliar la experiencia a un grupo reducido de bibliotecas populares “amigas”. En 2007 se incorporaron 6 catálogos en línea de Bibliotecas Populares a través del sitio Web de la Biblioteca Popular República Argentina.

Con motivo del Encuentro Nacional de Bibliotecas Populares en Mayo de 2007, se presentó a la Comisión Nacional de Bibliotecas Populares, organizadora del evento, un proyecto ampliado, que proponía crear un espacio digital para las Bibliotecas Populares y sus lectores que sirva de herramienta de cooperación entre las Bibliotecas Populares, de plataforma colaborativa entre lectores y que permitiera localizar el patrimonio cultural existente en las Bibliotecas Populares.

Se proponía lograr estos objetivos a través de un portal que contenga no solamente las páginas Web de las Bibliotecas, sino también sus catálogos en línea, e incluyera herramientas de comunicación y de difusión de las actividades de las Bibliotecas Populares tales como una Agenda Cultural y Exposiciones Virtuales.

A partir de la experiencia de la Red Puntobiblio en la construcción de un catálogo colectivo de bibliotecas de diferentes tipologías: populares, escolares, especializadas, etc. la Comisión Nacional Protectora de Bibliotecas Populares solicitó a la responsable técnica del proyecto Puntobiblio, Verónica Lencinas, y a un informático especializado, Pablo Bianchi, la presentación de un proyecto que permitiera implementar un sistema de gestión bibliotecario actualizado en las bibliotecas populares argentinas con el objeto de reemplazar el sistema Sigebi que había quedado obsoleto. Una vez aprobado el proyecto a fines de 2009, se comenzó con la implementación del sistema Digibepé (Koha) dentro del marco del Programa de Inclusión Digital. Para ello fue

necesario migrar los datos, sobre todo los datos de catálogos de bibliotecas al formato Marc21.

### ***Migración de datos a Koha***

Una decisión temprana que se tomó respecto al proyecto de Conabip fue la migración de todos los datos bibliográficos y de socios. Al comienzo sólo se tenía seguridad de que los datos de Sigebi se iban a migrar. Un script<sup>1</sup> de migración (*sigebi2marc.pl*) ya había sido desarrollado para el proyecto Redbipa (Puntobiblio) a partir de desarrollos realizados en el marco del proyecto de investigación “Creación del Centro de documentación y servicios de la producción cinematográfica y audiovisual de un Polo Regional. Segunda Etapa: Sistema de Información y documentación para la producción cinematográfica y audiovisual”, años 2005-2006, Código 05/F482. También se disponía de un script de migración para bases bibliográficas de Aguapey (*aguapey2marc.pl*). Después de considerar que era necesario resguardar el trabajo realizado por bibliotecarios populares y teniendo en cuenta los problemas que habían tenido muchas bibliotecas en instalar o continuar con Sigebi<sup>2</sup> y por ello habían implementado otros sistemas, los responsables de la Conabip decidieron que se migrarían los datos migrables (catálogo y socios) de todos los sistemas.

### **Migración de sistemas basados en CDS/ISIS**

La mayoría de las migraciones se realizaron sobre sistemas CDS/ISIS. Los scripts de migración para Sigebi (*sigebi2marc.pl*) y Aguapey (*aguapey2marc.pl*) funcionan de manera similar, en primer instancia se obtiene un listado de toda la base (dump) usando *openisis*. Este primer listado tiene el formato de *openisis* que es número de campo – tabulador – contenidos del campo para cada campo, separando un registro del próximo mediante una línea en blanco. El segundo paso es ejecutar el script de migración indicando el archivo que contiene el listado de la base obtenido en el primer

- 
- 1 Scripts son programas que -a diferencia de un programa ejecutable- son leídos primero por el compilador, compilados en tiempo real y después ejecutados, es decir que son creados como ejecutables en el momento del uso. La ventaja de utilizar scripts en lugar de ejecutables es que es sencillo revisar el código fuente y modificarlo. Generalmente la documentación del programa mismo (ayudas, explicaciones) está contenida directamente en el código fuente. Al final de este aporte se incluyen dos scripts de migración como ejemplo.
  - 2 Al funcionar sobre el sistema operativo MS-DOS, y el hecho que la empresa Microsoft ha reducido la compatibilidad de sus sistemas con MS-DOS a partir de la Windows ME en el 2000, el uso de programas que requieren una compatibilidad completa con MS-DOS se complicó cada vez más. En cuanto al uso de Sigebi sobre Linux, el sistema CDS/ISIS (sobre el cual está basado Sigebi) funciona sobre el emulador Dosemu, pero la configuración de caracteres extendidos (letras acentuadas, eñes) es bastante complicada y escasamente documentada.

paso y el código de organización Marc21 para generar los campos 040 y 952<sup>3</sup>. A continuación el script lee una por una las líneas del listado obtenido de la base CDS/ISIS y según el análisis del campo en cuestión y los datos encontrados elige modificar los datos para que correspondan a la norma de Marc21 y realiza una salida de los datos en el mismo formato de OpenIsis. El script también genera datos en forma automática: fecha y hora (campo 005) y un campo 008. Este script también maneja el cambio de codificación de caracteres de la codificación en CDS/ISIS<sup>4</sup> a UTF-8 usado por Koha. En un tercer paso el archivo generado por este script se pasa al formato de archivos de Marc21 con el script *rec2marc.pl*, generando aquí una cabecera estándar. Una vez cumplido los tres pasos se dispone de un archivo en formato Marc21 que se puede subir a Koha. Hubiera sido posible crear un solo script que realice los tres pasos juntos. Sin embargo de esta manera se tiene acceso a cada uno de los pasos intermedios que se pueden revisar en caso que el resultado no es el esperado y ajustar los parámetros acorde.

En el caso de OpenMarcoPolo se creó un script similar, aunque es muy sencillo debido a que OpenMarcoPolo ya utiliza Marc21 y sólo hay que realizar ajustes de codificación y de registro de existencias.

Otros sistemas basados en CDS/ISIS, tales como Catalis o bases generados con CDS/ISIS o Winisis, se migraron con otro procedimiento ya que se presentaban bases de los más diversos formatos y no era posible generar un script para cada uno de ellos. En estos casos se creaba un archivo PFT<sup>5</sup> utilizado para imprimir la base completa en un archivo cuyo formato era similar al formato OpenIsis (número de campo, tabulador, contenidos del campo). Este archivo después se procesaba con una variante del programa *rec2marc.pl* (*rec2marc\_encode.pl*) que re-codificaba los caracteres, generaba un archivo Marc correcto y se podía importar directamente al Koha.

### **Migración de datos basados en MS Excell, MS Access, dbase, VisualFox y otros sin relaciones**

Se trabajó con más de 100 sistemas de bibliotecas basados en listado de MS Excell, MS Access, dbase, VisualFox y otros sistemas relacionales pero cuyo diseño no

---

3 En el campo 040 de Marc21 se guardan los códigos de las bibliotecas que intervinieron en la generación y la modificación del registro. En el campo 952, Koha almacena los datos de las existencias: inventarios, signaturas, etc. incluyendo el código de la biblioteca.

4 En realidad CDS/ISIS no tiene configuración de caracteres, sino que la configuración es realizada por el sistema operativo MS-DOS.

5 PFT significa "Printformat" y es un archivo que permite indicar el formato de visualización de un registro o una parte de un registro. La salida se produce en la pantalla o a través de la impresión. Para crear estos archivos de migración es necesario tener conocimiento del lenguaje de formateo CDS/ISIS.

incluía relaciones en las bases bibliográficas o lo hacía en forma sencilla. A diferencia de las bases CDS/ISIS que trabajan con estructuras de datos bibliográficas, es decir la capacidad de generar campos repetibles y de longitud variable en la que la normalización se realiza durante la catalogación, las bases de datos relacionales se caracterizan por tener en cuenta la normalización de los datos en el diseño mismo, evitando la duplicación de datos. Este tipo de estructuras dificultan la creación de bases de datos bibliográficas, salvo que se simplifiquen al máximo. Por ejemplo un diseño común era el siguiente:

<b>Autor</b>	<b>Título</b>	<b>Editorial</b>	<b>Inventario</b>	<b>Estante</b>	<b>Materia</b>
Rosenberg, Raquel	Contabilidad 1	El Ateneo	2945	5 – 5	Contabilidad

Este tipo de diseño no contempla los casos en los que hay más de un autor o si es necesario colocar más de una materia. Las variantes de este esquema fueron múltiples, por ejemplo en algunos casos recibimos archivo MS Excell en el que cada pestaña se encontraban los libros de una materia determinada.

En todos los casos se generó un archivo delimitado por comas (terminación .csv) mediante diversas herramientas<sup>6</sup> que en una segunda instancia se migraba utilizando el programa MarcEdit. MarcEdit no es software libre, pero si está disponible gratuitamente en Internet<sup>7</sup>. Este sistema está disponible para sistemas operativos MS Windows y GNU/Linux. Dentro de MarcEdit se incluye un utilitario que permite pasar de un archivo delimitado por comas a un archivo Marc en formato .mrk. Este formato (.mrk) es muy similar -pero no idéntico- al formato usado por Openisis. En este caso cada línea comienza con el signo igual, seguido por el número de campo representado por tres dígitos (por ejemplo =008), seguido por dos espacios y a partir de allí comienzan los contenidos del campo, es decir indicadores y subcampos. El mismo MarcEdit incluye una herramienta para generar el formato Marc final.

Esta herramienta permite asignar a cada columna un campo o subcampo Marc21. También se pueden combinar diferentes subcampos en un solo campo y generar automáticamente la cabecera y el 008. Dentro del editor MarcEdit es posible realizar sustituciones u otras transformaciones.

Por lo general este tipo de migraciones requieren un trabajo dedicado de pruebas y ajustes y pueden llegar a demorar bastante tiempo en caso de bases de varios miles de registros.

---

6 Los sistemas MS Excell y dbase se trabajaron el LibreOffice que permite la exportación en formato csv. Los sistemas MS Access se trabajaron mayoritariamente con MDB Viewer. En el caso de VisualFox, se obtuvo una salida a medida utilizando el mismo sistema (Visual Fox).

7 <http://people.oregonstate.edu/~reaset/marcedit/>

## Migración de sistemas basados en Trooya, Pέργamo y bases de datos relacionales.

En estos casos también se trataba de bases de datos relacionales, pero con la dificultad de que uno o varios campos se encontraban relacionados con otras tablas. En este caso nos encontrábamos con situaciones como la siguiente estructura de datos<sup>8</sup>:

campos tabla 1:

no\_registro  
título  
subtítulo  
editorial  
fecha\_publicación  
lugar\_editorial  
...

campos tabla 2:

no\_registro  
apellido\_autor  
nombre\_autor

campos tabla 3:

registro\_libro  
registro\_autor

con los siguientes contenidos:

tabla1:

no_registro	título	editorial	fecha_publicación	lugar_editorial
215	Seis problemas para don Isidro Parodi	Emecé	1995	Buenos Aires

tabla2:

no_registro	apellido_autor	nombre_autor
15	Borges	Jorge Luis
243	Bioy Casares	Adolfo

tabla3:

registro_libro	registro_autor
215	15

8 Este tipo de estructuras de datos son muy comunes en aplicaciones informáticas y requieren una planificación cuidadosa a fin de poder reflejar fielmente el modelo de datos elegido y mantener la integridad de los datos. No es el tipo de estructuras a los que estamos habituados los bibliotecarios ya que requieren generar tablas relacionadas para cada campo de datos que pueda tener más de una ocurrencia (repetibilidad de campos y subcampos)

Este tipo de estructura de datos permite mantener la normalización de los datos, pero requiere complejizar el modelo de datos para cada tipo de campo que puede ocurrir más de una vez, por ejemplo autores, instituciones, materias, editoriales, notas, etc.

Los sistemas PÉRGAMO y Trooya por ejemplo están construidas sobre un modelo de datos con gran cantidad de tablas que requieren reunir los datos de un solo registro a través de múltiples consultas.

En estos casos se optó por crear scripts de migración que realizaban las consultas y generaban un archivo plano que podía ser transformado en un registro marc. Para facilitar el acceso a los datos mediante herramientas uniformes, se optó generalmente por migrar primero las tablas a una base MySQL y realizar el trabajo de generación del archivo marc en un segundo paso.

### **Problemas y dificultades encontradas**

En el transcurso del proyecto se tuvo que enfrentar y solucionar gran cantidad de problemas. El problema más común era identificar la codificación de caracteres en las que se encontraban los datos. Esto generalmente se hacía con un editor que permitía identificar y cambiar la codificación<sup>9</sup>. Dentro de esta problemática se encontraba también el molesto caso de las diferencias en la codificación del final de línea que es diferente en MS Windows y en GNU/Linux. Especialmente MarcEdit reaccionaba muy celosamente a finales de línea que no estaban codificados por MS Windows<sup>10</sup>.

En varias bases de datos – sobre todo las de origen MS Access- se encontraban códigos de formato, mayoritariamente RTF<sup>11</sup>, pero también saltos de línea y códigos HTML mezclado con los datos. Este tipo de codificación de formato dentro de un campo de datos resultaba siempre muy molesto ya que no se pudo localizar una herramienta general para lidiar con el problema y estos códigos solían romper la compilación a un archivo marc. La solución era generalmente realizar varios pasos de sustitución a fin de limpiar los códigos o -en caso que no era posible o que quedaran residuos- editar los datos a mano. En algunos casos de campos de escasa utilidad, se optó por no migrarlos.

Otro problema consistía en encontrar códigos locales inidentificables dentro de los datos. Por ejemplo una tabla con códigos de una sola letra sin que la etiqueta del

---

9 Según el tipo de archivos que se habían enviado a Conabip se trabajó mayoritariamente con LibreOffice Calc y el editor Kate.

10 En estos casos MarcEdit modificaba el archivo de origen y eliminaba TODOS los saltos de línea lo que obligaba en casos en los que se había hecho algún trabajo de pre-procesamiento de volver a repetirlo.

11 Rich Text Format.

campo de una pista sobre qué datos se trataba. Generalmente se debía interrumpir el proceso de migración y consultar con la biblioteca que había enviado la base. En muchos casos se pudo obviar estos datos -por ejemplo si el libro se encontraba prestado o no- en otros casos requirió una traducción de los códigos a una forma utilizable dentro de Koha, por ejemplo si hacían referencia al modo de adquisición o a una clasificación propia.

Un problema especial lo constituyó la migración del Kardex de Sigebi. En las bibliotecas universitarias se utilizan generalmente dos tipos de registro para describir datos pertenecientes a publicaciones periódicas: registros de título de la publicación que contienen los datos de la publicación en general (por ejemplo título, editor, ISSN, periodicidad, etc.) y registros analíticos que describen artículos en las revistas. En el caso de Sigebi existían dos tipos de registros: registros de títulos y registros de números (fascículos) y ambos tipos de registros se encontraban relacionados, de manera que se podía visualizar en un registro de título cuáles eran los fascículos de este título y en los registros de números se visualizaban los datos de la publicación. Sin embargo se detectaron registros en los que estas relaciones se encontraban rotas y no era posible relacionar un número con un título. Se generaron dos scripts de migración, uno de ellos para generar registros de títulos que cargaba los datos de los números de la revista como ejemplares. Este script tenía el problema que en la práctica se perdían los descriptores temáticos ya que Koha no prevé cargar materias en los datos de los ejemplares. Se migraban estos datos a un campo de nota local del ejemplar, pero no se podía recuperar el registro a través de estas notas. El segundo script de migración continuaba con el mismo modelo bibliográfico de Sigebi y fue el elegido por todas las bibliotecas que utilizaban el módulo de Kardex de Sigebi. Igualmente la migración de los Kardex no fue del todo satisfactoria en cuanto a recuperación y visualización de datos en Koha.

### ***Formación de nodos regionales***

El proyecto presentado a Conabip incluía la creación de nodos regionales que daban soporte técnico. En agosto de 2010 se crea el primer nodo regional la Ciudad de Córdoba, Argentina, en la funcionó hasta junio de 2012. Debido a la gran cantidad de migraciones que se debían realizar y estaban generando un cuello de botella en la implementación, se capacitó a los operadores del Nodo Córdoba a fin de que comenzaran a realizar migraciones de Sigebi y Aguapey. Posteriormente con la implementación de los Nodos Conabip, La Plata y Patagonia se realizaron capacitaciones para los operadores de estos nodos y las migraciones se comenzaron a distribuir entre todos los Nodos. Actualmente el Nodo La Plata realiza migraciones



de MS Excell, el Nodo Patagonia las migraciones de Sigebi y Aguapey y el Nodo Córdoba las migraciones de Winisis. El Nodo Conabip realiza diferentes tipos de migraciones equilibrando las demandas puntuales y Verónica Lencinas como asesora técnica de Conabip realiza las migraciones más complejas.

## **Conclusiones**

En los algo más de cinco años en los que funciona el programa Digibepé, el sistema está siendo usado por más de 700 bibliotecas en todo el país. Conabip no solamente ofrece el sistema a las bibliotecas populares sino también a otras instituciones con bibliotecas públicas. En este tiempo el sistema se ha puesto en marcha para bibliotecas que no tenían sistema se han realizado más de 400 migraciones de una cantidad innumerable de sistemas tanto caseros como profesionales. Se ha podido adaptar Digibepé a las necesidades de las bibliotecas populares y públicas en la Argentina.

Durante el tiempo que lleva el programa, se logro promover la creación de una comunidad de usuarios alrededor de la herramienta Digibepé como también facilitarel uso del sistema a través de la asistencia presencial, telefónica, y “on line” por medio de los foros Digibepé y del correo electrónico. Se está planteando mantener y profundizar la capacitación de los bibliotecarios en bibliotecas populares tanto en técnica bibliotecológica, informática básica y uso del sistema Digibepé con la finalidad de formar formadores. Que bibliotecas populares se encaminen a ser referentes regionales y continúen la tarea de construcción de conocimiento y apropiación tecnológica.

La presencia ya activa de un catálogo colectivo como herramienta para difundir el patrimonio bibliográfico de las Bibliotecas Populares Argentinas, facilitar la localización de material bibliográfico por consulta gratuita y promover el uso de las colecciones de las bibliotecas debe acompañarse todavía por el uso masivo de la catalogación por copia (Z39.50). Actualmente se busca una mayor presencia en el escenario regional, llevando adelante jornadas de capacitación en un nivel más técnico, acentuando la formación en los módulos de catalogación, gestión de socios, diseño OPAC y administración de preferencias del sistema, promoviendo encuentros con las Federaciones Provinciales a fin de fortalecer la capacitación en Digibepé de bibliotecarios que puedan convertirse en referentes y crear así una fuerte comunidad de uso local y regional que asegura la permanencia de la tecnología bibliotecaria más allá de los programas específicos.

## Anexos

### Sigebi2marc.pl

```
#!/usr/bin/perl

# sigebi2marc.pl
# script para reformatar bases de datos Isis de Sigebi a Marc21

# uso: ./sigebi2marc.pl <archivo> <marcorgcode>
# donde <archivo> es el archivo de texto plano obtenido a través de un dump
# de la base de datos con openisis (por ejemplo: openisis -db MARC > marc.txt
# y marcorgcode es el código de la Biblioteca. Este código tiene que estar
# configurado en el sistema antes de importar a los registros.

# El archivo resultante puede ser utilizado por otra aplicación y se encuentra
# en el formato usado por openisis: tag<tab>campo

# (c) 2008 Veronica Lencinas
# Realizado con aportes del proyecto Secyt-UNC 05/F615

#

# ver 1.7 - diciembre 2010 // transforma a ASCII la signatura

use Encode;

$marcorgcode = $ARGV[1];

$mfnc = 1;

$tu_flag = 0; # flag para campo 27 "Título uniforme"

# fecha de hoy que vamos a necesitar mas tarde
@fecha = localtime;
$ano = $fecha[5] + 1900;
$mes = sprintf("%02d", $fecha[4]+1);
$fecha1 = $ano . $mes . sprintf("%02d", $fecha[3]) . sprintf("%02d",
$fecha[2]) . sprintf("%02d", $fecha[1]) . sprintf("%02d", $fecha[0]);
$ffecha = $fecha1 . ".0";
$nn = 0;
my $lib;
#-----

# 1. chequear si estan todos los parametros requeridos
$file = $ARGV[0];

print "No especificó un archivo para leer\n" if $ARGV[0] eq "";

# 2. abrir archivo de texto

$f = 0; # flag para comienzo del archivo
$l = 0; # flag para detectar registros borrados y eliminarlos

open (BASE, $file);

while (<BASE>){

    if ($f == 0){
        $f = 1; next if $_ eq "\n";
    } #elimina la primer linea en blanco

    if ($l == 0){
        next if $_ eq "\n"; # caso de dos o mas lineas en blanco
```

```

}
$l = 1 if $_ ne "\n";
$l = 0 if $_ eq "\n";

# 3. reformatear

if ($l == 0){ # una linea en blanco # 4. sacar por <STDOUT>

    # Modificar acá si encontramos DGMS no definidos aquí, pero
    # presentes en la base de origen. Se deben definir los DGMS
    # como tipos de items en el sistema de recepción (itemtypes en Koha)

    $dgm = "texto" if $dgm eq "";
    $lib = "LIB" if $dgm =~ /texto/;
    $lib = "VID" if $dgm =~ /(video|Video|VI)/;
    $lib = "DVD" if $dgm =~ /DVD/;
    $lib = "CD" if $dgm =~ /(disco compacto|CD)/;
    $lib = "GRAB" if $dgm =~ /grabac/;

    print "1\t$mf\n";
    print "3\t$marcorgcode\n";
    print "5\t$fecha\n"; undef($modif_registro);
    print "942\t ^c$lib\n"; # Aquí va el itemtype, tomar de 24^d,
ahora en $dgm

    $v8 = substr($alta_registro,2); undef($alta_registro);
    $v8 = sprintf("%06d",$v8);
    $fecha =~ s/\D//g;
    $fecha = sprintf("%04d",$fecha);
    $v8 .= "s$fecha |||c";
    print "8\t$v8\n"; undef($v8); undef($fecha);
    print "40\t \^a$marcorgcode\^bspa\^c$marcorgcode\^d$marcorgcode\n";
    print foreach @registro; undef(@registro);
    foreach (@descri){
        print "650\t 4\^a$_\n" unless $_ eq "" or $_ eq "\n";
    }
    undef(@descri);
    chomp($sigtop);
    $sigtop =~ s/\^s/\^b/;
    $sigtop =~ s/\^n/\^h/;
    $sigtop =~ s/\^l/\^m/;
    $sigtop =~ s/\^v/\^t/;

    if ($sigtop =~ /\^b/i){

        # Subcolecciones, ubicación.

        $subcol = substr($sigtop,index($sigtop,"\^b")
+2,index($sigtop,"\^h")-2);
    }

    $subcol = "CG" if $subcol eq "";

    $class = substr($sigtop,index($sigtop,"\^h")
+2,index($sigtop,"\^m")-index($sigtop,"\^h")-2);

    $sigtop = substr($sigtop,index($sigtop,"\^h")+2);
    $sigtop =~ s/\^./ /g;
    $sigtop =~ s/[\^!-\s]//g;
    $subcol =~ s/[\^!-\s]//g;

    #print "82\t04\^a$class\^q$marcorgcode\n"; # clasificacion -
deshabilitado, no todos usan CDD
    foreach (@invent) {

```

```

        $_ =~ s/\^i//;
        print "952\t
\^8$subcol\^a$marcorgcode\^b$marcorgcode\^c$subcol\^p$_\^y$lib\^o$sigtop\n";
    }

    print "\n"; undef(@invent); undef($dgm); undef($subcol); $stu_flag =
0; undef($sigtop);
    $mfncnt ++;
}

else { # tenemos un campo, ahora reformateamos
    ($tag,$campo) = split(/\t/, $_);

    # Modifique esta línea si tiene otra codificación de caracteres

    #Encode::from_to($campo, 'cp437', 'utf-8');
    Encode::from_to($campo, 'cp850', 'utf-8');

    #print "$tag: $campo"; # DEBUG

    $salta_registro = substr($campo,2,8) if $tag == 3;

    if ($tag == 10){ # ISBN

        $campo =~ s/\^n//;
        $campo =~ s/-//g;
        $campo =~ s/\^c.*//g; # Chau códigos raros que puedan estar cargados

        push @registro, "20\t ^a$campo";
        # Insertar aquí generador de ISBN-13.
    }

    if ($tag == 15){ # ISSN

        $campo =~ s/\^n//;
        $campo =~ s/\^c.*//g; # Chau códigos raros que puedan estar cargados
        push @registro, "22\t ^a$campo";
    }

    if ($tag == 24){ # Area de título y mención de responsabilidad

        $tit1 = substr($campo,0,1);
        if ($tit1 ne "\^"){ $campo = "\^" . $campo;}
        $campo =~ s/\^t/\^a/;
        $campo =~ s/\^s/\^b/g;
        $campo =~ s/\^d/\^h[/;
        chomp($campo);
        $campo .= "]" if $campo =~ /\^h/;
        $campo = "10" . $campo;
        $dgm = substr($campo,index($campo,"[",index($campo,"])-
index($campo,"[") if $campo =~ m/\^h/;
        $dgm =~ s/(\[|\])/g;
        push @registro, "245\t$campo\n";
    }

    if ($tag == 27){ # Título uniforme

        $campo = "10^a" . $campo if $tag == 27;

        push @registro, "240\t$campo" if $stu_flag == 0; # Título uniforme
        push @registro, "740\t $campo" if $stu_flag == 1; # Otro Título
uniforme
        $stu_flag = 1 if $stu_flag == 0;

```

```

    }

    if ($tag == 28){ # Otros autores, creadores, personas vinculadas con la
creación o modificación

        $campo =~ s/$^r//g;
        $campo =~ s/$^n//g;
        $campo =~ s/$^a//g;
        $campo =~ s/^\^n/, /g;
        $campo =~ s/^\^r/,^\^e/g;
        $campo = "1 " . $campo unless $campo eq "" or $campo eq "\n";

        push @registro, "700\t$campo";

    }

    if ($tag == 42){ # Edición

        $campo = " ^a" . $campo;
        push @registro, "250\t$campo";

    }

    ($lugar, $editor) = split(/^\^e/, $campo) if $tag == 43;
    $tit1 = substr($lugar, 0, 1);
    if ($tit1 ne "\^"){ $lugar = "\^" . $lugar; }
    $lugar =~ s/^\^l/^\^a/;
    chomp($lugar); # en algunos casos no hay editor cargado.

    if ($tag == 45){
        $v260b = $campo;
        $fecha = substr($campo, index($campo, "\^f")+2);
        if ($v260b =~ m/^\^e/ ){
            ($descri, $fecha) = split("\^f", $v260b);
            $descri =~ s/^\^e/^\^f/;
            $descri =~ s/^\^l/^\^e/;
            $v260b = "\^c" . $fecha . "(" . $descri . ")";
            undef($descri);
        }
        else {
            $v260b =~ s/^\^f/^\^c/;
        }
        chomp $editor;
        push @registro, "260\t $lugar\^b$editor$v260b";
        undef($lugar); undef($editor); undef($v260b);
    }

#50 DESCRIP. FISICA % ^abcdx   abcdx           50 150 0 1
# /// en la BPRA se encuentra la siguiente construcción:
#           ^a2 videocassetes^eVHS^fPal^g50 min.^bson.^hcol.
# correcto: $a1 videocassette of 1 (Beta) (30 min.) : $bsd., col. ; $c1/2 in.
#3(2 copies)

# Aquí hay que continuar trabajando y controlar DVDs, grabaciones, CDs, mapas
y lo que pueda aparecer.

    if ($tag == 50){
        $campo =~ s/(\^h|\^f)/, /;
        $campo =~ s/^\^e/ \(/;
        $campo =~ s/^\^g/ \)\(/;
        $campo =~ s/^\^h/ /;
        $campo =~ s/ \)\)/;
        $campo =~ s/\( /\(/;
        push @registro, "300\t $campo";
    }

#54 NOTAS GRALES. % ^nf           nf           54 500 0 1
# /// BPRA aqui un registro con ^a, muchos otros con ^c (= notas de contenido)
# /// en algunos casos se cargaron las notas de contenido como descriptores
# /// entre <> y en mayusculas sostenidas

```

```
# Digibepe - algunas bibliotecas tienen varios subcampos en este campo
```

```
if ($tag == 54){
    $campo =~ s/></ -- /g;
    $campo =~ s/(<|>)//g;

    @v500 = split(/\^/, $campo);

    foreach (@v500){
        chomp($_);
        $cod_sc = substr($_, 0, 1);
        $campo_5xx = substr($_, 1);
        if ($cod_sc eq "n"){
            push @registro, "500\t \^a$campo_5xx\n";
        }
        if ($cod_sc eq "c"){
            push @registro, "505\t \^a$campo_5xx\n";
        }
    }
}

if ($tag == 65){ # Descriptores

    # ver 1.6

    if ($campo =~ /\^d/ ) { # base LIBRI

        $campo =~ s/\^d//;

        @descri1 = split(/></, $campo); # OJO, esto puede requerir
adaptaciones # bases FACS0 no funciona bien ya que no tienen los descriptores
entre < > sino que el campo es repetible.
# en ese caso cada campo genera dos ocurrencias y después hay
problemas # ya que el campo que genera es 650 \^a^DESCRIPTOR

        for (@descri1){
            $_ =~ s/(<|>)//g;
            chomp;
            push @descri, $_;
        }
    }

    else { # base FACS0

        chomp($campo);
        push @descri, $campo;
    }

    undef(@descri1);
}

#70 NUM. INVENTARIO % ^ive      ive      70 150 0 0
# /// La BPRA ha usadp ^p en lugar de ^e
```

```

if ($tag == 70){
    $campo =~ s/^\^p/\^t/;
    $campo =~ s/^\^e/\^t/;
    chomp $campo;
    $i_flag = substr($campo,2,2);

    if($i_flag eq "VI" or $i_flag eq "CD"){
        $dgm = "VI" if $i_flag eq "VI";
        $dgm = "CD" if $i_flag eq "CD";
    }
    push @invent, $campo;
}

if ($tag == 75){
    $sigtop = $campo;
}

#87 AREA DE SERIE % ^tsrinx      tsrinx                87 250 0 1
# /// BPRA tiene cargado ^y que es en este contexto el vol. de la subserie

if ($tag == 87){
    $campo =~ s/^\^t/0 \^a/;
    $campo =~ s/(\^n|\^y)/ ;\^v/;
    $campo =~ s/^\^x/. /;
    push @registro, "490\t$campo";
}
if ($tag == 100){
    $tit1 = substr($campo,0,1);
    if ($tit1 ne "\^"){ $campo = "\^" . $campo;}
}
$campo =~ s/^\^n/, / if $tag == 100;
$campo =~ s/^\^a/1 \^a/ if $tag == 100;
push @registro, "100\t$campo" if $tag == 100;

#110 ENTIDAD (E. PRINC.) %^ns ns                110 200 0 1
# /// BPRA tiene ^rEditor en algs. casos

$campo =~ s/^\^n/2 \^a/ if $tag == 110;
push @registro, "110\t $campo" if $tag == 110;
$campo = "1 \^a" . $campo if $tag == 130;
push @registro, "790\t$campo" if $tag == 130;

#156 seudonimos                156 60 0 0
# /// Aquí la BPRA carga el nombre mas conocido en el caso de
# seudónimos. En estos casos el 100 incl. el nombre real
# A partir de este campo mejor utilizamos registros de autoridad.

$campo = "1 \^a" . $campo if $tag == 156;
push @registro, "796\t$campo" if $tag == 156;

}

}

close BASE;

#=====

```

## Aguapey2marc.pl

```
#!/usr/bin/perl

# aguapey2marc.pl
# script para reformatear bases de datos Aguapey a Marc21

# uso: ./aguapey2marc.pl <archivo> <marcorgcode>
# donde <archivo> es el archivo de texto plano obtenido a través de un dump
# de la base de datos con openisis (por ejemplo: openisis -db MARC > marc.txt
# y marcorgcode es el código de la Biblioteca. Este código tiene que estar
# configurado en el sistema antes de importar a los registros.

# El archivo resultante puede ser utilizado por otra aplicación y se encuentra
# en el formato usado por openisis: tag<tab>campo

# (c) 2008 Veronica Lencinas
#

# ver 1.5 - adaptado a Koha 3.0 // Itemtypes se obtienen a través del campo
902

use Encode;

$marcorgcode =$ARGV[1];

# fecha de hoy que vamos a necesitar mas tarde
@fecha = localtime;
$ano = $fecha[5] + 1900;
$mes = sprintf("%02d", $fecha[4]+1);
$fecha1 = $ano . $mes . sprintf("%02d", $fecha[3]) . sprintf("%02d",
$fecha[2]) . sprintf("%02d", $fecha[1]) . sprintf("%02d", $fecha[0]);
$ffecha = $fecha1 . ".0";
$nn = 0;
$firstaut = 0;

my $lib;
#-----

# 1. chequear si estan todos los parametros requeridos
$file = $ARGV[0];

print "No especificó un archivo para leer\n" if $ARGV[0] eq "";

# 2. abrir archivo de texto

$f = 0; # flag para comienzo del archivo
$l = 0; # flag para detectar registros borrados y eliminarlos

open (BASE, $file);

while (<BASE>){

    if ($f == 0){
        $f = 1; next if $_ eq "\n";
    } #elimina la primer linea en blanco

    if ($l == 0){
        next if $_ eq "\n"; # caso de dos o mas lineas en blanco
    }
    $l = 1 if $_ ne "\n";
    $l = 0 if $_ eq "\n";

# 3. reformatear

    if ($l == 0){ # una linea en blanco # 4. sacar por <STDOUT>
```



```

$dgm = "texto" if $dgm eq "";
$sgm = "libro" if $sgm eq "";
chomp $dgm; chomp $sgm;
print "3\t$marcorgcode\n";
print "5\t$ffecha\n";
print "40\t \^a$marcorgcode\^bspa\^c$marcorgcode\n";
foreach (@registro){
    $_ =~ s/\#/ /g;
    print;
}
undef(@registro);
print "\n"; undef(@registro); undef($dgm); undef ($sgm);
$nn = 0;
$lib = ""; # vaciar para el prox. registro
$firstaut = 0;
}
else { # tenemos un campo, ahora reformateamos
($tag,$campo) = split(/\t/, $_);

# Si tiene otra codificación, modifique esta línea

#Encode::from_to($campo, 'latin1', 'utf-8');
#Encode::from_to($campo, 'cp437', 'utf-8');
Encode::from_to($campo, 'cp850', 'utf-8');

if ($tag == 2){
    $record_type = substr($campo,5,1);
    $bibl_level = substr($campo,6,1);
    push @registro, "901\t \^a$record_type\^b$bibl_level\n";
    undef($record_type); undef($bibl_level);
}
if ($tag == 8){
    $campo =~ s/espa\244ol/spa/;
    $campo =~ s/español/spa/;
    $campo =~ s/francés/fre/;
    $campo =~ s/inglés/eng/;
    push @registro, "41\t$campo";
}
if ($tag == 18){
    push @registro, "8\t$campo";
}
if ($tag == 20){
    $campo =~ s/-//g;
    push @registro, "$tag\t$campo";
}
if ($tag == 22 or $tag == 24 or $tag == 246 or $tag == 250 or $tag
== 310 or $tag == 500){
    push @registro, "$tag\t$campo";
}
if ($tag == 600 or $tag == 610 or $tag == 611 or $tag == 651 or $tag
== 504 or $tag == 520 or $tag == 521 ){
    push @registro, "$tag\t$campo";
}
if ($tag == 650 or $tag == 653 or $tag == 659){
    push @registro, "650\t$campo";
}

if ($tag == 856 or $tag == 866){
    $cod = substr($campo,2,1);
    if ($cod eq '^'){
        push @registro, "$tag\t$campo";
    }
}

```

```

else {
    push @registro, "$tag\t ^a$campo";
}
undef($cod);
}
if ($tag == 100){
# Aqui la B.P. Santiago Coronel incluye varios autores separados
por guión.
# Hay bibliotecas que tienen varios campos 100.
# Usamos $firstaut = 0 para indicar el primer autor

#$campo =~ s/^\^d/,^\^d/;
chomp $campo;

# Hemos tenido registros donde metieron mano vía Winisis. Aquí el
hack para
# arreglar el error del bibliotecario.
$cod = substr($campo,2,1);

# Corregir campos 100 múltiples
if ($firstaut == 1){
    $tag = "700";
}

if ($cod eq '^'){
    push @registro, "$tag\t$campo\n";
}
else {
    push @registro, "$tag\t1 ^a$campo\n";
}
$firstaut = 1;
undef($cod);
} # FIN if $tag == 100

if ($tag == 110 or $tag == 710){
#$campo =~ s/^\^b/.^\^b/;
#$campo =~ s/ \./\./;
$campo =~ s/]/]/;
$campo =~ s/^[/^[/;
push @registro, "$tag\t$campo";
}
if ($tag == 111){
#$campo =~ s/^\^n/\^n\(/;
#$campo =~ s/^\^d/ :\^d/;
#$campo =~ s/^\^c/ :\^c/;
chomp $campo;
$campo .= ")" if $campo =~ m/\(/;
push @registro, "$tag\t$campo\n";
}
if ($tag == 245){
if ($campo =~ /\^h/){
$lib1 = substr($campo,index($campo,"\^h"),rindex($campo,"\^h")-
index($campo,"\^h"));
#print "LIB: $lib1\n";
$lib = "VID" if $lib1 =~ /video/;
$lib = "GRAB" if $lib1 =~ /grabac/;
$lib = "CF" if $lib1 =~ /recurso elec/;
$lib = "MP" if $lib1 =~ /material cartogr/;
$lib = "MU" if $lib1 =~ /sica/;
$lib = "CIN" if $lib1 =~ /película/;
$lib = "GRAF" if $lib1 =~ /estampa/;
$lib = "REV" if $lib1 =~ /revista/;
$lib = "DIAR" if $lib1 =~ /diario/;
$lib = "JUE" if $lib1 =~ /juego/;
$lib = "ARTE" if $lib1 =~ /arte/;
$lib = "MOD" if $lib1 =~ /modelo/;
}
}

```

```

    }
    chomp $campo;
    #$cod = substr($campo,2,1);
    #if ($cod eq '^'){
        push @registro, "$tag\t$campo\n";
    #}
    #else {
        if ($campo =~ m/^\^h/){
            $dgm = substr($campo,index($campo,"")+1,length($campo)-
index($campo,"["-2));
        }
        # push @registro, "$tag\t \^a$campo\n";
    #}
    #undef($cod);
}
if ($tag == 902){
    chomp $campo;
    $lib = "LIB" if $campo == 1;
    $lib = "CF" if $campo == 2;
    $lib = "MP" if $campo == 3;
    $lib = "REV" if $campo == 4;
    $lib = "VID" if $campo == 5;
    $lib = "MU" if $campo == 6;

    push @registro, "$tag\t$campo\n";
}

if ($tag == 255){
    #$campo =~ s/^\^b/ ;\^b/;
    chomp $campo;
    push @registro, "$tag\t$campo\n";
}
if ($tag == 260){
    #$campo =~ s/^\^b/ :\^b/;
    #$campo =~ s/^\^c/ ;\^c/;
    $campo =~ s/ / /;
    $campo =~ s/[/\(/;
    $campo =~ s/[/\)/;
    chomp $campo;
    push @registro, "$tag\t$campo\n";
}
if ($tag == 300){
    #$campo =~ s/^\^b/ :\^b/;
    #$campo =~ s/^\^c/ ;\^c/;
    #$campo =~ s/^\^e/ \+\^e/;
    $campo =~ s/[/\(/;
    $campo =~ s/[/\)/;
    chomp $campo;
    push @registro, "$tag\t$campo\n";
}
if ($tag == 309){
    $campo =~ s/^\#/ /g;
    $campo =~ s/^\^a/ /g;
    chomp $campo;
    $sgm = $campo;
    chomp $sgm;
    push @registro, "$tag\t$campo\n";
}
if ($tag == 362){
    $campo =~ s/[/\(/g;
    $campo =~ s/[/\)/g;
    chomp $campo;
    push @registro, "$tag\t$campo\n";
}
if ($tag == 440){
    #$campo =~ s/^\^p/.\^p/g;
    #$campo =~ s/^\^v/ ;\^v/g;

```

```

        chomp $campo;
        push @registro, "490\t$campo\n";
    }
    if ($tag == 505){
        #$campo =~ s/^\^r/ \^\^r/g;
        chomp $campo;
        push @registro, "$tag\t$campo\n";
    }
    if ($tag == 800){
        #$campo =~ s/^\^e/,^\^e/g;
        chomp $campo;
        push @registro, "$tag\t$campo\n";
    }

    if ($tag == 773){
        #$campo =~ s/^\^d/.\^d/g;
        chomp $campo;
        push @registro, "$tag\t$campo\n";
    }
    if ($tag == 859){
        subcampos($tag,$campo);
        Encode::from_to($v859a, 'utf-8', 'ascii');
        Encode::from_to($v859m, 'utf-8', 'ascii');
        Encode::from_to($v859n, 'utf-8', 'ascii');
        push @registro, "952\t
\^a$marcorgcode\^b$marcorgcode\^y$lib\^o$v859m $v859n\^p$v859a\n";
    }
}

}

close BASE;
#-----
sub subcampos { # agregado ver. 1.4????
    $tag = $_[0];
    $val = $_[1];
    @subcampos = split(/\^/, $val);
    foreach (@subcampos){
        $subcampo = substr($_, 0, 1);
        $vsubcampo = substr($_, 1);
        $vsubcampo =~ s/\W+$/ /g unless $tag =~ /(100|700)/ and $subcampo eq
"d";
        $vsubcampo =~ s/^\W+//g;
        if ($vsubcampo ne ""){
            eval("\$v$tag$subcampo = \"$vsubcampo\"");
        }
    }
}

}

#=====

```

## Rec2marc.pl

```
#!/usr/bin/perl

# rec2marc.pl - script para crear registros en Marc21 a partir
# de una registro del tipo de OpenISIS (tag <tab> content)
#
# (c) 2007 Veronica Lencinas
# GPL, enjoy!

# Uso: ./rec2marc.pl < file

# TODO:
# * poder pasarle un leader y que lo incluya.
# * Doku

#$Version = 0.9;

$ino = 1;
my @tag;
my @campo;
my @lon;
$i = 0;

foreach(<STDIN>){ # lee el archivo provisto por stdin

    if ($_ eq "\n"){ # Fin de registro

        @rec = split(/\n/, $rec); # transforma el registro en un vector

        foreach (@rec){
            ($t,$v) = split(/\t/, $_);
            $v =~ s/^\^\x1f/g; # reemplazar código de subcampo
            $tag[$i] = sprintf("%03d", $t); # carga el valor del tag
            $campo[$i] = $v; # carga el campo
            $lon[$i] = sprintf("%04d", length($v) + 1); # carga longitud del
campo + cierre campo
            $i++;
        }
        &do_marc;
        undef(@tag);
        undef(@campo);
        undef(@lon);
        $i = 0;
        $rec = "";
    }
    else {
        $rec .= $_ unless $_ eq "\n"; # Colecciona en $rec el registro
    }
}

# -----

sub do_marc{ # Armar archivo marc

    $sep_cmarc = "\x1e";
    $sep_rmarc = "\x1d";
    $nl = "\r\n"; # Si va a usar sus registros bajo unix modifique
# a "\n".

my $total_camos = @campo;

# inicio_datos = 24 + (total_camos * 12) + 1
my $inicio_datos = sprintf("%05d", 24 + ($total_camos * 12) + 1);

# largo_registro = $inicio_datos + suma_de_largo_camos
my $largo_registro = $inicio_datos + 1;
```

```

foreach(@lon){
    $largo_registro += $_;
}

# leader: $largo_registro . "0000022" . $inicio_datos . "0004500"
# Si quiere incluir datos dentro del leader, modifique esta linea.
#$leader = sprintf("%05d", $largo_registro) . "nam 22" . $inicio_datos . "ua
4500"; # esto es para un registro Marc21 Ansel. Recodifique adecuadamente
$leader = sprintf("%05d", $largo_registro) . "nam a22" . $inicio_datos . "ua
4500"; # registro Marc21 Unicode

# directorio:
my $bas = 0;
undef($directorio);

for ($j=0; $j < $total_campos; $j++){
    $directorio = $directorio . $tag[$j] . $lon[$j] . sprintf("%05d", $bas);
    $bas = $bas + $lon[$j];
}

my $iso = $leader . $directorio . $sep_cmarc;

for ($j=0; $j < $total_campos; $j++){
    $iso = $iso . $campo[$j] . $sep_cmarc;
}

$iso .= $sep_rmarc;

print "$iso\nl";

} # fin sub
#####
# Fin

FIN:

```

## Ejemplo de una migración de Winisis. Archivo PFT para migración: 2Marc.PFT

```
'1      'mfñ/
'3      0018'/,
'5      201102140000.0'/,
'8      000000s'v43'      |||||||||||||000 | spa||'/,
if p(v47) then '20      ^a',v47/,fi,
'40      ^a0018^bspa^c0018'/,

if v6='m' or v6='mc' or a(v6) then
(if p(v16) then if occ=1 then '100      1 ^a' else '700      1 ^a' fi, v16/fi,)
(if p(v17) then if occ=1 then '110      1 ^a' else '710      1 ^a' ,fi, v17/fi,)),
'245      10^a'v18/
if p(v41) then '250      ^a'v41,/fi,
if p(v38) then '260      ^a'v39,'^b'v38'^c'v43/fi,
if p(v42) then '300      ^a'v42/fi,
if p(v20) then '300      ^a'v20 if p(v118) then '^edisquete'fi, /fi,
if p(v90) then '990      ^a'v90/fi,
if p(v113) then '300      ^a' if v113='Video' then v112' video',fi "
("v117")," ("v116)"," ^b"v114/fi,
fi,

if v6='am' then
(if p(v10) then if occ=1 then '100      1 ^a' else '700      1 ^a' fi, v10/fi,)
(if p(v11) then if occ=1 then '110      1 ^a' else '710      1 ^a' ,fi, v11/fi,))
'245      10^a'v12/
if p(v20) then '300      ^a'v20/fi,
if p(v18) then '773      ^a' if p(v16) or p(v17) then v16^*,v17,'^t'v18 else
v18,fi,"^b"v41,"^c"v39," : "v38," , "v43      /fi,
fi,

(if p(v25) then '490      0^a'v25,"^v"v21/fi,)

(if p(v23) then '700      1 ^a'v23,/fi,)
(if p(v24) then '710      1 ^a'v24,/fi,)

(if p(v68) then '500      ^a'v68/fi,)
(if p(v27) then '500      ^aLa colecciøn tiene un total de 'v27'
volumenes'/fi,)
(if p(v30) then '500      ^aTitulo de la publicacion periodica: 'v30/fi,)
(if p(v134) then '500      ^aExistencias completas: 'v134/fi,)
(if p(v219) then '500      ^aEstado del material: 'v219/fi,)
(if p(v400) then '500      ^aMonografico: 'v400/fi,)
(if p(v72) then '520      ^a'v72/fi,)
(if p(v81) then '521      ^aNivel: 'v81/fi,)
(if p(v115) then '525      ^aMaterial complementario: 'v115/fi,)
(if p(v64) then '546      ^aIdioma: 'v64/fi,)
(if p(v168) then if occ=1 then '505      ^a' fi, v168+| -- |fi,if
occ=nocc(v168) then / fi,)

(if p(v80) then '650      14^a'v80/fi,)
(if p(v76) then '650      4^a'v76^*/fi,)
(if p(v99) then '653      4^a'v99/fi,)

if p(v5) then '905      ^a'v5/fi,
if p(v6) then '906      ^a'v6/fi,
if p(v91) then '900      ^a'v91/fi,
if p(v97) then '907      ^a'v97/fi,
if p(v62) then '962      ^a'v62/fi,
if p(v218) then '986      ^a'v218/fi,

(if p(v2) then |952      ^a0018^b0018^p|,v2 , if v120 = 'd' then '^eDON' fi,
'^o'v121[1] ,'^yLIB'/fi,)#
```