



E-Ciencias de la Información
Revista electrónica publicada por la
Escuela de Bibliotecología y Ciencias de la Información,
Universidad de Costa Rica, 2060 San José, Costa Rica



E-Ciencias de la Información

Revista electrónica semestral, ISSN-1659-4142

Volumen 3, número 1, artículo 1

Enero - junio, 2013

Publicado 1 de enero, 2013

<http://revistaebci.ucr.ac.cr/>

Clasificación del *software* libre orientado a la automatización integral de bibliotecas según el nivel de complejidad de la biblioteca: bibliotecas simples, bibliotecas de mediana complejidad y bibliotecas de alta complejidad

Mynor Fernández Morales



Protegido bajo licencia Creative Commons
Universidad de Costa Rica

Clasificación del *software* libre orientado a la automatización integral de bibliotecas según el nivel de complejidad de la biblioteca: bibliotecas simples, bibliotecas de mediana complejidad y bibliotecas de alta complejidad

Classification of free software oriented at the integral automation of libraries according to the level of complexity of the library: simple libraries, libraries of medium complexity and libraries of high complexity

*Mynor Fernández Morales*¹

RESUMEN

En este artículo se presenta una propuesta para clasificar el *software* orientado a la automatización integral de bibliotecas de acuerdo a los siguientes subniveles: una biblioteca simple, biblioteca de mediana complejidad o una bibliotecas de alta complejidad. Para clasificar a las bibliotecas en cada uno de estos subniveles, y con el fin de realizar el dimensionamiento correspondiente, se utilizan seis variables: tipo de biblioteca, el tamaño de la colección, tipo de materiales que maneja, cantidad y tipo de usuarios que atiende la biblioteca, el nivel de especialización del recurso humano que tiene la biblioteca y, finalmente, la capacidad técnica y económica que tenga la biblioteca. Además de clasificar las bibliotecas, también se clasifica el *software* libre orientado a la automatización, según los siguientes subniveles: *software* simple, *software* de mediana complejidad y *software* de alta complejidad. Para esta segunda clasificación, se utilizarán siete criterios: el nivel de robustez del motor de base de datos que soporta la aplicación, el nivel de parametrización, la plataforma donde se encuentra desarrollado el *software*, la clase del soporte técnico brindado a la aplicación informática, el nivel de facilidad para el desarrollo de nuevos requerimientos, la interoperabilidad del *software* y el manejo de estándares internacionales para importar y exportar información. Al final se presenta una lista clasificada de *software* orientado a la automatización integral de bibliotecas.

Palabras clave: automatización de bibliotecas, software libre, niveles de automatización

ABSTRACT

This article presents a proposal to classify oriented software automation of libraries according to the following sublevels: a simple library, library of medium complexity or high complexity libraries. To classify a library in each of these sublevels, six variables are used to perform the appropriate sizing, which are: type of library, the collection size, type of materials handled, number and type of users served by the library, the level of expertise of human resources that handles the library, and finally the technical and economic capacity to take into the library. Besides classifying libraries also is classified the free software oriented automation of these as the following sublevels: simple software, software of medium complexity and high complexity software. For this second classification, used seven criteria including: the level of robustness of the database engine that supports the application, the level of parameterization, the platform on which the software is developed, the type of support provided to the computer application, the level of ease for the development of new requirements, software interoperability, and finally the management of international standards to import and export information. At the end is included a list-oriented software classified the automation of libraries.

Keywords: Library automation, free software, automation levels.

Fecha de recibido: 29 oct. 2012 **Fecha de aprobado:** 9 nov. 2012

¹ Universidad de Costa Rica. Escuela de Bibliotecología y Ciencias de la Información. COSTA RICA.
mynor.fernandez@ucr.ac.cr

1. INTRODUCCIÓN

Este estudio del *software*, orientado a la automatización integral de bibliotecas, es un estudio realizado en el marco preliminar del desarrollo del proyecto de investigación de *software* libre para la automatización de bibliotecas que se está desarrollando en la Escuela de Bibliotecología y Ciencias de la Información de la Universidad de Costa Rica, e inscrito en la Vicerrectoría de Investigación, bajo el número 029 y titulado: “*Análisis del software libre disponible en la WEB en idioma español, orientado a la automatización de las unidades de información, en los tres niveles a saber: a. catálogos automatizados, b. repositorios digitales y c. sistematización integral de las funciones operativas de las unidades de información*”².

Hoy en día, las bibliotecas se enfrentan al reto de automatizar sus principales funciones, como adquisiciones, catalogación, circulación, inventario, referencia y servicio al cliente, entre otras; para esto, se enfrentan a dos posibilidades: utilizar *software* privado y con licenciamiento, o utilizar *software* público y libre. En el primer caso, tenemos que el *software* privado y licenciado es oneroso, y las bibliotecas cuentan con un bajo nivel presupuestario. Como señala Chinchilla (2011), este aspecto adquiere especial importancia, por cuanto las bibliotecas son casi siempre las últimas en contar con presupuesto para el desarrollo de proyectos de automatización. Por tanto, la alternativa de utilizar un *software* público y libre, es una excelente opción que se le presenta a las bibliotecas para automatizar, en forma integral, sus operaciones.

Las bibliotecas son organizaciones que presentan distintos niveles de complejidad, que puede ir desde un nivel sencillo, pasando por un nivel medio, hasta alcanzar un nivel de alta complejidad. Para realizar esta clasificación, se utilizarán seis variables, con el fin de establecer el nivel de complejidad de la biblioteca. Las variables son las siguientes: tipo de biblioteca, tamaño de la colección, cantidad y tipo de usuarios que maneja, nivel de especialización del recurso humano que tiene la biblioteca y, finalmente, la capacidad técnica y económica de la biblioteca. De acuerdo con esto, y por razones de conveniencia práctica, estableceremos tres subniveles de clasificación de las bibliotecas: nivel sencillo, nivel de mediana complejidad y nivel de alta complejidad.

Como primer punto, es recomendable que la biblioteca, que será objeto de automatización integral a través de *software* libre, se ubique en uno de los tres subniveles mencionados, según corresponda, con la finalidad de obtener un *software* apropiado que esté orientado a su automatización.

Asimismo, el *software* libre, orientado a la automatización integral de bibliotecas, también puede ser clasificado por su nivel de complejidad, que puede ir desde un nivel sencillo, pasando por un nivel medio, hasta alcanzar un alto nivel de complejidad. Para realizar esta clasificación, se utilizarán siete variables técnicas: el nivel de robustez del motor de base de datos que soporta la aplicación, el nivel de parametrización, la plataforma en que se encuentra desarrollado el *software*, la clase del soporte técnico brindado a la aplicación informática, el nivel de facilidad para el desarrollo de nuevos requerimientos, la interoperabilidad del *software* y, por último, el manejo de estándares internacionales para importar y exportar información.

² Proyecto B1183. Vicerrectoría de Investigación, Universidad de Costa Rica. <http://www.vinv.ucr.ac.cr/>

Al respecto, Hetzum (2002) concluye que cuanto más el sistema esté integrado en la ejecución de flujo trabajo de los usuarios, es posible que surjan problemas más severos, por lo que la implementación organizacional implica una complejidad extra, que requiere trabajo extra, y facilidades extra del sistema. Este nivel de complejidad comprende las actividades que preparan a las organizaciones y a los usuarios de un nuevo sistema, así como aquellas que preparan el sistema para el período de transición durante el cual entra en funcionamiento y toma el relevo de los sistemas anteriores. Por lo tanto, los ingenieros de *software* son propensos a experimentar problemas si tratan la implementación organizacional como un mero apéndice al proceso de desarrollo.

De esta forma, es claro que existen varios niveles de complejidad de *software* que van en concordancia con la complejidad de la organización que se desea automatizar. Así en el desarrollo e implantación de proyectos de sistemas, existe una relación directa entre el tamaño de la organización y la complejidad del *software*, siendo completamente válido afirmar que una organización pequeña generalmente es soportada por *software* sencillo, mientras que una organización grande es soportada por *software* más complejo. Relación que es válida con totalidad, ya que un *software* complejo, por lo general, requiere de mayor cantidad de recursos técnicos y humanos, que ofrezca mayor soporte técnico y mayor esfuerzo operativo para cumplir, en forma satisfactoria, con una funcionalidad más amplia. Es así como se puede afirmar que un *software* sencillo está orientado a satisfacer las necesidades de información de una biblioteca simple, mientras que un *software* complejo está orientado a satisfacer las necesidades de una biblioteca más compleja.

Por último, se hace una evaluación preliminar del *software* orientado a la automatización integral de bibliotecas, de acuerdo con los siguientes criterios técnicos: robustez del motor de base de datos, nivel de parametrización, plataforma informática del *software*, soporte técnico brindado, nivel de facilidad para el desarrollo de nuevos requerimientos, interoperabilidad del *software* y manejo de estándares internacionales para la importación y exportación de datos, para clasificar el *software* de acuerdo a su nivel de complejidad y obtener, finalmente, tres listas de *software* clasificado, según el nivel de complejidad de la biblioteca a que están orientados a automatizar.

2. SUBNIVELES DE AUTOMATIZACIÓN INTEGRAL DE BIBLIOTECAS

El presente artículo se enfocará en el tercer nivel de la automatización en una biblioteca. De acuerdo con Chinchilla (2011), existen tres niveles: automatización de catálogos, repositorios digitales y automatización integral de bibliotecas. Por lo tanto, en este artículo nos enfocaremos en el *software* libre en español de acceso público, dirigido a la automatización integral de bibliotecas.

Según Chinchilla “*el software libre se define como aquella aplicación que es adquirida sin pagar y es suministrada junto con el código fuente, el cual puede ser copiado, distribuido, modificado y adaptado a las necesidades de las personas usuarias*” (2011, p3).

La implantación de un paquete de *software* libre orientado a la automatización integral de bibliotecas, requiere determinar, de antemano, el nivel de complejidad de la biblioteca que será

objeto de la automatización, donde se establecieron tres subniveles, una biblioteca simple, una biblioteca de mediana complejidad o una biblioteca de alta complejidad.

Para hacer la clasificación de una biblioteca en cada uno de los subniveles mencionados, se proponen seis variables para clasificar la complejidad organizacional de una biblioteca, que va desde un nivel sencillo hasta un nivel complejo. Variables que surgen de la experiencia y sentido práctico, y que serán explicadas, una a una, para clarificar su relevancia y atino, para realizar esta clasificación.

La primera variable se refiere al tipo de biblioteca, que puede ser desde una sencilla biblioteca escolar hasta una compleja biblioteca pública.

Como segunda variable usaremos el tamaño de la colección, que implica el número de títulos que debe administrar la biblioteca, que igualmente puede ser pequeño en el orden de centenas de títulos, hasta alcanzar el orden de millares de títulos.

La tercera variable se refiere al tipo de materiales que maneja, que puede ser muy homogéneo, como solo libros y revistas, hasta un nivel completamente heterogéneo, que incluye libros, revistas, mapas, planos, material multimedia, material en la nube, y otros diversos tipos de materiales muy diversos, que convierten los procesos de la bibliotecas más técnicos y especializados.

Después, como cuarta variable, se utilizará la cantidad y tipo de usuarios que atiende la biblioteca, donde la cantidad también puede ir desde decenas de usuarios hasta millares de usuarios.

Como quinta variable, usaremos el nivel de especialización del recurso humano que se maneja en la biblioteca, donde se puede partir desde un solo bibliotecario, para el caso de una biblioteca escolar hasta equipos interdisciplinarios que administran y operan la biblioteca, tal como una biblioteca de una universidad pública.

Finalmente, como sexta variable se usará la capacidad técnica y económica, que se tenga en la biblioteca, donde se ubica el nivel de complejidad de la biblioteca, así podemos decir que en una biblioteca escolar existe un bajo nivel de capacidad técnica y económica, mientras que en una biblioteca universitaria pública existe un alto nivel de capacidad técnica y económica.

La clasificación de una biblioteca en cada uno de los subniveles mencionados: simple, de mediana complejidad o de alta complejidad, a través del estudio de las seis variables descritas anteriormente, será determinante para seleccionar el *software* libre orientado a la automatización integral de la biblioteca.

3. SUBNIVELES DE *SOFTWARE*

Además de la clasificación de la biblioteca según su tamaño, se deberá clasificar el *software* libre orientado a la automatización de bibliotecas, que también se clasificará en tres subniveles a saber: *software* simple, *software* de mediana complejidad o *software* de alta complejidad; para determinar la complejidad del *software* se utilizarán siete criterios que permitirán clasificar el *software* según su nivel de complejidad.

El primer criterio se refiere al nivel de robustez del motor de base de datos que soporta la aplicación, que puede ir desde el uso de simples programas manejadores de archivos hasta el uso de complejos motores de bases de datos.

Así, por ejemplo, un *software* simple, podríamos decir que es un *software* soportado con manejadores de archivos, tales como CDS/ISIS, FoxBase, OpenOffice Base, y otros similares cuyo sistema de archivos se caracteriza por la existencia de datos separados y aislados, donde la estructura física de los datos no es independiente de su estructura lógica, esto significa que las aplicaciones que utilizan este tipo de sistemas para depósito y recuperación de datos son dependientes de las estructuras de los mismos, lo que provoca que un cambio en uno de estos en su estructura física implica un cambio en todas las aplicaciones que lo utilizan, lo que incrementa los costos de mantenimiento de las aplicaciones para atender nuevas necesidades.

En suma, los sistemas de almacenamiento plano, que soporta este tipo de *software*, son poco robustos, es decir, presentan bajos niveles de confiabilidad, seguridad y un alto consumo de recursos, debido a que su funcionamiento implica altos niveles de consumo de ancho de banda, por donde la red circulará toda la información almacenada en ellos, con la finalidad de poder atender las diferentes transacciones del sistema.

Mientras que un *software* más complejo es soportado por motores de bases de datos, que son más robustos y, por tanto, con mayores niveles de confiabilidad, integridad y seguridad para la administración y utilización de los datos almacenados dentro de sus estructuras; además, los datos tienen independencia lógica de su estructura física, lo que implica que un cambio en la base de datos, no afecta a todas las aplicaciones que las utilizan, disminuyendo el costo del cambio para implementar nuevos requerimientos que impliquen transformaciones en las bases de datos.

Asimismo, con estos motores de bases de datos se puede utilizar una plataforma Cliente-Servidor, donde ya no viajan todos los datos almacenados por la red, como en un sistema de archivos planos descrito, sino solamente los datos correspondientes a cada transacción del sistema. Por ejemplo, dentro de los motores de bases de datos más robustos que utilizan las aplicaciones de *software* libre orientadas a la automatización integral de bibliotecas, tenemos los motores MySQL y PostgreSQL.

Es importante señalar que este criterio referente a la robustez del *software* no es tan significativo para clasificar un *software* como sencillo, mediano o complejo, porque la mayoría de estas aplicaciones enfocadas hacia la automatización integral de bibliotecas, hoy en día, utilizan motores de bases de datos, en lugar de sistemas de archivos. Sin embargo, sí es totalmente válido

afirmar, claramente y con toda seguridad, que una aplicación orientada a automatizar una biblioteca compleja tiene que estar soportada obligatoriamente por un motor de bases de datos robusto, y nunca por un sistema de archivos.

El segundo criterio se refiere al nivel de parametrización, donde tenemos que un *software* simple es concebido para atender necesidades muy puntuales, por lo que presenta un mínimo número de parámetros de configuración. La configuración paramétrica inicial del sistema permite prepararlo para distintos escenarios de funcionamiento, para adaptarlo de acuerdo a un conjunto de necesidades que presenta la biblioteca. Por lo tanto, a menor complejidad del *software*, tenemos un menor número de parámetros, ya que este tipo de *software* está orientado a satisfacer las necesidades de una biblioteca simple, que son más puntuales y sencillas.

Como ejemplo de una sencilla parametrización, podemos establecer parámetros muy básicos, como lenguaje de la aplicación, en qué idioma serán generadas las diferentes pantallas y reportes de la aplicación. Otro parámetro que se podría establecer serían los tipos de usuarios que tendrá que soportar la aplicación, así como los diferentes plazos de entrega del material que tendría cada tipo de usuario, según las necesidades organizacionales. Al igual que el nombre de la biblioteca y horario. Además, se podrían establecer diferentes montos de multa por día o por hora que se cobrará a cada tipo de usuario que haga entrega tardía de los materiales prestados. Y así sucesivamente se pueden establecer diferentes condiciones de funcionamiento vía parámetros de inicialización.

De igual manera, un *software* más complejo presenta un mayor número de parámetros, para atender distintos escenarios de funcionamiento que se presentan en bibliotecas de mayor complejidad, donde es importante destacar que, a partir de la parametrización que se haga del sistema en la etapa de su configuración, se pueden obtener diferentes escenarios de automatización, acorde con las necesidades que presenta el usuario y la organización.

De este modo, se da la posibilidad de que el mismo *software*, con diferente parametrización, atienda diferentes vistas de automatización con un mayor o menor número de funciones y reportes orientados a la atención de necesidades específicas. Así, la parametrización permite poner en funcionamiento sistemas con diferentes funcionalidades que se asemejan a aplicaciones distintas, pero que, en realidad, es la misma aplicación con distintos valores en sus parámetros de configuración inicial.

Como tercer criterio, tenemos la plataforma en que se encuentra desarrollado el *software*. En Wikipedia (Plataforma informática), existe la siguiente definición:

En informática, una plataforma es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible. Dicho sistema está definido por un estándar alrededor del cual se determina una arquitectura de hardware y una proforma de software (incluyendo entornos de aplicaciones). Al definir plataformas se establecen los tipos de arquitectura, sistema operativo, lenguaje de programación o interfaz de usuario compatibles.

Como primer ejemplo, bajo esta definición de una plataforma sencilla, puede constituir un simple computador de escritorio, funcionando como servidor, con un simple sistema operativo mono-

usuario, donde funciona una sencilla aplicación que solventa las necesidades de automatización para una pequeña biblioteca, soportada bajo un sistema de archivos. Este tipo de plataforma puede ser muy común en una pequeña biblioteca escolar, donde la aplicación podría ser el programa Open Marco Polo, que está orientado a satisfacer las necesidades de automatización de bibliotecas simples. De esta forma, para una biblioteca simple probablemente se utilizará una plataforma informática simple para la implementación y operación de la aplicación requerida por este tipo de bibliotecas. Aunque este tipo de plataforma no es recomendable técnicamente, debido a que un servidor por su naturaleza técnica debe ser un ordenado dedicado y nunca utilizarse como estación de trabajo, esto se presenta como un ejemplo común en muchas bibliotecas simples, donde por las limitaciones económicas no es posible tener un servidor dedicado.

Como segundo ejemplo, podríamos tener una plataforma más elaborada, donde se involucraría el uso de varios computadores, conectados a través de una red local, servidos por un servidor de red con un sistema operativo multiusuario y donde la aplicación utilizaría un motor de bases de datos robusto. Esta aplicación podría ser el *software* OpenBiblio.

Según Daniel (1996), la definición informática de cliente/servidor, se da cuando el cliente se supone que es un computador personal unido por una red a un servidor. Las solicitudes de servicios el cliente las realiza al servidor. Esto separa las funciones de manera que el cliente y el servidor, posiblemente, pueden ser más eficientes.

Finalmente, como tercer ejemplo, se tendría una plataforma más compleja, que involucra el uso de varios servidores, conectados a través de una red local, que interactúa con servidores remotos a través de la internet, utilizando sistemas operativos multiusuario más complejos e inteligentes, así como robustos motores de bases de datos. Estos dan soporte tanto a aplicaciones locales como aplicaciones remotas orientadas a la internet, utilizando tanto los servidores locales como los servidores virtuales, con modelos de distribución y almacenamiento de la información de acuerdo a las demandas y necesidades de los usuarios. En este último caso, se habla de una plataforma WEB. Para este caso podríamos utilizar las aplicaciones llamadas Koha o PMB.

De acuerdo con Coffman y otros (2010), las aplicaciones empresariales se están moviendo a la Web para una variedad de razones, incluyendo la facilidad de despliegue, gestión y consistencia y seguridad de los datos empresariales. El logro de un mejor rendimiento para los usuarios que se han acostumbrado a aplicaciones tradicionales requiere de nuevos marcos y metodologías en la Web. Este reto es agrava aún más cuando grandes cantidades de datos tienen que ser presentados visualmente y modificados dinámicamente como cambia el contexto del usuario.

Como cuarto criterio, se tiene la clase del soporte técnico brindado al *software*. El soporte brindado puede ser desde un simple manual en línea hasta la interacción virtual con una comunidad virtual a través de la internet. Para una aplicación sencilla, el uso de un manual en línea puede ser suficiente para soportar la automatización básica de una biblioteca, donde los problemas y requerimientos de soporte técnico se reducen a la comprensión y manejo de funciones simples y puntuales de fácil manejo. Sin embargo, es claro que para un *software* más complejo se requiere, además, de manuales en línea bien documentados, el apoyo de una comunidad virtual, cuya existencia es importante para la participación en foros virtuales con miembros de la comunidad, donde se resuelvan las diferentes consultas y problemas técnicos relacionados tanto

con la instalación del *software*, así como con el desarrollo de nuevos requerimientos y con el funcionamiento del mismo.

También, es importante señalar que este criterio, que se refiere al soporte técnico de la aplicación, tampoco es tan significativo para ponderar un *software* como sencillo, mediano o complejo. Sin embargo, sí es completamente válido afirmar que un *software* orientado a satisfacer las necesidades de automatización de una biblioteca compleja, es primordial y obligatorio que tenga un fuerte soporte técnico en línea, acceso a una comunidad de usuarios a través de la internet, con la disposición de foros virtuales, así como una formal y completa documentación del *software* tanto a nivel de usuario como a nivel técnico.

Como quinto criterio, tenemos el nivel de facilidad para el desarrollo de nuevos requerimientos. Este criterio nos permitirá establecer el nivel de complejidad para la incorporación de nuevas necesidades de la biblioteca. Podemos afirmar que un *software* bien documentado, desarrollado con un lenguaje de programación común y de uso universal, con la disponibilidad de los programas fuentes, presenta mayor facilidad para el desarrollo de nuevos requerimientos. Aquí es importante señalar que una aplicación sencilla, por estar orientada a resolver necesidades muy puntuales de una biblioteca sencilla, es menos probable que requiera cambios o adaptaciones, mientras que un *software* más complejo orientada a satisfacer las necesidades de una biblioteca más compleja, sí requiere, con mayor frecuencia, el desarrollo de nuevas funciones para solventar necesidades de información específicas, no cubiertas por el mismo, de ahí la importancia de este criterio concatenado con el de soporte técnico para contar con un fuerte apoyo logístico que facilite la incorporación de nuevas necesidades a través de la programación de las nuevas funciones que sean requeridas.

Como sexto criterio, nos ocupamos de la interoperabilidad del *software*, que mide la facilidad del sistema para interactuar con otros sistemas en uso en la biblioteca, sin mayor esfuerzo por parte del usuario. Es claro que, en una aplicación informática sencilla, el grado de interoperabilidad requerido es bajo, en el entendido de que difícilmente esta aplicación requerirá compartir información con otros sistemas, ya sea en la misma biblioteca o con bibliotecas externas. Así podemos concluir que una biblioteca sencilla, soportada por una aplicación informática sencilla, se presenta menor necesidad de interoperabilidad; mientras que una biblioteca compleja, generalmente, tiene la necesidad de una mayor interacción con un mayor número de aplicaciones, tanto dentro de la bibliotecas como en bibliotecas externas, que facilite el cumplimiento de la atención de las necesidades de información de sus usuarios.

Finalmente, tenemos el último criterio, que se refiere al manejo de estándares internacionales para importar y exportar información, desde la aplicación hacia otras aplicaciones y viceversa, donde es claro que a mayor número de manejo de diferentes estándares internacionales para importar y exportar dato, la aplicación es más compleja. Como ejemplo, ente los estándares internacionales que tenemos para la importación y exportación de datos, se encuentran: ISO2709, protocolo Z39.50, MARC21, MARCXML y SRU, XML, Dublin Core.

Resumiendo, en la Tabla N.º 1 presentamos un sencillo modelo de clasificación de *software*, que se utilizará para la clasificación en tres subniveles del *software* orientado a la automatización integral de bibliotecas, siendo el primer subnivel *software* simple, en el segundo subnivel *software* de mediana complejidad, y en el tercer subnivel *software* de alta complejidad; modelo

en el que se presentan los siete criterios de clasificación explicados anteriormente, y se utilizan tres valoraciones para calificar cada uno de estos siete criterios, para obtener, finalmente, la subclasificación propuesta para las aplicaciones orientadas a la automatización integral de bibliotecas.

Tabla N.º 1 Criterios para clasificar el software

Criterio de clasificación	Nivel		
	Bajo	Medio	Alto
Robustez Motor de Base de Datos	Bajo	Medio	Alto
Plataforma de <i>hardware</i> y <i>software</i>	Archivos	Cliente/servidor	Web
Nivel de Parametrización	Bajo	Medio	Alto
Facilidad de Desarrollo de nuevos requerimientos	Bajo	Medio	Alto
Interoperabilidad del <i>software</i>	Bajo	Medio	Alto
Manejo Estándares Internacionales	Bajo	Medio	Alto
Soporte Técnico	Bajo	Medio	Alto

Así utilizando este modelo, podemos obtener una clasificación, donde un *software* simple, para una biblioteca simple, se caracteriza, por lo general, por estar soportado por un sistema de archivos, en lugar de un motor de base de datos, aunque algunos de estas aplicaciones tienen interface WEB; para el manejo del OPAC su mayor debilidad sigue siendo que están basados en un sistema de archivos planos. Mientras que un *software* complejo para una biblioteca de alta complejidad está soportado por un motor de base de datos, para mayor seguridad e integridad de las transacciones que realiza. Asimismo, todos los demás criterios de evaluación técnica de este modelo son calificados con el nivel alto.

4. SOFTWARE PARA AUTOMATIZACIÓN INTEGRAL DE BIBLIOTECAS

Para una validación de este modelo, utilizaremos la lista de aplicaciones orientadas a la automatización integral de bibliotecas, que es producto del proyecto de investigación arriba señalado. Chinchilla y Fernández (2012) proporcionan la lista de *software* libre que se utilizará como base para realizar este proceso de clasificación por subniveles de las bibliotecas. Cada aplicación será sometida a este sencillo modelo de clasificación, con lo que se consiguen agrupar, según el tamaño de la organización a que está dirigido la aplicación, ya sea a: una biblioteca simple, una biblioteca de mediana complejidad o para una biblioteca de alta complejidad. A continuación, presentamos la lista de *software* clasificado, según estos tres subniveles de automatización para una biblioteca.

4.1 *Software* para bibliotecas simples

Software ABCD			
Criterio de clasificación	Nivel		
Robustez Motor de Base de Datos	CDS/ISIS		
Plataforma de <i>hardware</i> y <i>software</i>	Archivos		
Nivel de Parametrización	Bajo		
Facilidad de Desarrollo de nuevos requerimientos		Medio	
Interoperabilidad del <i>software</i>	Bajo		
Manejo Estándares Internacionales			Alto
Soporte Técnico	Bajo		
Página web: http://bvsmodelo.bvsalud.org/php/level.php?lang=es&component=27&item=13			
Descargas: http://bvsmodelo.bvsalud.org/php/level.php?lang=es&component=27&item=13			

Módulos del sistema:

OPAC.

Circulación / préstamos.

Adquisiciones.

Estadísticas.

Servicios bibliográficos, tales como SDI, impresión de código de barras, control de calidad.

Facilidad de desarrollo de nuevos requerimientos:

Capacidad muy amplia para definir FDT. Es posible usar subcampos repetibles, dar nombre a los subcampos y asociarlos a mensajes especiales de ayuda y tablas.

Construcción dinámica de planillas de ingreso de datos basada en las especificaciones de la FDT. Está desarrollado en el lenguaje de programación PHP y hay disponibilidad de fuentes.

Interoperabilidad del *software*:

Compatible con la tecnología CDS/ISIS: lectura de registros ISIS, lenguaje de formatos (PFT) e indización de registros (FST).

Manejo Estándares Internacionales:

Protocolo Z39.50

MARC21

Soporte técnico:

Lista de discusión

Software Aguapey			
Criterio de clasificación	Nivel		
Robustez Motor de Base de Datos	CDS/ISIS		
Plataforma de <i>hardware</i> y <i>software</i>	Archivos		
Nivel de Parametrización	Bajo		
Facilidad de Desarrollo de nuevos requerimientos			Alto
Interoperabilidad del <i>software</i>		Medio	
Manejo Estándares Internacionales			Alto
Soporte Técnico		Medio	
Página web: http://www.bnm.me.gov.ar/redes_federales/aguapey/			
Descarga: http://aguapeycordoba.wordpress.com/descargas/			

Módulos del sistema:

Base bibliográfica.
Base préstamo.
Base de usuarios.
Especificaciones.

Facilidad de desarrollo de nuevos requerimientos:

Esta desarrollado en lenguaje de programación Perl y hay disponibilidad de fuentes.

Interoperabilidad del *software*:

Utiliza OAI-PMH para aumentar su interoperabilidad.

Manejo Estándares Internacionales:

ISO 2709
TXT
MARC21
MARCXML

Soporte técnico:

Foros

<i>Software CAMPI</i>			
Criterio de clasificación	Nivel		
Robustez Motor de Base de Datos	CDS/ISIS		
Plataforma de <i>hardware</i> y <i>software</i>	Archivos		Web
Nivel de Parametrización	Bajo		
Facilidad de Desarrollo de nuevos requerimientos			Alto
Interoperabilidad del <i>software</i>	Bajo		
Manejo Estándares Internacionales			Alto
Soporte Técnico		Medio	
Página web: http://campi.uns.edu.ar/			
Descargas: http://sourceforge.net/projects/campi/files/			

Módulos del sistema:

Integra los módulos básicos de gestión bibliotecaria.

El módulo de catalogación se basa en Catalis.

Los módulos de circulación y estadísticas en Open MarcoPolo.

Facilidad de Desarrollo de nuevos requerimientos:

Está desarrollado en lenguaje de programación PHP y hay disponibilidad de fuentes.

Interoperabilidad del *software*:

Compatible con Catalis y Open Marcopolo.

Manejo Estándares Internacionales:

MARC21

MARCXML

SRU, XML

Soporte técnico:

Guías on-line / Preguntas frecuentes

Software Open Marcopolo			
Criterio de clasificación	Nivel		
Robustez Motor de Base de Datos	CDS/ISIS		
Plataforma de <i>hardware</i> y <i>software</i>	Archivos		
Nivel de Parametrización		Medio	
Facilidad de Desarrollo de nuevos requerimientos		Medio	
Interoperabilidad del <i>software</i>		Medio	
Manejo Estándares Internacionales		Medio	
Soporte Técnico		Medio	
Página web: http://marcopolo.uner.edu.ar/			
Descargas: http://marcopolo.uner.edu.ar/descarga.htm			

Módulos del sistema:

Módulo de consulta OPAC.
Módulo de circulación.
Módulo Administración.
Módulo Estadística.

Facilidad de Desarrollo de nuevos requerimientos:

Esta desarrollado en lenguaje de programación HTML y WXIS.

Interoperabilidad del *software*:

Trabaja íntegramente con bases de datos ISIS, lo que posibilita una compatibilidad total con Micro Isis y WinIsis.

Es capaz de interactuar con cualquier sistema de información que utilice el protocolo OAI-PMH.

Manejo Estándares Internacionales:

MARC21
CEPAL

Soporte técnico:

Ofrece información para contactar

Software PHL			
Criterio de clasificación	Nivel		
Robustez Motor de Base de Datos	CDS/ISIS		
Plataforma de <i>hardware</i> y <i>software</i>	Archivos		Web
Nivel de Parametrización		Medio	Alto
Facilidad de Desarrollo de nuevos requerimientos	Bajo		
Interoperabilidad del <i>software</i>	Bajo		
Manejo Estándares Internacionales			Alto
Soporte Técnico	Bajo		
Página web: http://www.elysio.com.br/			
Descargas: http://www.elysio.com.br/site/downloads.html			

Módulos del sistema:

OPAC.
Circulación.
Adquisiciones.
Diseminación Selectiva de la Información.

Facilidad de desarrollo de nuevos requerimientos:

Esta desarrollado en lenguaje de programación XML.

Interoperabilidad del software:

Interoperable con otros productos de la familia ISIS.

Manejo Estándares Internacionales:

Protocolo Z39.50
TXT
MARC21
MARCXML
SRU, XML

Soporte técnico:

Correo electrónico

<i>Software PhpMyLibrary</i>			
Criterio de clasificación	Nivel		
Robustez Motor de Base de Datos			MySQL
Plataforma de <i>hardware</i> y <i>software</i>			Web
Nivel de Parametrización	Bajo		
Facilidad de Desarrollo de nuevos requerimientos			Alto
Interoperabilidad del <i>software</i>	Bajo		
Manejo Estándares Internacionales			Alto
Soporte Técnico	Bajo		
Página web: http://code.google.com/p/phpmylibrary/			
Descargas: http://code.google.com/p/phpmylibrary/			

Módulos del sistema:

Catalogación.
Circulación.
OPAC.

Facilidad de desarrollo de nuevos requerimientos:

Posee lenguaje de programación PHP.

Interoperabilidad del *software*:

Es compatible con el estándar de USMARC.

Manejo Estándares Internacionales:

ISO 2709
Protocolo Z39.50
MARC21
MARCXML
SRU, XML

Soporte técnico:

Correo electrónico

4.2 Software para bibliotecas de mediana complejidad

<i>Software Emilda</i>			
Criterio de clasificación	Nivel		
Robustez Motor de Base de Datos			MySQL
Plataforma de <i>hardware</i> y <i>software</i>			Web
Nivel de Parametrización	Bajo		
Facilidad de Desarrollo de nuevos requerimientos			Alto
Interoperabilidad del <i>software</i>	Bajo		
Manejo Estándares Internacionales			Alto
Soporte Técnico	Bajo		
Página web: http://www.emilda.org/			
Descargas: http://sourceforge.net/projects/emilda/files/latest/download?source=files			

Módulos del sistema:

Circulación.
Catalogación.
Administración general y de usuarios.
OPAC.

Facilidad de desarrollo de nuevos requerimientos:

Posee PHP como lenguaje de programación.
Ofrece la posibilidad de adaptar su interface gráfica web fácilmente.

Interoperabilidad del *software*:

No presenta interoperabilidad.

Manejo Estándares Internacionales:

Protocolo Z39.50
TXT
MARC21
MARCXML
SRU, XML

Soporte técnico:

Guías en línea

Software Infocid			
Criterio de clasificación	Nivel		
Robustez Motor de Base de Datos			PostgreSQL
Plataforma de <i>hardware</i> y <i>software</i>			Web
Nivel de Parametrización		Medio	
Facilidad de Desarrollo de nuevos requerimientos			Alto
Interoperabilidad del <i>software</i>	Bajo		
Manejo Estándares Internacionales	Bajo		
Soporte Técnico			Alto
Página web: http://sourceforge.net/projects/infocid/			
Descargas: http://sourceforge.net/projects/infocid/			

Módulos del sistema:

Módulo de procesos.
Módulo de consulta.
Módulo de préstamos y devoluciones.
Módulo de estadísticas.
Módulo de inventario.

Facilidad de desarrollo de nuevos requerimientos:

Esta desarrollado en lenguaje de programación PHP.

Interoperabilidad del *software*:

No presenta interoperabilidad.

Manejo Estándares Internacionales:

Protocolo Z39.50
MARC21

Soporte técnico:

Comunidad virtual

Software OpenBiblio			
Criterio de clasificación	Nivel		
Robustez Motor de Base de Datos			PostgreSQL
Plataforma de <i>hardware</i> y <i>software</i>			Web
Nivel de Parametrización			Alto
Facilidad de Desarrollo de nuevos requerimientos		Medio	
Interoperabilidad del <i>software</i>	Bajo		
Manejo Estándares Internacionales	Bajo		
Soporte Técnico			Alto
Página web: http://obiblio.sourceforge.net/			
Descargas: http://sourceforge.net/projects/obiblio/files/OpenBiblio/0.7.1/			

Módulos del sistema:

Circulación.
 Administración del sistema.
 Informes.
 Administración de personal.
 OPAC.

Facilidad de desarrollo de nuevos requerimientos:

Esta desarrollado en lenguaje de programación PHP.

Interoperabilidad del *software*:

No presenta interoperabilidad.

Manejo Estándares Internacionales:

USAMARC
 MARC21

Soporte técnico:

Comunidad virtual

4.3 *Software* para bibliotecas de alta complejidad

<i>Software Koha</i>			
Criterio de clasificación	Nivel		
Robustez Motor de Base de Datos			MySQL / PostgreSQL
Plataforma de <i>hardware</i> y <i>software</i>			Web
Nivel de Parametrización			Alto
Facilidad de Desarrollo de nuevos requerimientos			Alto
Interoperabilidad del <i>software</i>			Alto
Manejo Estándares Internacionales			Alto
Soporte Técnico			Alto
Página web: http://www.koha.org			
Descargas: http://www.koha.org/liblime-koha			

Módulos del sistema:

Adquisiciones.

Catalogación.

Circulación.

OPAC.

Koha tiene una serie de parámetros básicos y parámetros adicionales, como las impresoras y los servidores Z39.50 para configuración del sistema de acuerdo a las necesidades del usuario.

Facilidad de desarrollo de nuevos requerimientos:

Esta desarrollado en lenguaje de programación Perl.

Interoperabilidad del *software*:

KOHA está desarrollado usando las normas, estándares y protocolos que garanticen la interoperabilidad entre KOHA y otros sistemas y tecnologías (incluyendo dispositivos móviles), mientras soporta los flujos de trabajo y herramientas existentes.

Manejo Estándares Internacionales:

ISO 2709

Protocolo Z39.50

MARC21

Soporte técnico:

Comunidad virtual

Software PMB			
Criterio de clasificación	Nivel		
Robustez Motor de Base de Datos			MySQL
Plataforma de <i>hardware</i> y <i>software</i>			Web
Nivel de Parametrización			Alto
Facilidad de Desarrollo de nuevos requerimientos			Alto
Interoperabilidad del software			Alto
Manejo Estándares Internacionales			Alto
Soporte Técnico			Alto
Página web: http://www.sigb.net/			
Descargas: http://www.software112.com/products/pmb.html			

Módulos del sistema:

Circulación.

Catálogo.

Autoridades.

OPAC.

Diseminación Selectiva de la Información (DSI).

Informes.

Facilidad de desarrollo de nuevos requerimientos:

Posee lenguaje de programación PHP y disponibilidad de fuentes.

Interoperabilidad del *software*:

Interoperabilidad con ABCD

Manejo Estándares Internacionales:

Protocolo Z39.50

MARC21

MARCXML

Soporte técnico:

Comunidad virtual

5. CONCLUSIONES

Siendo este artículo un resultado parcial del proyecto de investigación que se desarrolla en la Escuela de Bibliotecología de la Universidad de Costa Rica, se debe tener presente que luego de una evaluación más profunda de cada uno de estos programas, según se lo propone el mismo proyecto, algunos de los datos aquí expuestos podrían ser ampliados o complementados para darle una mayor profundidad y riqueza a los mismos. Asimismo, este proyecto se puede convertir en una plataforma de lanzamiento de futuras proyectos de investigación relacionados con este tema.

Bajo este marco, esta lista de *software* orientado a la automatización integral de bibliotecas, no pretende ser un listado único y exhaustivo, por cuanto el desarrollo tecnológico hoy en día es vertiginoso, y periódicamente se libera *software* nuevo para este mismo fin, sino que con esta lista se pretende ofrecer al lector una gama de posibles aplicaciones para la automatización integral de una biblioteca.

Se ha determinado que existe una gran variedad de *software* libre orientado a la automatización integral de bibliotecas, que se ha clasificado en tres subniveles, bibliotecas pequeñas, bibliotecas medianas y bibliotecas grandes, donde es importante señalar que la complejidad del *software* así como los requisitos técnicos y humanos para implantar un *software* para una biblioteca grande supera, con creces, los requeridos para una biblioteca pequeña, de ahí la importancia de seleccionar el *software* dependiendo del nivel en que se ubique la organización, con la finalidad de no tener una aplicación subutilizada, en el caso de utilizar un *software* complejo para una biblioteca pequeña, ni una aplicación deficiente, en el caso de utilizar *software* simple para una biblioteca grande, con el propósito de cubrir las necesidades organizacionales.

La clasificación del *software* orientado a la automatización integral de bibliotecas, dada en este artículo, según el tamaño de la organización a que va dirigido, ya sea una biblioteca pequeña, mediana o grande, tiene la finalidad de convertirse en una guía técnica preliminar de selección y evaluación de primera mano, para las bibliotecas que están planeando utilizar un *software* de este tipo. Pero no pretende, de ninguna forma, sustituir el trabajo de análisis recomendado para determinar, finalmente, cuál es el *software* apropiado que satisfaga los requerimientos técnicos y organizacionales de cada biblioteca en particular.

Para terminar, el nivel de recursos tecnológicos con que cuente una biblioteca, tanto humano como de equipo, es un fundamental en la toma de decisiones sobre el *software* a utilizar para resolver sus necesidades de información. En general, podemos afirmar que las bibliotecas tienen pocos recursos tecnológicos, debido a sus bajos presupuestos económicos, donde el problema es más crítico para bibliotecas pequeñas, en las que existe mayor deficiencia de estos recursos informáticos y donde, generalmente, observamos que existe un bajo nivel de automatización, que se justifica por la existencia de estas limitaciones. De aquí surge una recomendación fundamental de este artículo: la utilización de un *software* libre orientado a bibliotecas pequeñas, que facilite la automatización integral de este tipo de bibliotecas con una inversión económica baja, tanto en recursos económicos como en esfuerzo y, con ello, se contribuya a aumentar los niveles de automatización en las bibliotecas.

6. BIBLIOGRAFÍA

Chinchilla-Arley, Ricardo (2011). El software libre: una opción para la automatización de unidades de Información. *Bibliotecas*. 29, 1.

Chinchilla-Arley, Ricardo; Fernández-Morales, Mynor (2012). Bibliotecas automatizadas con software libre: establecimiento de niveles de automatización y clasificación de las aplicaciones. *Bibliotecas*. 30, 2.

Coffman, Daniel; Soroker, Danny; Narayanaswami, Chandra; Zinman, Aaron. (2010). A client-server architecture for state-dependent dynamic visualizations on the web. Proceedings of the 19th international conference on World wide web (WWW '10). *ACM*, 1237-1240. DOI=10.1145/1772690.1772891

Daniel, S. E. (1996). Client/server computing. *Journal Of Systems Management*, 47(3), 58.

Hertzum, Morten (2002). Organisational implementation: a complex but under-recognised aspect of information-system design. In Proceedings of the second Nordic conference on Human-computer interaction (NordiCHI '02). *ACM*, 201-204. DOI=10.1145/572020.572045

Plataforma (informática). Wikipedia. Fuente: <http://es.wikipedia.org/w/index.php?oldid=60169416> Contribuyentes: Acratta, Adatzb, Camilo, Dianai, Diegusjaimes, Echani, Farisori, Folga, GermanX, Hb2, Isha, Linfocito B, Matdrodes, Paintman, Technopat, UA31, Vitamine, 32 ediciones anónimas