



# **Universidad de las Ciencias Informáticas**

## **Facultad 2**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

# **Diseño e Implementación del Módulo**

## **Adquisición del Sistema ABCD 3.0**

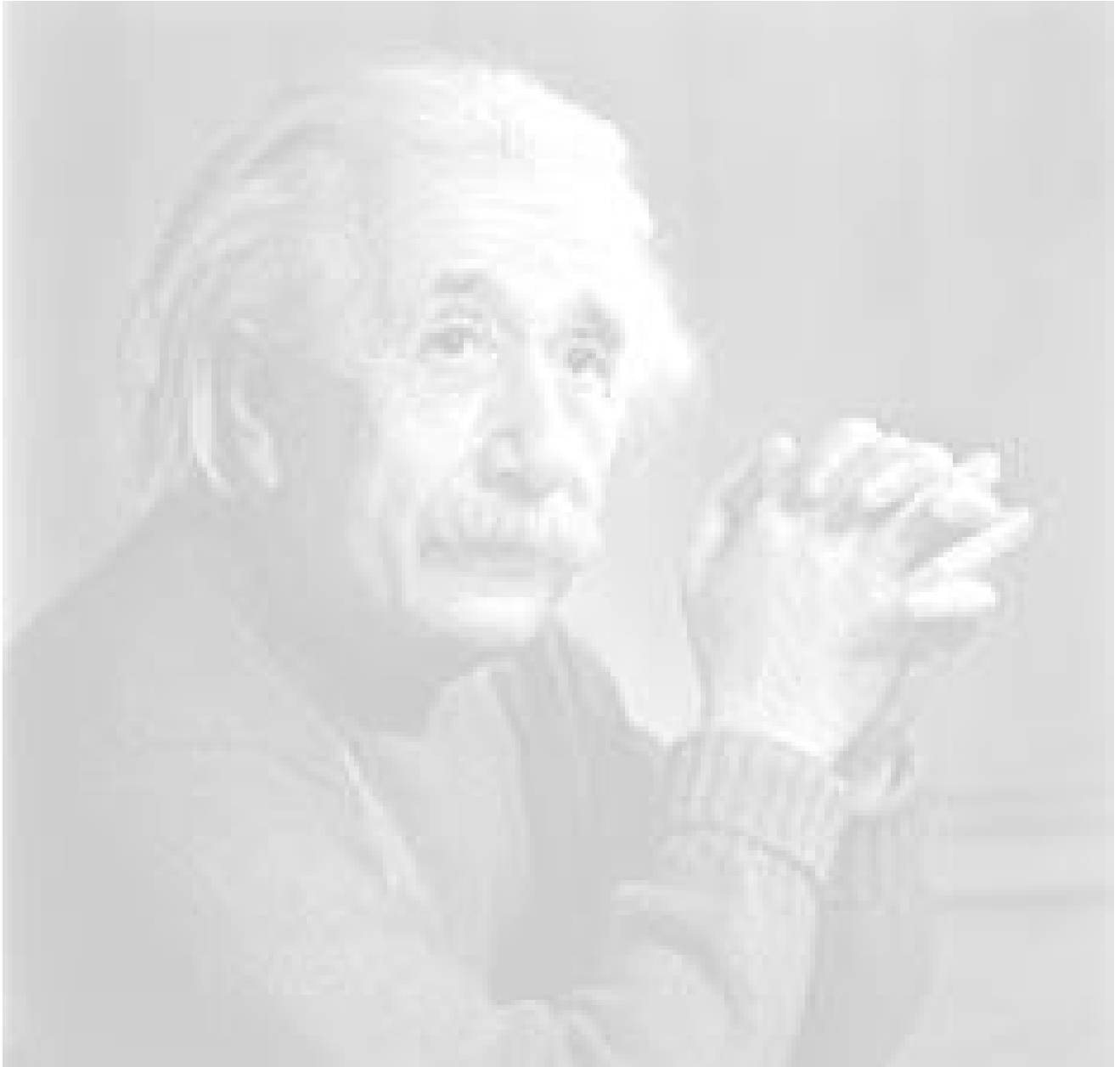
**Autor(es):** Gisselle Laffita Morales

Juan Miguel Pérez Leal

**Tutor:** Ing. Yaksel Duran Rivas

**Co-Tutor:** Ing. Reynier Carbonell Sánchez

“La Habana, junio 2015”



“Nunca consideres el estudio como una obligación, sino como la oportunidad para penetrar en el bello y maravilloso mundo del saber.”

Albert Einstein

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de la presente investigación y autorizamos al Centro de Informatización y Gestión Documental (CIGED) de la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio.

Para que así conste firmamos a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Firma del autor

---

Firma del autor

---

Firma del tutor

---

Firma del co-tutor

## Agradecimientos

A “mita” y “pito” (mis padres), las personas más importantes en mi vida, por su amor y cariño, por jugar en mi vida el rol de padres perfectos, amigos y confidentes, por guiarme siempre por el buen camino, por apoyarme en las decisiones que he tomado en mi vida. Si alguna vez no les he dicho cuanto los amo, cuanto los quiero y cuanto los necesito, quiero que lo sepan en este momento. Todo lo que he sido y en lo que hoy me convierto se lo debo a ustedes. Los quiero con la vida.

A mi hermanito del alma Ricardo (el único) por existir.

A mi Tía Yanelis por ser la mejor tía del mundo, por quererme tanto y consentirme en todo, por sus consejos, por ser mi guía, por ayudarme cuando más lo he necesitado. Gracias por ser esa persona especial que da el amor de una madre, el apoyo de una hermana y la comprensión de una amiga, gracias por existir...Te adoro.

A mi “hermana” Osmel por ser mi amigo, compañero y confidente, por estar ahí para mí en los momentos buenos y malos, por todo su apoyo incondicional, gracias por tu amistad, te quiero mucho.

A mi compañero de tesis Juan Miguel por aguantarme durante todo un semestre, seguro para el pareció más tiempo (jaja), sin él no habría sido posible este logro.

*A mis tutores Yaksel y Reynier, por la ayuda brindada, por aclarar mis dudas cuando podían.*

*Gracias.*

*A mi primo Jorge Luis (El Chiquitico) que a pesar de estar lejos siempre me dio ánimos en sus mensajes para seguir adelante.*

*En fin, a todas las personas que de una forma u otra posibilitaron la realización de este trabajo.*

## Agradecimientos

*Antes que nada quiero dedicar este resultado a mi abuela, que no se encuentra conmigo. Agradezco a todas las personas que de una manera u otra contribuyeron a mi formación como ingeniero, en especial a mi madre, mi tía y mis hermanos que siempre tuvieron presentes tanto en los momentos malos como en los buenos.*

*Al Ingeniero Leinys Amel Pons que me ayudo en la elaboración y revisión de la tesis. A todos los amigos que hice en la carrera, los viejos, los nuevos, a los que no continuaron pero siempre estuvieron presentes y los que continuaron hasta el final, especialmente a los amigos de primer año y a mi grupo.*

*Un agradecimiento especial a mi amigo, y más que amigo mi hermano Alexei Darias. A mi compañera de tesis y a mis tutores que batallaron tanto conmigo para que este trabajo saliera.*

*Gracias a todos.*

## ATOS DE CONTACTO

**Tutor:** Ing. Yaksel Durán Rivas.

Graduado de Ingeniero en Ciencias Informáticas en el año 2011, se desempeña actualmente como Especialista "B" en Ciencias Informáticas. Tiene cuatro años de experiencia en desarrollo de aplicaciones desempeñando el rol de desarrollador.

**Co-Tutor:** Ing. Reynier Carbonell Sanchez.

Graduado de Ingeniero en Ciencias Informáticas en el año 2014, desempeña actualmente el rol de desarrollador en el proyecto ABCD. Tiene 1 años de experiencia en el desarrollo de software.

### Resumen

Las bibliotecas y centros de documentación<sup>1</sup>, son organizaciones cuya función consiste en mantener una colección de materiales bibliográficos y facilitar mediante los servicios del personal, el uso de documentos necesarios para satisfacer las necesidades de información, investigación, educación y ocio de sus lectores. A medida que transcurre el tiempo, necesitan perfeccionar sus servicios, dado el volumen de información que circula y el creciente número de usuarios que acceden a él. La aparición de las Tecnologías de Información y la Comunicación (TIC), posibilita informatizar los procesos que se llevan a cabo en estos centros. La Universidad de Ciencias Informáticas (UCI), en particular el Centro de Informatización de la Gestión Documental (CIGED), trabajó en el desarrollo del sistema de Automatización de Bibliotecas y Centros de Documentación versión 3.0 (ABCD versión 3.0) (entre otros módulos, Adquisición); cuya función básica es la de informatizar los procesos que se llevan a cabo en las bibliotecas. En el módulo Adquisición del sistema ABCD versión 3.0 se corrigieron deficiencias arrojadas en evaluaciones realizadas a la versión 1.2 de dicho sistema, deficiencias que dificultan los procesos que se realizan en las bibliotecas. El objetivo de la presente investigación es diseñar e implementar el módulo Adquisición del sistema ABCD versión 3.0, para perfeccionar la gestión del proceso de Adquisición de materiales bibliográficos en dicho sistema.

**Palabras claves:** ABCD, adquisición, bibliotecas, materiales, Orden de Compra, Desiderata, Registro de Adquisición, Registro de Adquisición de un Ejemplar, Sistema Integrado de Gestión Bibliotecaria (SIGB), Solicitud de Compra.

---

<sup>1</sup> Bibliotecas y Centros de Documentación: definidos, según la norma ISO 2789-1991 (Norma UNE-EN ISO 2789) sobre estadísticas internacionales de bibliotecas.



Contenido	
<b>INTRODUCCIÓN</b> .....	<b>13</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>18</b>
<b>Introducción</b> .....	<b>18</b>
<b>1.1 Proceso de adquisición en las bibliotecas</b> .....	<b>18</b>
<b>Vías de Adquisición</b> .....	<b>18</b>
<b>Vía de Adquisición comercial</b> .....	<b>18</b>
<b>Vías de Adquisición no comerciales</b> .....	<b>18</b>
<b>Sistemas Integrados de Gestión Bibliotecaria</b> .....	<b>19</b>
<b>1.2 Sistemas existentes</b> .....	<b>19</b>
<b>Koha</b> .....	<b>19</b>
<b>PHPMyBibli</b> .....	<b>20</b>
<b>ABCD versión 1.2</b> .....	<b>21</b>
<b>Principales características en las que coinciden los anteriores SIGB</b> .....	<b>22</b>
<b>Comparación del módulo Adquisición de los SIGB (12)</b> .....	<b>23</b>
<b>1.3 Metodología de desarrollo de software y lenguajes</b> .....	<b>24</b>
<b>1.3.1 Metodología</b> .....	<b>24</b>
<b>1.3.2 Lenguaje de modelado UML versión 2.0</b> .....	<b>25</b>
<b>1.3.3 Lenguaje de desarrollo Java</b> .....	<b>25</b>
<b>1.4 Herramientas utilizadas</b> .....	<b>26</b>

## Contenido

1.4.1 Visual Paradigm versión 8.0.....	26
1.4.2 Eclipse versión Kepler Release 4.3 .....	26
1.4.3 Gestores de Bases de Datos.....	27
PostgreSQL versión 9.3 .....	27
J-ISIS versión Septiembre 2014.....	27
1.5 Tecnologías.....	28
1.5.1 Eclipse Virgo versión 3.6.3.....	28
1.5.2 RAP versión 3.0M5.....	29
1.5.3 OSGi Equinox versión 3.8.2 .....	29
1.5.4 Spring Dynamic Modules versión 3.1.0.....	30
1.5.5 EclipseLink versión 2.3 .....	31
Conclusiones del capítulo.....	31
<b>CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DE LA SOLUCIÓN .....</b>	<b>33</b>
Introducción.....	33
2.1 Propuesta de Solución .....	33
Descripción de Procesos .....	33
2.2 Descripción de Actores .....	34
2.3 Resumen de los casos de uso del módulo Adquisición .....	35
2.4 Arquitectura .....	38
2.5 Patrones de diseño GRASP .....	40

## Contenido

2.5.1 Experto .....	40
2.5.2 Creador .....	40
2.5.3 Bajo Acoplamiento .....	40
2.5.4 Alta Cohesión.....	40
2.5.5 Controlador .....	41
2.6 Patrones de diseño GOF .....	41
2.7 Diagrama de Clases del Diseño .....	41
2.8 Diagrama de Secuencia.....	43
2.9 Modelo de Base de Datos.....	44
2.9.1 PostgreSQL .....	45
2.9.2 Base de Datos Adquisición (J-Isis).....	45
2.9.3 Base de Datos Ejemplar (J-Isis).....	46
Conclusiones del capítulo.....	47
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DE LA SOLUCIÓN.....</b>	<b>48</b>
3.1 Implementación.....	48
3.2 Diagramas de componentes. ....	48
3.2 Diagrama de despliegue.....	49
3.3 Pruebas. ....	50
Conclusiones del capítulo.....	65
CONCLUSIONES .....	66
RECOMENDACIÓN .....	67

## *Contenido*

REFERENCIAS BIBLIOGRAFICAS.....	68
GLOSARIO DE TÉRMINOS .....	72
ANEXOS.....	73
<b>Anexo 1: Preguntas de la Entrevista al Cliente. ....</b>	<b>73</b>

## ÍNDICE DE TABLAS

Tabla 1 Comparación de SIGB. ....	23
Tabla 2 Descripción de Procesos para Adquirir Material Bibliográfico.....	34
Tabla 3 Descripción de Actores del sistema. ....	35
Tabla 4 Comportamiento de Casos de Usos.....	38
Tabla 5 Modelado de la Base de Datos Adquisición. ....	46
Tabla 6 Modelado de la BD Ejemplar.....	47
Tabla 7 Caso de Prueba Gestionar Registro de Adquisición, SC1 Registrar Registro de Adquisición..	57
Tabla 8 Descripción de las variables.....	57
Tabla 9 Resultados de las pruebas de Caja Negra aplicadas al módulo Adquisición. ....	58

## ÍNDICE DE FIGURAS

Figura 1 Fases de RUP. ....	25
Figura 2 Macro-Proceso para Adquirir un Material.....	33
Figura 4 Diagrama de Clases del Diseño Gestionar Registro de Adquisición. ....	43
Figura 5 Diagrama de Secuencia Registrar Registro de Adquisición. ....	44
Figura 6 Modelado de la BD en PostgreSql. ....	45
Figura 7 Diagrama de Componentes del CU Gestionar Registro de Adquisición, perteneciente al Módulo Adquisición del sistema ABCD versión 3.0. ....	49
Figura 8 Método de Prueba, Caja Negra.....	51
Figura 9 Ejemplo de utilización de beans.....	60
Figura 10 Carpeta de resources del sistema ABCD versión 3.0.....	61
Figura 11 Contenido de la carpeta usr del sistema ABCD versión 3.0. ....	63
Figura 12 Archivo .xml encontrado en el directorio servidor\abcdconfig\menú\menu.xml.....	64

## INTRODUCCIÓN

Los adelantos tecnológicos en el sector de la informática, han influido de manera decisiva en el desarrollo del proceso docente educativo, y lo han renovado de manera sustancial en diversas aristas que lo integran. El advenimiento de las tecnologías de la información, ha hecho posible que las bibliotecas incorporen herramientas para satisfacer necesidades informativas de los usuarios. Mediante la introducción de servicios automatizados que perfeccionen los procesos de gestión de materiales bibliográficos en las instituciones, se facilita la accesibilidad a la información almacenada (1) (2).

Según recoge el Diccionario de la Real Academia de la Lengua Española, en una de sus acepciones, “biblioteca”, es una institución cuya finalidad consiste en la adquisición, conservación, estudio y exposición de libros y documentos (3). Debido al incremento del flujo de información y aumento de la producción bibliográfica, dichas instituciones, requieren ampliación en su capacidad de almacenamiento, como consecuencia de ello, es necesario mayor nivel de organización y aplicación de procedimientos más sofisticados en aras de solventar los problemas que emanan en dicho campo (4).

Ben White<sup>2</sup>, en su artículo “La función que desempeñan las bibliotecas para garantizar el acceso a conocimientos” planteó que se consideran puertas de acceso a los conocimientos, a la cultura y desempeñan una función fundamental en la sociedad. Los recursos y los servicios que ofrecen dan la oportunidad de aprender, sirven como apoyo a la alfabetización, a la educación, y ayudan a dar forma a las nuevas ideas y perspectivas que son vitales dentro de una sociedad creativa e innovadora. Asimismo, garantizan la existencia de un registro auténtico de los conocimientos creados y acumulados por las generaciones pasadas. Si no existieran las bibliotecas, sería difícil avanzar en la investigación y los conocimientos humanos, preservar los conocimientos acumulados y el patrimonio cultural para generaciones futuras (4). El sentido del término “biblioteca” hace referencia a cualquier recopilación de datos recogida en formatos como: microfilms, revistas, grabaciones, películas, diapositivas, cintas magnéticas y de vídeo, así como otros medios electrónicos (5). Dichos medios responden en términos de calidad a las necesidades educativas, investigativas y culturales de los usuarios y comunidad en general.

López Jiménez<sup>3</sup> y Alfonso Sánchez<sup>4</sup>, en su artículo “Las bibliotecas al comienzo del siglo XXI”, hicieron referencia al desarrollo de las TIC en el contexto bibliotecario y el uso generalizado de Internet;

---

<sup>2</sup> Ben White: Jefe de Propiedad Intelectual, *British Library. Organización Mundial de Propiedad Intelectual (OMPI)*.

<sup>3</sup> Licenciada en Información Científico -Técnica y Bibliotecología. Centro de Documentación. Oficina Panamericana de la Salud-Cuba (OPS).

<sup>4</sup> Máster en Informática Médica. Profesor Auxiliar. Docencia e Investigaciones. Centro Nacional de Información de Ciencias Médicas-Infomed.

## Introducción

mencionan que dichas transformaciones cambian el entorno de trabajo de las bibliotecas y centros de documentación respecto al modo de hacer los procesos y prestar servicios, lo que condujo a la aparición de las llamadas bibliotecas electrónicas, digitales y virtuales. El bibliotecario o profesional de la información, está llamado a incorporar a su perfil de trabajo nuevas funciones, entre ellas, acercar el uso de las TIC a aquellos usuarios que por diferentes razones no las manejan. La “transgresión<sup>5</sup>” de las TIC en estos escenarios, suscitó la automatización de las bibliotecas y centros de documentación, también tributó a la aparición de nuevas formas de trabajo y nuevos servicios. Un ejemplo de estos avances, lo constituye el surgimiento de los Sistemas Integrados de Gestión Bibliotecaria (SIGB), herramientas tecnológicas que permiten automatizar las operaciones bibliotecarias (6).

Típicamente un SIGB abarca la adquisición, catalogación, circulación y consulta de materiales. La tendencia de los SIGB consiste en ampliar su oferta con nuevos productos o módulos integrados en sus sistemas de automatización (por ejemplo, sistemas de resolución de enlaces, metabuscadores, gestores de recursos electrónicos, portales web de consulta) (7). Un ejemplo es el Sistema ABCD versión 1.2, que ofrece herramientas para la gestión de bases de datos (por ejemplo: bibliográficas/documentales, tales como adquisiciones y préstamos), entrada de datos, estadísticas, la circulación, control de publicaciones periódicas y funciones de búsqueda.

En la UCI, específicamente en el CIGED, perteneciente a la facultad 2, se realizó una evaluación del módulo Adquisición del sistema ABCD versión 1.2. El mismo arrojó deficiencias que dificultan el mejor desempeño de los procesos que se realizan en las bibliotecas; a continuación se listan las más significativas:

- Inadecuado flujo de pasos para realizar la Sugerencia<sup>6</sup>. La Sugerencia debe pasar por diferentes estados de forma obligatoria: aprobación, licitación y decisión. Luego, se realiza la Orden de Compra, aunque no exista compra alguna para poder registrar.
- Los usuarios para realizar una Sugerencia deben ir hasta la biblioteca, ya que no tienen la posibilidad de acceder a esta funcionalidad por la red.
- La adquisición de un nuevo material bibliográfico debe tener asociada obligatoriamente una Sugerencia de adquisición para ser registrada en el sistema, o sea, el bibliotecario debe también realizar una Sugerencia de un material bibliográfico como un usuario común.
- En el momento de rechazar una Sugerencia el sistema no ofrece variedad de motivos necesarios por los cuales este desea rechazar la Sugerencia.

---

<sup>5</sup> Violación de un precepto, de una ley o de un estatuto. (46)

<sup>6</sup> Idea que se sugiere sobre un material bibliográfico.

## Introducción

- En el momento de aprobar una Sugerencia el sistema no permite conformar directamente una Desiderata<sup>7</sup>.
- El sistema no permite crear una Solicitud de Compra antes de crear la Orden de Compra.
- El flujo de adquisición de un material bibliográfico no es el que se realiza en las bibliotecas (Ver Figura 2: Flujo para adquirir un material bibliográfico).

Teniendo en cuenta las deficiencias arrojadas en la evaluación realizada al módulo Adquisición del sistema ABCD versión 1.2, se ratificó la necesidad de corregir dichas deficiencias mediante el desarrollo de una nueva versión del sistema, lo que permitirá el mejor desempeño de los procesos de gestión de materiales bibliográficos que se lleva a cabo en las bibliotecas. Teniendo en cuenta la situación problemática anterior, se identifica como **problema a resolver**:

¿Cómo perfeccionar el proceso de gestión de la adquisición de materiales bibliográficos en el sistema ABCD?

El problema a resolver lleva a los autores de la investigación a definir como **objeto de estudio**: El proceso de gestión de la adquisición de materiales bibliográficos en los SIGB.

Definiendo como **campo de acción**: El proceso de gestión de la adquisición de materiales bibliográficos en el sistema ABCD.

Para la solución del problema se planteó como **objetivo general**: Diseñar e implementar el módulo Adquisición del sistema ABCD versión 3.0 perfeccionando el proceso de gestión de la adquisición de materiales bibliográficos en dicho sistema.

Para el desglose del objetivo general se definieron los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación para proveer un sistema coordinado y coherente de conceptos y proposiciones que permitan abordar el problema.
2. Diseñar el módulo Adquisición del sistema ABCD versión 3.0 con el objetivo de desarrollar las directrices propuestas durante el análisis para satisfacer los requisitos funcionales y no funcionales del sistema.
3. Implementar el módulo Adquisición del sistema ABCD versión 3.0 y verificar que cumple los requerimientos funcionales y no funcionales.

---

<sup>7</sup> Hoja que se emplea en las bibliotecas para sugerir la adquisición de nuevas obras. (45)



## *Introducción*

Para dar cumplimiento a los objetivos específicos de esta investigación se definen las siguientes **tareas de la investigación**:

1. Análisis de soluciones nacionales e internacionales enmarcadas en el objeto de estudio para identificar funcionalidades que se puedan incorporar al módulo de Adquisición del sistema ABCD versión 3.0.
2. Descripción de las herramientas y tecnologías para el desarrollo del módulo Adquisición del sistema ABCD versión 3.0.
3. Diseño de los diagramas de clases e interacción para el desarrollo del módulo Adquisición del sistema ABCD versión 3.0.
4. Diseño de la Base de Datos (BD) para el desarrollo del módulo Adquisición del sistema ABCD versión 3.0.
5. Caracterización de la arquitectura definida por el equipo de desarrollo del sistema ABCD versión 3.0 para el desarrollo del módulo Adquisición.
6. Elaboración de los diagramas de componentes para el desarrollo del módulo Adquisición del sistema ABCD versión 3.0.
7. Implementación de funcionalidades para el desarrollo del módulo Adquisición del sistema ABCD versión 3.0.
8. Diseño del diagrama de despliegue del módulo Adquisición del sistema ABCD versión 3.0 para modelar el hardware utilizado en las implementaciones del sistema y las relaciones entre sus componentes.
9. Diseño de casos de prueba para aplicárselos al módulo Adquisición del sistema ABCD versión 3.0.
10. Aplicación de pruebas al módulo Adquisición del sistema ABCD versión 3.0 para verificar el correcto funcionamiento del sistema.

Los **Métodos Teóricos** aplicados a la presente investigación fueron:

**El Análisis Histórico-Lógico** facilitó el acercamiento a antecedentes bibliográficos que abordan la evolución y desarrollo de las bibliotecas y los SIGB. Como resultado de su uso se delimitaron, corrigieron y adecuaron los datos adquiridos a las particularidades de la investigación.

## *Introducción*

**El Inductivo-Deductivo** permitió llegar a conclusiones a partir de los conocimientos adquiridos durante la búsqueda bibliográfica en la investigación. Posibilitó tener mayor dominio sobre los SIGB y así poder desarrollar el módulo cumpliendo el objetivo propuesto.

**La Modelación** permitió diseñar las interfaces correspondientes al módulo Adquisición, así como la realización de los diagramas de clases del diseño, interacción, diseño de la BD, componentes y despliegue. Además, facilitó la comprensión de la arquitectura de la propuesta de solución y de las relaciones entre sus componentes.

Complementaron metodológicamente esta investigación los **Métodos Empíricos**:

**La Entrevista** se utilizó para documentar el flujo actual de los procesos del negocio en las Bibliotecas Nacionales y los procesos del módulo Adquisición del sistema ABCD versión 1.2, así como sus diferencias y similitudes con la nueva versión 3.0 que se desea implementar. **(Ver Anexo 1)**

Se desarrollaron, además, un conjunto de técnicas y procedimientos como la **Revisión Documental** para delinear las principales fuentes y/o referencias bibliográficas afines con la investigación.

El contenido del presente trabajo quedó estructurado en tres capítulos organizados de la siguiente forma:

**Capítulo.1“Fundamentación teórica”:** Presenta un estudio del módulo Adquisición de los SIGB. Se muestran las vías de adquisición y se describe la metodología de desarrollo de software, las principales herramientas y tecnologías a utilizar para el desarrollo del módulo.

**Capítulo.2 “Características y diseño de la solución”:** Se describe la propuesta de diseño del sistema y se muestran los diagramas que reflejan dichas descripciones.

**Capítulo.3“Implementación y Prueba de la solución”:** Se detalla el diagrama de componentes, el modelo de despliegue del sistema y los componentes que lo conforman. Además, se describen las pruebas realizadas y se exponen ejemplos de pruebas aplicadas al sistema.

# Capítulo 1: Fundamentación Teórica

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### Introducción

En el presente capítulo se muestran funcionalidades y características de los SIGB utilizados a nivel nacional e internacional y se realiza un estudio de las características fundamentales en los procesos de adquisición de estos. Además se realiza un resumen de la metodología, lenguajes, herramientas y tecnologías escogidas por el grupo de arquitectura de software de ABCD versión 3.0 para llevar a cabo la implementación del módulo.

### 1.1 Proceso de adquisición en las bibliotecas

La adquisición es la forma en la que la biblioteca incorpora a su colección obras previamente seleccionadas. Supone el abastecimiento de la colección bibliográfica, una de las actividades más importantes de la biblioteca, puesto que está dirigida a cubrir las necesidades de información de los usuarios en tiempo, forma y calidad. La adquisición es considerada un proceso bibliotecario fundamental y, por lo tanto, gestionada y/o dirigida por el bibliotecario (8).

### Vías de Adquisición

Las vías de adquisición de las fuentes de información se clasificaron en (9):

**Comerciales:** media el dinero como base del intercambio.

**No comerciales:** se sustenta en los productos que intercambian las organizaciones.

### Vía de Adquisición comercial

- **Compra:** La compra para las bibliotecas puede realizarse a particulares u organismos e instituciones estatales. La persona encargada de dicha actividad, conjuntamente con especialistas de distintas salas, visitará los diferentes suministradores y seleccionará los documentos que sean de interés para la institución. Debe regirse por la política de desarrollo de colecciones y tener en cuenta además, el catálogo de Desideratas no satisfechas (9).

### Vías de Adquisición no comerciales

- **Donación:** Es la vía de adquisición por la cual se obtienen de forma gratuita los documentos, pero tiene el inconveniente que a veces se reciben materiales que no son de interés para la institución,

## Capítulo 1: Fundamentación Teórica

o muchos ejemplares, como en el caso de los textos destinados a la municipalización de la enseñanza; no obstante, es una vía útil y en ocasiones muy ventajosa (9).

- **Canje:** Constituye el cambio de documentos entre instituciones, tanto nacionales como internacionales; los documentos destinados al canje son fundamentalmente las publicaciones de la provincia, duplicados y documentos que se compran con este fin (9).
- **Depósito Legal:** El depósito legal es una de las principales vías para el completamiento de las colecciones de las bibliotecas del sistema, respaldado por el Decreto-Ley No. 265 en Cuba, aprobado el 14 de mayo del 1999. En dicho decreto se establece que "...toda persona natural o jurídica que edite o esté responsabilizada con la edición de una obra publicada en el territorio nacional, independientemente de quien conserve los derechos de edición y el idioma en que se publiquen, estará obligada a enviar con carácter gratuito y sin costo de remisión (...) dos ejemplares a la correspondiente biblioteca pública provincial donde radique la editorial, departamento de edición, imprenta o entidad donde se produce la obra, teniendo dichas bibliotecas la responsabilidad de la conservación, compilación y publicación de la bibliografía. Este envío debe ejecutarse en un término no mayor de 30 días siguientes a la fecha de su terminación. Quedan comprendidas (...) las nuevas ediciones de obras anteriormente publicadas, las ediciones de obras derivadas, facsimilares y reimpressiones, y aquellas que contengan variaciones de cualquier género, aunque sea solo en el formato, calidad de papel o soporte material (9).

### Sistemas Integrados de Gestión Bibliotecaria

Un Sistema Integrado de Gestión de Bibliotecas (SIGB) es una herramienta tecnológica que permite automatizar las operaciones bibliotecarias más comunes. Típicamente abarca la catalogación, circulación, consulta y adquisición de materiales (10).

#### 1.2 Sistemas existentes

Con el objetivo de lograr un mejor entendimiento del proceso de adquisición en los SIGB, se llevó a cabo un estudio del módulo de adquisición de varios sistemas. A continuación se muestra la relación de las funcionalidades del módulo adquisición de cada uno de ellos:

##### Koha

Es un sistema que posee módulos para procesar la Adquisición, Circulación, Registro de Control de Usuarios, Reclamos y Sanciones, además, un Catálogo en línea u OPAC (en inglés, *Online Public Access Catalog*). Dicho sistema, presenta una interfaz simple y clara para bibliotecarios y usuarios (11),

## *Capítulo 1: Fundamentación Teórica*

cuenta con búsqueda configurable y funciona tanto en servidores GNU/Linux como MS-Windows, incluso en MacOS, o sea, es un sistema multiplataforma. Posee un módulo completo de adquisiciones que permite ser administrado de modo simple, incluye presupuestos e información de tasación, capacidad de hacer frente a cualquier número de sedes, de artículos y otros datos.

Este sistema posee la característica singular que permite al bibliotecario administrar la biblioteca remotamente, utilizando un teléfono móvil o un asistente personal. Koha es basado en Web, por lo que puede utilizarse en terminales simples (por ejemplo, terminales sin disco duro o hardware especializado) para las consultas y el manejo de la biblioteca, en su diseño se contempla el modelo de bases de datos relacionales. Es compatible con el formato MARC21 y UNIMARC, soporta el protocolo Z39.50 y como motor de búsqueda utiliza Zebra. Su última versión estable es (v3.00.04) (12).

El módulo encargado de la gestión de adquisiciones en Koha presenta las siguientes características: (12)

- En el área de adquisición se almacenan los datos de proveedores, datos del producto solicitado, el pedido y también permite recibir Sugerencias de los usuarios, efectuados a través del OPAC.
- Permite controlar la procedencia de publicaciones seriadas, es decir, suscripción o donación.
- Muestra el estado del proceso del pedido.
- El personal tiene una única cuenta de acceso y puede modificar, detener o cancelar pedidos.
- Permite fijar la forma de pago y el tipo de pago. Así como la emisión de facturas sobre los pagos con datos del recibo como fecha y número de pedido.
- Permite habilitar la opción de emisión de informes de pago, cantidad total del gasto y acerca del producto número de pedido y situación.

### **PHPMyBibli**

Es un sistema integrado de automatización de bibliotecas de código abierto. Bajo la licencia GNU GPL (Licencia General Pública) diseñada para trabajar en los sistemas operativos Windows y Linux. Ofrece varios módulos como: Adquisiciones, Circulación, Catalogación, Administración, Catálogo de Acceso Público en Línea (OPAC) y Diseminación Selectiva de Información (DSI). Además de monografías, permite gestionar publicaciones periódicas y posee control de autoridades. Cumple con estándares como el MARC21 y permite la búsqueda de registros bibliográficos mediante el protocolo Z39.50. Además, la importación de registros es en formato MARC (12).

## *Capítulo 1: Fundamentación Teórica*

PHPMyBibli es un software configurable, que puede adecuarse a diversas necesidades, tanto por su naturaleza de software libre, como por la posibilidad de establecer parámetros específicos para su uso. Es un sistema multiplataforma (11). El módulo Adquisición no se visualiza por defecto cuando se instala el programa, debe activarse desde el módulo de Administración en el cual además se deben hacer una serie de configuraciones y definiciones. Por ejemplo, los datos de identificación de la biblioteca, los ejercicios contables, los impuestos, las categorías de productos, los gastos, modos de pago y presupuestos, que serán luego mostradas en el menú del módulo Adquisición.

El módulo encargado de la adquisición presenta las siguientes características (12):

- Control de proveedores.
- Generación de reportes del estado del pedido<sup>8</sup>.
- Recepción de Sugerencias de títulos para comprar.
- Elaboración de listado por cada título ejemplar, asignando un porcentaje del total de presupuesto de compra.
- Control de las entregas.
- Control de las facturas.
- Control del presupuesto.

### **ABCD versión 1.2**

El ABCD v1.2 es un sistema integrado en línea que permite la gestión automatizada de bibliotecas y centros de documentación. Esta herramienta integrada de gestión de bibliotecas cubre todas sus funciones principales, es decir, adquisiciones, gestión de bases de datos bibliográficas, catalogación, gestión de préstamos, control de publicaciones periódicas, y servicios de información. Incluye un portal de contenidos que pone a disposición de la comunidad universitaria e interesados distintos recursos de información que la biblioteca posee, ya sea en formato físico o digital y que permitirá visualizar de manera integral las distintas herramientas existentes (13).

Es una aplicación en plataforma Web, de código abierto, multilingüe, que soporta búsquedas en servidores Z39.50. Su tecnología, completamente en ambiente Web, se basa en la familia extendida del

---

<sup>8</sup> Se denomina pedido a la agrupación de una o varias sugerencias del usuario, relacionadas a un mismo elemento (libro/artículo, revista etc.). El pedido contiene los siguientes datos: título, autor, editor cantidad de copias que se desea adquirir y listado de sugerencias asociadas. El bibliotecario puede crear pedidos sin que partan de sugerencias existentes. Para crear un pedido el usuario debe introducir los datos del pedido (título, autor y cantidad de copias) y seleccionar las sugerencias relacionadas en ese pedido. Luego acceder al botón Adicionar y se creará un Pedido. (51)

## Capítulo 1: Fundamentación Teórica

software ISIS, aplicaciones mundialmente difundidas para la administración de bases de datos documentales (13).

Principales características del módulo de Adquisición:

- Registro y estado de títulos pendientes de parte de editorial.
- Permite la creación de Registro de Adquisición<sup>9</sup>.
- Creación de kardex<sup>10</sup> para publicaciones seriadas.
- Gestión de Sugerencias<sup>11</sup>.
- Gestión de Órdenes de compra<sup>12</sup>.

**Principales características en las que coinciden los anteriores SIGB**

- Basados en Web.
- Plataforma multilingüe.
- Software Libre.

Con el objetivo de incluir funcionalidades y perfeccionar las existentes, se realiza la comparación del módulo Adquisición de los SIGB Koha, PhpMyBibli y ABCD versión 1.2. El equipo de desarrollo decidió asignar varios criterios para evaluar dichas características, estos, se detallan a continuación:

Gutiérrez Coral, en su tesis para optar el Título Profesional de Licenciado en Bibliotecología y Ciencias de la Información, titulado “Estudio comparativo de los Sistemas Integrados de código abierto para biblioteca: Koha y Phpmybibli” (12), hace alusión a varios de los presentes indicadores como mínimos requisitos con que debe contar el módulo de adquisición de los SIGB para garantizar un mejor proceso de gestión de la adquisición de materiales bibliográficos.

Inexistente (✗): criterio que no existe.

Existente (✓): criterio que existe.

---

<sup>9</sup> Su objetivo es crear un registro de adquisición en el cual aparezcan los datos generales del registro y los datos específicos de la adquisición de cada uno de los ejemplares. En la primera interfaz de usuario se valida la existencia del registro mediante una búsqueda en la Base de Datos (BD) de adquisiciones por los campos: título y nombre del autor.

<sup>10</sup> Se utiliza para el control de inventarios.

<sup>11</sup> El usuario debe ir hasta la biblioteca para realizar las sugerencias deseadas ya que no posee la posibilidad de acceder al sistema a través de la red.

<sup>12</sup> La Orden de Compra se crea seleccionando al proveedor, la fecha de la compra, los pedidos y especificando el precio de cada ejemplar del pedido.

## Capítulo 1: Fundamentación Teórica

### Comparación del módulo Adquisición de los SIGB (12)

Indicadores	Koha	PhpMyBibli	ABCD v1.2
Registro de datos de proveedores.	✓	✓	✓
Visualizar el estado del proceso de compra.	✓	✓	✓
Recepción de Sugerencias de los usuarios de OPAC.	✓	✓	✗
Control de presupuesto.	✓	✓	✓
Generación de cartas de solicitud y reclamos.	✓	✗	✗
Menús de ayuda en todas las ventanas de trabajo.	✓	✓	✓
Creación de Kardex para publicaciones seriadas.	✓	✗	✓
Registro y estado de títulos pendientes de entrega por parte de la editorial.	✓	✗	✓
Base de Datos	Relacional	Relacional	No Relacional / Relacional

Tabla 1 Comparación de SIGB.

El estudio del módulo Adquisición de los SIGB (Koha, PhpMyBibli y ABCD versión 1.2) evidenció que ABCD versión 1.2 cumple con varios de los indicadores asignados para la evaluación de los sistemas. Además, se determinó agregar la funcionalidad “Recepción de Sugerencias de los usuarios de OPAC”. Luego de un análisis realizado por expertos en Ciencias de la Información, quienes visitaron el CIGED para contribuir con los cambios realizados a ABCD versión 3.0, se llegó a la conclusión de que no solo se agregará la funcionalidad identificada en el estudio de los SIGB anteriormente realizado. Los expertos aportaron como parte del perfeccionamiento del proceso de gestión de la adquisición de materiales bibliográficos, la funcionalidad “Crear Solicitud de Compra” y cambiar el término Pedido por Desiderata, agregándole varios datos necesarios para su creación, pues según los expertos dichas funcionalidades constituyen también parte del flujo básico realizado en las bibliotecas para la adquisición de materiales bibliográficos. En ABCD versión 3.0 también se solucionaron las deficiencias detectadas en la evaluación realizada al módulo Adquisición del sistema ABCD versión 1.2, perfeccionando de esta manera el desempeño de los procesos de gestión de materiales bibliográficos que se lleva a cabo en estas instituciones (Ver Capítulo 2: Características y Diseño de la Solución). Para la implementación de ABCD versión 3.0, el proceso de cambios se realizó sin deterioro de calidad de servicios ofrecidos en versiones anteriores y las deficiencias identificadas en ABCD versión 1.2 fueron corregidas.



## Capítulo 1: Fundamentación Teórica

A continuación se describe el ambiente de desarrollo de ABCD versión 3.0, cuya selección de metodología, tecnologías y herramientas quedan fuera del alcance de la investigación ya que fueron seleccionadas por el equipo de arquitectura del proyecto al que tributa la presente investigación.

### 1.3 Metodología de desarrollo de software y lenguajes

**Metodología de desarrollo de software** se refiere al entorno utilizado para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información (14).

Un **Lenguaje de Programación** es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador expresar el procesamiento de datos y sus estructuras en la computadora. Cada lenguaje posee sus propias sintaxis (15).

En el presente epígrafe se caracteriza la metodología de desarrollo de software y lenguajes de desarrollo utilizados:

#### 1.3.1 Metodología

Proceso Racional Unificado (en inglés, *Rational Unified Process* (RUP)), es una metodología de desarrollo de software y junto al Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso) (16).

El alcance de la presente investigación inicia en la fase de Elaboración, a partir de la disciplina Análisis y Diseño (Figura 1). Por dicho motivo no se hace alusión en este documento a los artefactos que se generan en la fase inicial, pues los mismos forman parte del expediente de desarrollo v4.0 del proyecto ABCD v3.0. La Figura 1 muestra como varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

## Capítulo 1: Fundamentación Teórica

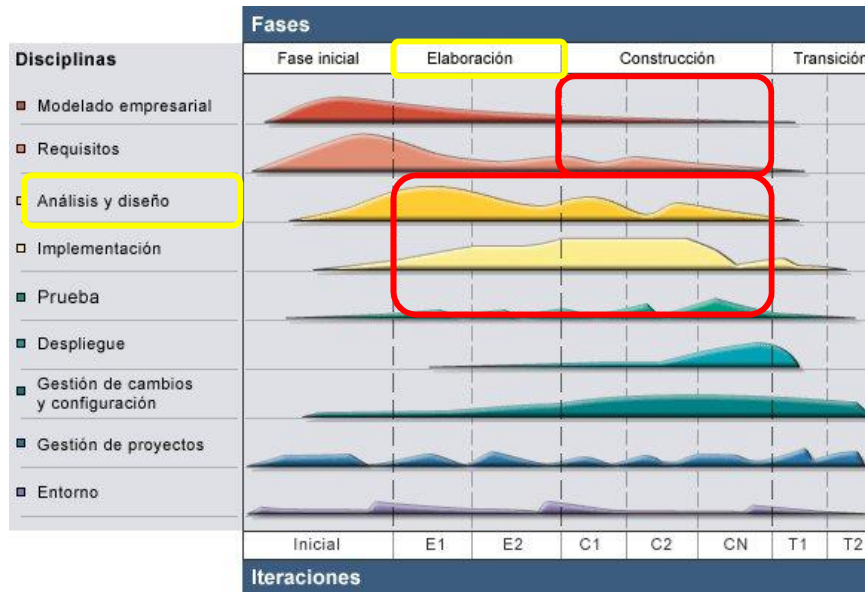


Figura 1 Fases de RUP.

### 1.3.2 Lenguaje de modelado UML versión 2.0

El Lenguaje de Modelado Unificado (UML) permitió especificar, visualizar, construir y documentar artefactos del sistema de software, así como del modelado del negocio. UML se ha convertido en la notación visual estándar de *facto*<sup>13</sup> y *de iure*<sup>14</sup> para el modelado orientado a objetos (17).

Algunas propiedades de UML como lenguaje de modelado estándar son:

- Modela estructuras complejas.
- Las estructuras más importantes que soporta tienen su funcionamiento en las tecnologías orientadas a objetos, tales como: objetos, clases, componentes y nodos.
- Permite modelar el comportamiento del sistema mediante: casos de uso, diagramas de secuencia y de componentes, entre otros.

### 1.3.3 Lenguaje de desarrollo Java

Java es un lenguaje compilado ya que todo programa realizado en Java ha de compilarse, e interpretado debido a que el código que se genera (*bytecodes*) es interpretado por la máquina virtual de Java donde existen varios niveles de seguridad, desde el ámbito del programador (permite realizar comprobación

<sup>13</sup> De Hecho: Con reconocimiento jurídico, legalmente. Se opone a *de facto*.

<sup>14</sup> De Derecho

## Capítulo 1: Fundamentación Teórica

estricta de tipos durante la compilación, evitando con ello problemas tales como el desbordamiento de la pila), hasta el ámbito de la ejecución en la máquina virtual (gestiona la memoria dinámicamente). Una fuente común de errores en programación proviene del uso de punteros. En Java se han eliminado los punteros, el acceso a las instancias de clase se realiza a través de referencias. Además, el programador siempre está obligado a tratar las posibles excepciones que se produzcan en tiempo de ejecución. Dicho lenguaje define procedimientos para tratar estos errores, también posee mecanismos para garantizar la seguridad durante la ejecución comprobando, antes de ejecutar código, que no se viola ninguna restricción de seguridad del sistema donde se va a ejecutar. También cuenta con un cargador de clases, de modo que todas las clases cargadas a través de la red tienen su propio espacio de nombres para no interferir con las clases locales. Otra característica de Java es que está preparado para la programación concurrente sin necesidad de utilizar ningún tipo de biblioteca (18).

### 1.4 Herramientas utilizadas

En el presente epígrafe se caracterizan las herramientas que facilitaron el desarrollo del módulo Adquisición del sistema ABCD versión 3.0.

#### 1.4.1 Visual Paradigm versión 8.0

Visual Paradigm es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software (análisis y diseño, construcción, pruebas y despliegue). Permite modelar diagramas de clases, secuencia, componentes, despliegue, modelo de BD (para BD relacionales), generar documentación, etc. Posee un diseño centrado en Casos de Uso, enfocado al negocio y una capacidad de ingeniería directa e inversa al integrarse con el IDE<sup>15</sup> de desarrollo Eclipse (también permite efectuar ingeniería inversa y directa con la BD relacional PostgreSQL, Oracle, entre otras). Esta herramienta funciona sobre múltiples plataformas, es fácil de instalar y actualizar, además, es compatible con otras ediciones. Visual Paradigm brinda soporte al modelado visual con UML y posee disponibilidad de múltiples versiones en dependencia de las necesidades o condiciones del usuario (19).

#### 1.4.2 Eclipse versión Kepler Release 4.3

Eclipse es un entorno de desarrollo integrado que facilitó las tareas de edición, compilación y ejecución de programas durante la fase de desarrollo. Este se considera una plataforma ligera, pues puede ser

---

<sup>15</sup> *Integrated Development Environment (IDE)*. En español, **Entorno de desarrollo integrado**. Es una aplicación de software, que proporciona servicios integrales para facilitarle al programador de computadora el desarrollo de software. Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador. Tales como NetBeans y Eclipse. (57)

## Capítulo 1: Fundamentación Teórica

personalizada mediante la incorporación de *plugins*<sup>16</sup> para proporcionar funcionalidades en dependencia de para lo que se necesite, lo que lo hace versátil y portable. Para desarrollar el módulo Adquisición se utilizan los siguientes *plugins*: Virgo Tooling, Subclipse, Spring IDE, WTP, JBoss Tools. Eclipse también provee ayuda para codificar, ejemplo de esto es: posee un compilador en tiempo real, permite completamiento de código, generación de plantillas, formateo de código, manejo de notas en código, visualización de la documentación de código, etc (20).

### 1.4.3 Gestores de Bases de Datos

Un sistema de gestión de bases de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los mismos (21). La colección de dichos datos se denomina Base de Datos, (o en inglés, *Database* (DB)) (21). El equipo de desarrollo del sistema ABCD versión 3.0 decidió utilizar para la gestión de Bases de Datos (BD) J-ISIS (para los datos bibliográficos) y PostgreSQL (para los restantes datos), a continuación se caracterizan:

#### PostgreSQL versión 9.3

PostgreSQL es un SGBD libre. Corre en la mayoría de los Sistemas Operativos más utilizados incluyendo, GNU/Linux, OpenSuse y Windows. Cumple el principio ACID (Atomicidad, Consistencia, Integridad, Durabilidad) y tiene soporte completo para llaves foráneas, unión (en inglés, *joins*), vistas, subconsultas, disparadores (*triggers*), y procedimientos almacenados (en varios lenguajes), incluye la mayoría de los tipos de datos de los estándares SQL92 y SQL99, así como herencia entre tablas, por lo que se le considera un gestor de BD relacionales. Posee una documentación completa. Mediante un sistema denominado Acceso Concurrente Multiversión (por sus siglas en inglés, *Multiversion Concurrency Control* o MVCC) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*<sup>17</sup> (22).

#### J-ISIS versión Septiembre 2014

ISIS como software, una de las funciones que desempeña es fungir como servidor para proveer acceso a la Base de Datos J-ISIS, archivo en que la información está contenida en registros numerados

---

<sup>16</sup> (Plug-in) Programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones ni afectar la aplicación principal). (50)

<sup>17</sup> Una sentencia COMMIT en SQL finaliza una transacción de base de datos dentro de un sistema gestor de base de datos relacional y pone visibles todos los cambios a otros usuarios. El formato general es emitir una sentencia BEGIN WORK, una o más sentencias SQL, y entonces la sentencia COMMIT. Una sentencia COMMIT publicará cualquiera de los *savepoints* (puntos de recuperación) existentes que puedan estar en uso.

## Capítulo 1: Fundamentación Teórica

secuencialmente (en inglés, MFNs o *Master File Numbers*<sup>18</sup>) con valores sobre todo textuales entre otros (*integer, date, time, etc.*) almacenados en campos con una "etiqueta" (o identificador numérico) y subcampos (con identificador de un carácter). Subcampos, campos y registros son de longitud variable (se adaptan al tamaño de la información contenida en ellos) y de "ocurrencia variable", entre 0 (no presente) y cualquier número mayor de ocurrencias, en J-ISIS, sin límite. Puede manejar registros bibliográficos, junto con datos de usuarios y datos de transacción (por ejemplo, préstamos) en una única BD (23). En el módulo Adquisición del sistema ABCD versión 3.0 se utilizan dos bases de datos que permiten operar las adquisiciones, Adquisición: donde se registran los datos relacionados con las adquisiciones de materiales bibliográficos realizadas y Ejemplar: donde se registran los datos relacionados con cada ejemplar adquirido). Para la integración el módulo Adquisición con las BD J-Isis se utiliza el componente *JIsisDataProvider*.

### 1.5 Tecnologías.

En el presente epígrafe se caracterizan las tecnologías utilizadas en el desarrollo del módulo Adquisición del sistema ABCD versión 3.0.

#### 1.5.1 Eclipse Virgo versión 3.6.3

Virgo, es un servidor de aplicaciones completamente modular, basado en Java. Diseñado para ejecutar aplicaciones Java y Spring. Ofrece una plataforma para desarrollar y desplegar aplicaciones Java. Virgo *Kernel*<sup>19</sup> es compatible con conceptos básicos de Virgo, este está disponible como código abierto. Virgo *Kernel* también puede usarse independiente como una plataforma de aplicaciones OSGi.

Virgo dispone de (24):

- Extensión de consola de Equinox: Permite gestionar el servidor de aplicaciones Virgo y artefactos desplegados.
- Aprovisionamiento: Suministra automáticamente las dependencias de una aplicación incluyendo *bundles*<sup>20</sup>, planes, plan de archivos y configuraciones, desde repositorios locales y remotos.
- Spring: Puede ser fácilmente configurado para utilizar una versión diferente de Spring (en el caso de ABCD versión 3.0, soporta versiones menores de Spring 4.0).

---

<sup>18</sup> MFNs o *Master File Numbers*: en español, número en el archivo maestro.

<sup>19</sup> Un **núcleo** o **Kernel** es un software que constituye una parte fundamental del sistema operativo, se define como la parte que se ejecuta en modo privilegiado (conocido también como modo núcleo). Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema.

<sup>20</sup> Paquetes, módulos.

## Capítulo 1: Fundamentación Teórica

El servidor de aplicaciones Eclipse Virgo es una implementación de Eclipse. Permite la adición de *bundles* (posee mecanismos para gestionar la comunicación entre estos) y la configuración de planes para su ejecución.

### 1.5.2 RAP versión 3.0M5

Plataforma de Aplicaciones Remotas (por sus siglas en inglés, *RAP*), es un framework para aplicaciones modulares y Aplicaciones de Internet Enriquecidas (en inglés, *Rich Internet Application (RIA*<sup>21</sup>.)), desarrollado por Eclipse. RAP se integra con tecnologías de Eclipse (por ejemplo: Equinox y Virgo). Ofrece un amplio conjunto de componentes que posibilitan construir interfaces gráficas y aplicaciones (*widgets*<sup>22</sup>). Estas pueden ser desarrolladas para el escritorio, el navegador de Internet, y clientes móviles. Mediante la integración de RAP con tecnologías Java los *widgets* pueden ser desplegados directamente como paquetes OSGi en Virgo y otros servidores de aplicaciones. Los mismos pueden ser instalados de forma remota, iniciados, detenidos, actualizados y desinstalados sin necesidad de reiniciar (25). RAP implementa un conjunto de herramientas *widget* estándar (en inglés, *The Standard Widget Toolkit* (SWT<sup>23</sup>)) que forma parte de Eclipse, el programador al hacer uso de este, utiliza el lenguaje de programación Java, muy similar a como si programara para una aplicación de escritorio con SWT. De esta forma se genera el .html<sup>24</sup> y el .js (archivo de texto plano que contiene scripts de JavaScript, y que permite, ser modificado con cualquier editor de textos (26)) desde java, el cual es generado con *qooxdoo*<sup>25</sup>.

### 1.5.3 OSGi Equinox versión 3.8.2

*Open Services Gateway Initiative* (por sus siglas en inglés, OSGi), es una recomendación que posee diferentes implementaciones (por ejemplo: Apache Félix, Eclipse Equinox, Knoplerfish). Su objetivo es definir las especificaciones que permitan diseñar plataformas compatibles que puedan proporcionar múltiples servicios. OSGi intenta solventar los problemas del tradicional "*classloader*"<sup>26</sup> de la máquina virtual y de los servidores de aplicaciones Java. Este *framework* proporciona a los desarrolladores un entorno orientado a servicios y basado en componentes (27) (28).

---

<sup>21</sup>Una *Rich Internet Application (RIA)* es una aplicación web diseñada para ofrecer las mismas características y funciones que normalmente se asocian con aplicaciones *desktop*. (48)

<sup>22</sup>Un *Widget* es una pequeña aplicación o programa, usualmente presentada en archivos o ficheros pequeños. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual. (52)

<sup>23</sup> SWT es un conjunto de herramientas *widget* de código abierto para Java, diseñado para proporcionar un acceso eficiente y portátil a las instalaciones de la interfaz de usuario de los sistemas operativos en los que se aplique. (49)

<sup>24</sup> *HyperText Markup Language (HTML)* se utiliza para crear y representar visualmente una página web. (53)

<sup>25</sup> *Qooxdoo* es una biblioteca de JavaScript y de código abierto utilizada para el desarrollo de aplicaciones web interactivas. (54)

<sup>26</sup> El *Java Classloader* (en español, cargador de clases Java) es una parte del *Java Runtime Environment* que carga dinámicamente clases Java en la Máquina Virtual de Java. El cargador de clases es responsable de localizar bibliotecas, leer sus contenidos, y cargar las clases contenidas dentro de las mismas. Esta carga es normalmente hecha "bajo demanda", por lo que no ocurre hasta que la clase sea usada por el programa. Una clase con un nombre dado sólo puede ser cargada una vez por un *classloader* dado. (55)

## Capítulo 1: Fundamentación Teórica

Equinox proporciona un marco de trabajo java de uso general, seguro y administrado que soporta el despliegue dinámico de aplicaciones conocidas como "*Bundles*" o módulos (27).

Algunas de las características que componen este marco de trabajo (27):

- Es un sistema de módulos para la plataforma java.
- Incluye reglas de visibilidad, gestión de dependencias y versionado de los *bundles*.
- La instalación, arranque, parada, actualización y desinstalación de *bundles* se realiza dinámicamente en tiempo de ejecución sin tener que detener por completo la plataforma.
- Se trata de una arquitectura orientada a servicios dentro de la Máquina Virtual.

Podríamos destacar algunos de los principales beneficios que proporciona esta tecnología:

- Reutilización del código "*out of the box*"<sup>27</sup>.
- Simplificar los proyectos en los que participan muchos desarrolladores en diferentes equipos.
- Se puede utilizar en sistemas más pequeños.
- Gestiona los despliegues locales o remotos.
- No es una tecnología cerrada.

Equinox es una implementación de OSGi y es utilizado por la plataforma de Eclipse. El modelo de programación de OSGi (Equinox) permite definir componentes de software dinámicos, es decir, los servicios OSGi, que también pueden ser parte de una aplicación basada en Eclipse (29). OSGi (Equinox) determina una arquitectura común para proveedores de servicios y desarrolladores, también permite desarrollar, desplegar y trabajar con servicios de manera coordinada.

### 1.5.4 Spring Dynamic Modules versión 3.1.0

*Spring Dynamic Modules* (por sus siglas en inglés, *Spring DM*) es un proyecto de Spring que permite utilizar el Framework del mismo nombre para el desarrollo de aplicaciones que pueden ser ejecutadas en la plataforma OSGi, además, simplifica el desarrollo de paquetes OSGi utilizando conceptos y características de este, tales como la inyección de dependencias (la cual permite construir el software

---

<sup>27</sup> La expresión en inglés "*out of the box*", es una frase que los estadounidenses utilizan con frecuencia. Significa que el código está listo para usarse. Que no necesita cambios. Que tal y como lo "sacamos de la caja" ya es utilizable.

## Capítulo 1: Fundamentación Teórica

con poco acoplamiento) y de archivos XML<sup>28</sup> de configuración (beans-context y osgi-module-context). Entre otras funcionalidades de Spring DM se encuentran (30):

- Busca automáticamente paquetes basados en Spring y crea instancias del contexto de la aplicación Spring para cada *bundle* buscado.
- Busca e importa dinámicamente una referencia de servicios OSGi como *beans*<sup>29</sup> a través de configuraciones XML.

Ventajas de la Inyección de dependencias:

- Proporciona bajo acoplamiento entre componentes (provee componentes más desacoplados e independientes).
- La responsabilidad de cada uno de los componentes será muy clara y un cambio podrá ser más fácil de implementar.

### 1.5.5 EclipseLink versión 2.3

*EclipseLink* es un proyecto de código abierto que permite a los desarrolladores de Java interactuar con base de datos, *Web Services*, objetos XML, etc. La intención del proyecto es ofrecer unos estándares de solución centrada en JPA pero con la capacidad de utilizar funciones avanzadas para aquellas aplicaciones donde se requieren. Al centrarse primero en JPA, el proyecto *EclipseLink* permite la integración amplia y minimiza el acoplamiento, así como proporciona un marco de persistencia que se ejecuta en cualquier ambiente Java. *EclipseLink JPA* ofrece soporte avanzado para contenedores de Java y bases de datos relacionales (31).

### Conclusiones del capítulo

En el presente capítulo se describieron conceptos relacionados con los procesos a informatizar, lo que facilitó un mejor entendimiento del negocio. Se analizaron módulos de Adquisición de SIGB lo que permitió llegar a la conclusión de que ABCD versión 3.0 cumplirá con varios de los indicadores mencionados en la Tabla 1, adicionando las funcionalidades para la gestión bibliotecaria sugeridas por los expertos en información. Además, se realizó un estudio de la metodología, tecnologías, lenguajes y herramientas seleccionadas para el desarrollo del módulo Adquisición lo que permitió tener un mayor

---

<sup>28</sup> XML, siglas en inglés de *eXtensible Markup Language* ('lenguaje de marcas extensible'), es un lenguaje de marcas utilizado para almacenar datos en forma legible. Permite definir la gramática de lenguajes específicos. XML da soporte a Bases de Datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información. (56)

<sup>29</sup> *Beans*: Hace referencia a los servicios para recibir notificaciones cuando estos están configurados dependiente o independiente por Spring DM.



## *Capítulo 1: Fundamentación Teórica*

conocimiento de las herramientas utilizadas para el desarrollo del sistema, por ejemplo: OSGi (Equinox), Spring DM y del framework de presentación RAP, así como la BD relacional PostgreSQL y la no relacional J-ISIS.

Este estudio arrojó que:

- Una elección factible para desarrollar el módulo Adquisición, es la metodología RUP. Esta metodología constituye una guía de cómo desarrollar una aplicación de gran escala.
- Las herramientas seleccionadas facilitan el desarrollo en cuanto al control de las versiones, el modelado y la implementación del módulo.

En el presente capítulo, se obtuvieron los elementos conceptuales necesarios para iniciar la construcción del módulo Adquisición del sistema ABCD versión 3.0.

## Capítulo 2: Características y diseño de la solución

### CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DE LA SOLUCIÓN

#### Introducción

En el presente capítulo se realiza la caracterización de la solución propuesta a través de los Casos de Uso del Sistema (CUS) identificados, se describen los patrones de diseño e implementación utilizados y también la arquitectura definida para el sistema ABCD versión 3.0. Con la descripción de los casos de uso y la arquitectura se procede a diseñar el módulo mediante la elaboración de sus diagramas de clases y sus diagramas de secuencia. Contiene además, la descripción del modelo entidad-relación de la BD.

#### 2.1 Propuesta de Solución

El flujo del módulo Adquisición comienza cuando se desea registrar un nuevo material bibliográfico, por cualquiera de las vías de adquisición mencionadas anteriormente. La aplicación cuenta con cuatro procesos fundamentales, los cuales llevan a un macro-proceso que posibilita registrar el material bibliográfico. A continuación se muestra el diagrama de procesos para la adquisición de un Material Bibliográfico.

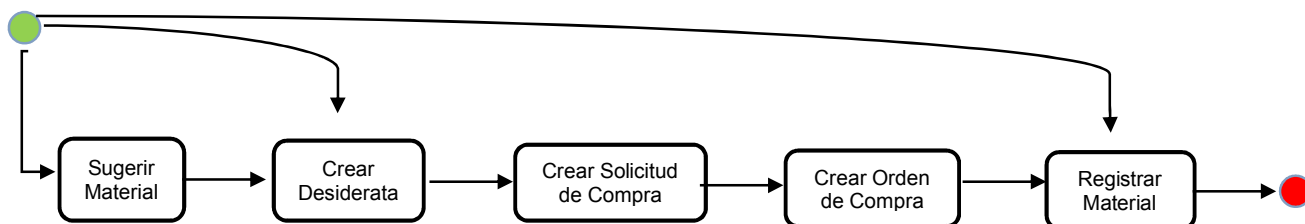


Figura 2 Macro-Proceso para Adquirir un Material.

En la siguiente tabla se detallan los pasos a realizar en cada proceso para registrar un Material Bibliográfico. En caso de ser la vía de adquisición Compra, se realizan los procesos mostrados anteriormente; por otra vía de adquisición solo se Registra la Adquisición y la Adquisición del Ejemplar.

#### Descripción de Procesos

Procesos	Pasos
Sugerir Material	<ul style="list-style-type: none"><li>- Se realiza la Sugerencia en el módulo Catálogo en línea u OPAC (inglés <i>Online Public Access Catalog</i>).</li><li>- La Sugerencia se muestra en estado Pendiente en el módulo Adquisición y permite ser aprobada o rechazada.</li></ul>
Crear Desiderata	<ul style="list-style-type: none"><li>- Aprobada la Sugerencia, se crea la Desiderata.</li></ul>

## Capítulo 2: Características y diseño de la solución

	<ul style="list-style-type: none"> <li>- Una Desiderata puede tener Sugerencias asociadas si es realizada por el usuario o sin Sugerencias asociadas si es realizada por el bibliotecario.</li> </ul>
Crear Solicitud de Compra	<ul style="list-style-type: none"> <li>- Creada la Desiderata, se realiza la Solicitud de Orden de Compra.</li> <li>- Una Solicitud de Orden de Compra puede ser aprobada o rechazada.</li> <li>- Aprobada la Solicitud de Orden de Compra, se crea la Orden de Compra.</li> </ul>
Crear Orden de Compra	<ul style="list-style-type: none"> <li>- Creada la Orden de Compra, se escoge el proveedor y permite ajustar los precios y cantidades de materiales.</li> </ul>
Registrar Material	<ul style="list-style-type: none"> <li>- Realizada la compra de los materiales se registran. Completando de esta manera el proceso Registrar Adquisición y Registrar Adquisición del Ejemplar.</li> <li>- Registrar Adquisición<sup>30</sup> y Registrar Adquisición del Ejemplar<sup>31</sup> son procesos obligatorios que constituyen el núcleo del módulo.</li> </ul>

Tabla 2 Descripción de Procesos para Adquirir Material Bibliográfico.

### 2.2 Descripción de Actores

Se llama actor a toda entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad. Esto incluye los operadores humanos pero también todos los sistemas externos. En el caso de los seres humanos se puede ver los actores como definiciones de rol por lo que un mismo individuo puede corresponder a uno o más actores (32). En la siguiente tabla se muestra la descripción de los actores que participan en los disímiles Casos de Uso del módulo Adquisición.

Actor	Objetivos
Aprobador de Sugerencias	Lista las Sugerencias pendientes y decide si aprobarlas o rechazarlas.

<sup>30</sup> El Registro de una Adquisición (CU Registrar Adquisición) puede incluir la Adquisición de varios Ejemplares. Para registrar una adquisición se debe introducir de forma obligatoria los siguientes datos: Título, Autor; y se debe seleccionar: Tipo de material y Sala de la biblioteca.

<sup>31</sup> Un Ejemplar (CU Registrar Registro de Adquisición de Ejemplar) es un material bibliográfico específico, del cual se registrarán datos como: Autor Corporativo, Volumen, Tomo, Ciudad, País, Código de Barra, etc.

## *Capítulo 2: Características y diseño de la solución*

Consultor de Sugerencias	de	Consulta todas las Sugerencias realizadas y permite ver sus datos. Elimina Sugerencias con estado: "Pendiente" o "Ejecutada".
Gestor de Desiderata	de	Registra, edita, visualiza y elimina Desideratas. Lista Sugerencias con estado: "Pendientes".
Consultor de Desideratas	de	Consulta Desideratas realizadas y visualiza datos de una Desiderata. Elimina Desideratas con estado: "Pendiente" o "Ejecutado".
Gestor de Solicitud de Orden de Compra		Registra, edita, visualiza y elimina Solicitudes de Órdenes de Compra. Lista Desideratas con estado: "Pendiente".
Consultor de Solicitud de Orden de Compra	de	Consulta todas las Solicitudes Órdenes de Compra realizadas y permite ver sus datos. Elimina Órdenes de Compra con estado: "Pendiente" o "Ejecutada".
Aprobador de Solicitud de Orden de Compra	de	Lista las Solicitudes de Órdenes de Compra pendientes y decide si aprobarlas o rechazarlas.
Gestor de Órdenes de Compra		Registra el proveedor y ajusta precios y cantidades de materiales bibliográficos.
Gestor de Registro de Adquisición		Registra, edita, visualiza y elimina Registros de Adquisición.
Consultor de Registro de Adquisición	de	Consulta Registros de Adquisición existentes y visualiza sus datos.
Sistema		Notifica las adquisiciones recientes.

*Tabla 3 Descripción de Actores del sistema.*

### **2.3 Resumen de los casos de uso del módulo Adquisición**

En el documento Especificación de Casos de Uso del Módulo Adquisición (33), se describen los Casos de Uso que reflejan el funcionamiento del sistema en cuanto a la gestión de la adquisición en ABCD versión 3.0. El entendimiento de los Casos de Uso es necesario para diseñar el módulo y para su posterior implementación. A continuación se listan los Casos de Usos definidos y su resumen.

<b>Casos de Uso</b>	<b>Resumen</b>
---------------------	----------------

## *Capítulo 2: Características y diseño de la solución*

<b>CUS_AQ Consultar Sugerencias</b>	El sistema brinda la posibilidad de introducir los datos elementales de búsqueda para realizar la consulta de Sugerencias. El actor introduce los datos, el sistema busca los resultados y muestra una lista de posibles coincidencias. El sistema permite imprimir y/o exportar los resultados obtenidos.
<b>CUS_AQ Aprobar Rechazar Sugerencias</b>	El sistema permite aprobar o rechazar cada una de las Sugerencias listadas.
<b>CUS_AQ Consultar Desiderata</b>	El sistema brinda la posibilidad de introducir los datos elementales de búsqueda para realizar la consulta de Desideratas. El actor introduce los datos, el sistema busca los resultados y muestra una lista de posibles coincidencias. El sistema permite imprimir y/o exportar los resultados obtenidos.
<b>CUS_AQ Gestionar Desiderata</b>	En caso de que el actor seleccione la opción de: <ul style="list-style-type: none"> <li>- Registrar una Desiderata, el sistema le permite especificar los datos y registrarlos.</li> <li>- Visualizar una Desiderata, el sistema muestra los datos y permite editarlos e imprimirlos.</li> <li>- Editar una Desiderata, el sistema muestra los datos y permite modificar sus valores y actualizar los datos.</li> <li>- Eliminar una Desiderata, el sistema muestra antes un mensaje de confirmación y elimina la Desiderata.</li> </ul>
<b>CUS_AQ Asociar Desiderata</b>	El sistema brinda la posibilidad de consultar las Desideratas registradas en el sistema y permite asociar una Desiderata al Registro de Adquisición del Ejemplar.
<b>CUS_AQ Consultar Solicitudes de Órdenes de Compra</b>	El sistema brinda la posibilidad de introducir los datos elementales de búsqueda para realizar la consulta de Órdenes de Compra. El actor introduce los datos, el sistema busca los resultados y muestra una lista de posibles coincidencias. El sistema permite imprimir y/o exportar los resultados obtenidos.
<b>CUS_AQ Gestionar</b>	En caso de que el actor seleccione la opción de:

## *Capítulo 2: Características y diseño de la solución*

<b>Solicitud de Orden de Compra</b>	<ul style="list-style-type: none"> <li>- Registrar una Orden de Compra, el sistema le permite especificar los datos y registrarlos.</li> <li>- Visualizar una Orden de Compra, el sistema muestra los datos y permite editarlos e imprimirlos.</li> <li>- Editar una Orden de Compra, el sistema muestra los datos y permite modificar sus valores y actualizar los datos.</li> <li>- Eliminar una Orden de Compra, el sistema muestra antes un mensaje de confirmación y elimina la Orden de Compra.</li> </ul>
<b>CUS_AQ Aprobar Rechazar Solicitud de Orden de Compra</b>	<p>El sistema lista las Órdenes de Compra que posean estado: "Pendiente" y permite aprobarlas o rechazarlas.</p>
<b>CUS_AQ Asociar Solicitud de Orden de Compra</b>	<p>El sistema brinda la posibilidad del consultar las Órdenes de Compra registradas en el sistema con estado: "Aceptada" y permite asociar una Orden de Compra al Registro de Adquisición de Ejemplar.</p>
<b>CUS_AQ Gestionar Orden de Compra</b>	<p>El sistema brinda la posibilidad de registrar el proveedor y ajustar los precios y cantidades de materiales bibliográficos.</p>
<b>CUS_AQ Consultar Registro de Adquisiciones</b>	<p>El sistema brinda la posibilidad de introducir los datos elementales de búsqueda para realizar la consulta de Registro de Adquisición. El actor introduce los datos, el sistema busca los resultados y muestra una lista de posibles coincidencias. El sistema permite imprimir y/o exportar los resultados obtenidos.</p>
<b>CUS_AQ Gestionar</b>	<p>En caso de que el actor seleccione la opción de:</p> <ul style="list-style-type: none"> <li>- Registrar una Adquisición, el sistema permite especificar los datos y registrarlos.</li> </ul>

## Capítulo 2: Características y diseño de la solución

<b>Registro de Adquisición</b>	<ul style="list-style-type: none"> <li>- Visualizar una Adquisición, el sistema muestra los datos y permite editarlos, eliminarlos e imprimirlos.</li> <li>- Editar una Adquisición, el sistema muestra los datos y permite modificar sus valores y actualizar los datos.</li> <li>- Eliminar un Registro de Adquisición, el sistema muestra un mensaje de advertencia y lo elimina.</li> </ul>
<b>CUS_AQ Gestionar Registro de Adquisición de Ejemplares</b>	<p>En caso de que el actor seleccione la opción de:</p> <ul style="list-style-type: none"> <li>- Registrar la Adquisición de un Ejemplar, el sistema le permite especificar los datos.</li> <li>- Visualizar la Adquisición de un Ejemplar, el sistema muestra los datos y permite editarlos e imprimirlos.</li> <li>- Editar la Adquisición de un Ejemplar, el sistema muestra los datos y permite modificar sus valores y actualizar los datos.</li> <li>- Eliminar la Adquisición de un Ejemplar, el sistema muestra antes un mensaje de confirmación y elimina el Registro de Adquisición de Ejemplar.</li> </ul>

Tabla 4 Comportamiento de Casos de Usos

### 2.4 Arquitectura

En la programación orientada a objetos se entiende por **patrón**, una solución probada que puede ser aplicada con éxito a determinados tipos de problemas que aparecen repetidamente en el desarrollo de sistemas de software (34). Por tanto, se pueden definir los **patrones de arquitectura** como esquemas de organización de sistemas de software. Estos especifican una serie de subsistemas y sus responsabilidades, e incluyen reglas y criterios para organizar las relaciones existentes entre ellos. Resuelven problemas arquitectónicos, adaptabilidad a requerimientos cambiantes, rendimiento, modularidad, acoplamiento, etc. (34).

La Arquitectura Base Orientada a Servicios (ABOS), definida por el equipo de desarrollo para el sistema ABCD versión 3.0, posee una estructura de n capas, esta, permitió dividir el trabajo modularmente.

Los principales beneficios del estilo de arquitectura basado en capas son (35):

- **Abstracción.** Las capas permiten cambios que se realizan a nivel abstracto. Se puede incrementar o disminuir el nivel de abstracción usado en cada capa de la “pila” jerárquica.

## Capítulo 2: Características y diseño de la solución

- **Aislamiento.** El estilo de arquitectura en capas permite asilar los cambios en tecnologías a ciertas capas para reducir el impacto en el sistema.
- **Rendimiento.** Distribuir las capas entre múltiples sistemas (físicos) puede incrementar la escalabilidad, la tolerancia a fallos y el rendimiento.
- **Mejoras en Pruebas.** La capacidad de realizar pruebas se beneficia de tener interfaces bien definidas para cada capa, así como de la habilidad para cambiar a diferentes implementaciones de las interfaces de las capa.

A continuación se muestra la estructuración de la arquitectura definida para el sistema ABCD versión 3.0:

- **Capa de presentación:** consiste en una interfaz gráfica que reúne los aspectos de software enfocados a la interacción con los diferentes tipos de usuarios. Es decir, incluye el manejo y aspecto de las ventanas, la autenticación, el formato de los reportes, menús, gráficos y demás elementos multimedia. En dicha capa se encuentra el *bundle cu.uci.abcd.acquisition.ui* donde se localizan las interfaces de usuario, en ellas se debe implementar *IContributor* para que todas tengan la misma estructura visual, además para la construcción de interfaces también se utilizan entre otros componentes los brindados por la Arquitectura (ABOS), p.e. *CRUDTreeTable*: utilizado para la construcción de tablas, la internacionalización: para para que la vista permita ser mostrada en varios idiomas, etc. En dicho *bundle* también se encuentran las clases controladoras, encargadas de controlar las funcionalidades del sistema.
- **La capa de negocios:** reúne los aspectos de software que automatizan los procesos de negocio. Conocida también como capa Lógica de la Aplicación. Recibe la entrada de la capa anterior, interactúa con los servicios de datos para ejecutar las operaciones y envía el resultado procesado a la capa de presentación.
- **La capa de datos:** en la cual se define la conexión con el servidor y la BD, es en esta capa donde se almacena, actualiza y consultan los datos del sistema. En dicha capa se encuentra el *bundle cu.uci.abcd.dao.acquisition* donde se localizan las interfaces de acceso a datos.

La arquitectura definida para el sistema ABCD versión 3.0 contiene además, una capa de infraestructura que se encuentra a lo largo de todas las demás. Esta, añade gestión dinámica de permisos proporcionando un modelo de seguridad capaz de autenticar el código a través de la ubicación del *bundle* y de sus paquetes, además provee una infraestructura para desplegar y manejar aplicaciones que deben ejecutarse en un entorno seguro y controlado. La comunicación entre capas se mantiene a través de servicios, donde cada



## Capítulo 2: Características y diseño de la solución

capa proporciona servicios a la capa inmediata superior y se sirve de las prestaciones que le brinda la inmediata inferior. Esto lo posibilita la utilización de OSGI, quien proporciona a los desarrolladores un entorno orientado a servicios y basado en componentes.

### 2.5 Patrones de diseño GRASP

**GRASP** significa *General Responsibility Assignment Software Patterns* (patrones generales de software para asignar responsabilidades). Estos describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (36). A continuación se muestra, los patrones utilizados en el diseño del sistema ABCD versión 3.0.

#### 2.5.1 Experto

Experto es un patrón basado en el principio básico de la asignación de responsabilidades; suele utilizarse en el diseño orientado a objetos. Con la utilización de este patrón en el módulo Adquisición se conservó el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases sencillas y más cohesivas que son fáciles de comprender y mantener (37).

#### 2.5.2 Creador

La responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtendrá un diseño con mayor cohesión y la información se mantendrá encapsulada (disminución del acoplamiento) (37).

#### 2.5.3 Bajo Acoplamiento

Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases (37).

#### 2.5.4 Alta Cohesión

La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase; por lo que puede ser calificada como alta cuando existen clases con responsabilidades estrechamente relacionadas que no realicen mucho trabajo.

## Capítulo 2: Características y diseño de la solución

Este patrón se basa en la asignación de responsabilidades, de modo que la cohesión siga siendo alta (37).

### 2.5.5 Controlador

El patrón controlador sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación para aumentar la reutilización de código y a la vez tener un mayor control (37).

### 2.6 Patrones de diseño GOF

Los patrones de diseño **GOF**<sup>32</sup> se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (38). En la implementación del módulo Adquisición de proyecto ABCD versión 3.0 se utiliza el siguiente patrón perteneciente a la categoría estructural.

- **Proxy:** Mantiene un representante de un objeto.

### 2.7 Diagrama de Clases del Diseño

Un Diagrama de Clases del Diseño (DCD) modela los conceptos del dominio de la aplicación y permite visualizar las relaciones entre las clases del sistema (39). El DCD que se muestra a continuación será explicado basándose en la solución dada al CU Gestionar Registro de Adquisición (Ver Figura 4), donde se evidencia la arquitectura ABOS.

Las restricciones de diseño son limitaciones al consumo de recursos durante el período de diseño (40), como por ejemplo: restricciones respecto a estándares o normas a seguir para el diseño del producto y restricciones respecto a procesos a seguir (políticas y procedimientos definidos para ellos ya sea por el área interna o por la organización), entre otros. ABOS, la arquitectura utilizada para el desarrollo del módulo Adquisición, posee restricciones para su diseño; a continuación se muestran las más significativas:

- Implementar *IContributor*<sup>33</sup> en las interfaces de usuario.

---

<sup>32</sup> Fue por los años 1994, que apareció el libro "*Design Patterns: Elements of Reusable Object Oriented Software*" escrito por los ahora famosos **Gang of Four** (GoF, que en español es la pandilla de los cuatro) formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Ellos recopilaron y documentaron 23 patrones de diseño aplicados usualmente por expertos diseñadores de software orientado a objetos. Desde luego que ellos no son los inventores ni los únicos involucrados, pero fue luego de la publicación de ese libro que empezó a difundirse con más fuerza la idea de patrones de diseño.

<sup>33</sup>La interfaz pública *IContributor* describe un registro contribuidor, es una entidad que suministra información al registro de extensión. Este se puede obtener llamando *IExtensionPoint.getContributor()*, *IExtension.getContributor()*, y *IConfigurationElement.getContributor()*. Alternativamente, un factor contribuidor apropiado para el registro en uso puede ser llamado para obtener directamente un objeto *IContributor*. Esta interfaz puede ser utilizada sin estar OSGi en funcionamiento, además posee la restricción de no estar destinado a ser implementado o extendido por los clientes. (47)

## Capítulo 2: Características y diseño de la solución

- Utilizar Spring DM tanto para la creación de *beans* como para exportar servicios en los *bundles*.
- Crear en cada módulo el *bundle* para las interfaces público y el de la implementación privado, donde cada módulo se nombra de la siguiente manera: `cu.uci.abcd.<nombre del módulo>.<x>`.
- Exportar las clases de las interfaces de usuario como servicios *IContributorFactory*.
- Utilizar Spring Data y Eclipse Link con anotaciones en la creación de la capa de acceso a datos.
- Ubicar las entidades del dominio en el *bundle* del dominio.
- Reutilizar componentes brindados por la arquitectura (ABOS), reportes, *CRUDTreeTable*, *validator* e internacionalización, que además de ser restricciones de dicha arquitectura también conforman los elementos que se utilizan de esta para la construcción de interfaces en el *bundle* `cu.uci.abcd.acquisition.ui`.

Un *bundle* es una aplicación empaquetada en un fichero `.jar`, que se despliega en la plataforma OSGI. Cada *bundle* contiene un fichero de metadatos organizado por pares de nombre “clave: valor” donde se describen las relaciones del *bundle* con lo que lo rodea (por ejemplo, qué paquetes importa y qué paquetes exporta). Este fichero viene dado con el nombre `MANIFEST.MF` y se encuentra ubicado en el directorio `META-INF` (25).

El módulo Adquisición del sistema ABCD versión 3.0 está compuesto por tres *bundles*, estos se especifican a continuación:

***cu.uci.abcd.acquisition***: Es un *bundle* público donde se definen las interfaces para la implementación de los servicios, cada una de ellas implementan *IContributor*.

***cu.uci.abcd.acquisition.impl***: Es un *bundle* privado en el que se implementan las interfaces (estas contienen las funcionalidades definidas en el *bundle* `cu.uci.abcd.acquisition`) y se publican los servicios definidos por las interfaces en el *bundle* público.

***cu.uci.abcd.acquisition.ui***: Consume los servicios de `cu.uci.abcd.acquisition` y se encarga de la gestión de la interfaz de usuario.

## Capítulo 2: Características y diseño de la solución

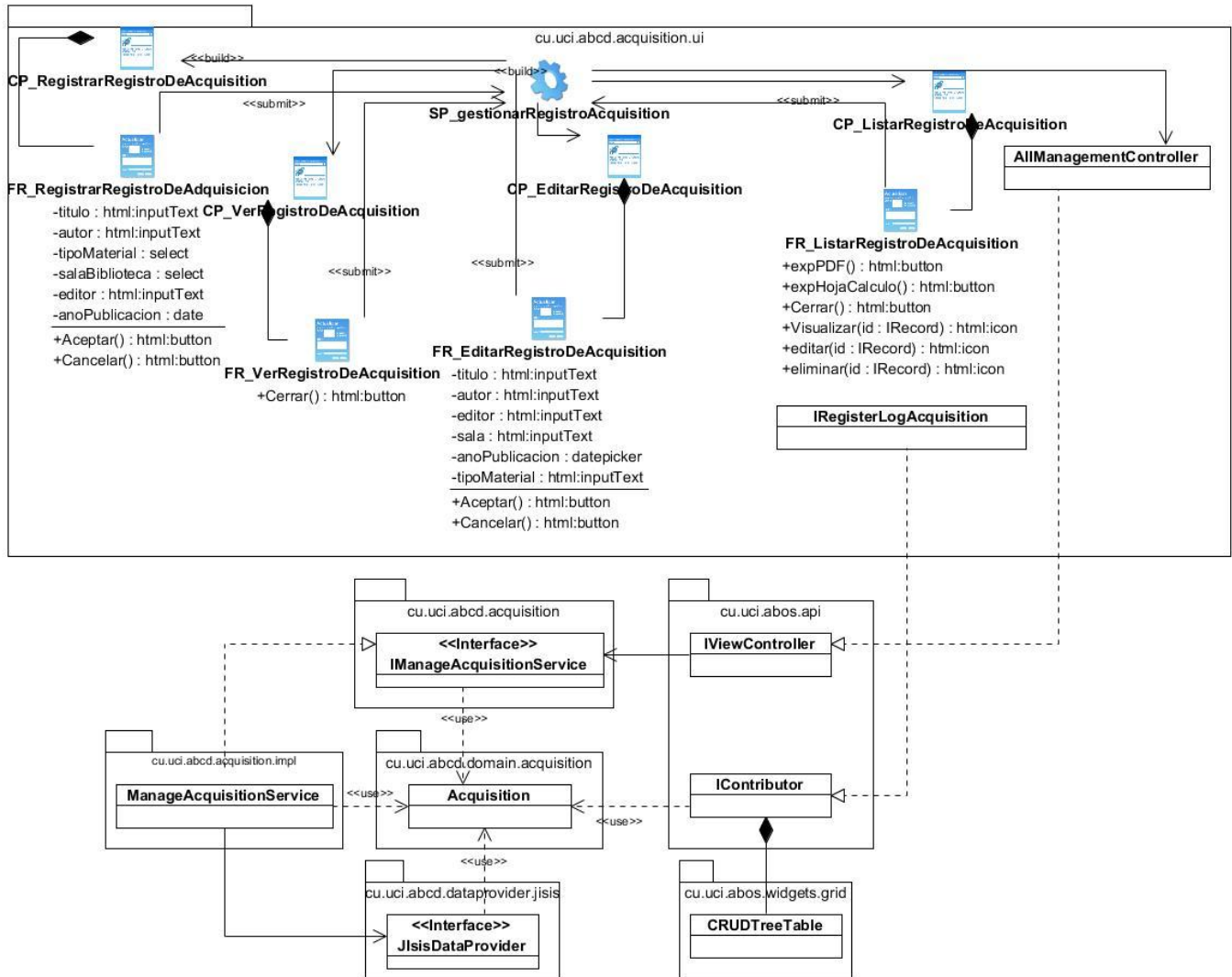


Figura 3 Diagrama de Clases del Diseño Gestionar Registro de Adquisición.

### 2.8 Diagrama de Secuencia

Los diagramas de interacción describen el comportamiento del sistema, para mostrar la interacción dinámica de los objetos en diferentes momentos durante la ejecución del sistema. Se agrupan en dos tipos de diagramas: secuencia y colaboración. En este caso, se escogió el diagrama de secuencia perteneciente al escenario 1 (SC 1) Registrar Registro de Adquisición referente al CU Gestionar Registro de Adquisición. En dicho diagrama no se especifican todas las clases representadas en el DCD para no aumentar la complejidad del presente diagrama y a su vez sea más entendible.

## Capítulo 2: Características y diseño de la solución

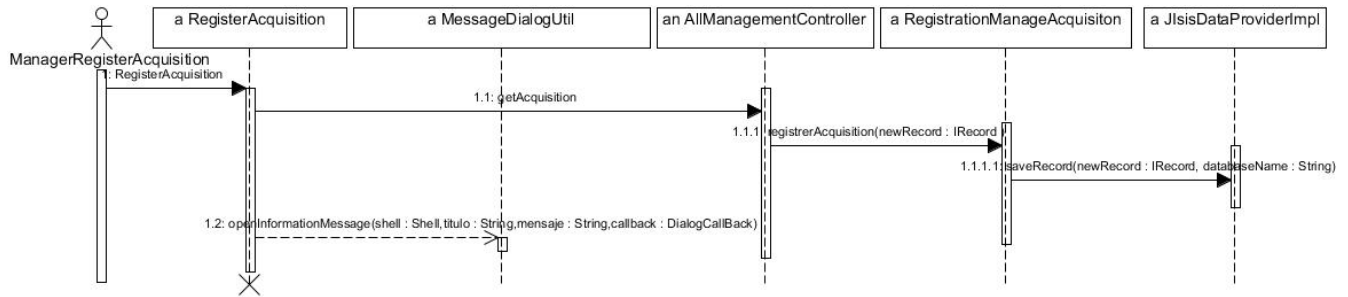


Figura 4 Diagrama de Secuencia Registrar Registro de Adquisición.

### 2.9 Modelo de Base de Datos

A continuación se presenta el modelo de la BD, el cual cuenta con diez entidades. De estas entidades se encuentran almacenadas en PostgreSQL las tablas: *Usuario*, *Desiderata*, *Solicitud de Compra* y *Orden de Compra* (se encuentran señaladas en azul), las restantes tablas (almacenadas en PostgreSQL) son necesarias para la integración del módulo Adquisición con el sistema ABCD versión 3.0. Por otra parte, *Registro de Adquisición* y *Registro de Adquisición del Ejemplar* se encuentran almacenadas en J-Isis. Dichas entidades representan por si solas una BD cada una, una de las principales características de la BD no relacional J-Isis.

# Capítulo 2: Características y diseño de la solución

## 2.9.1 PostgreSQL

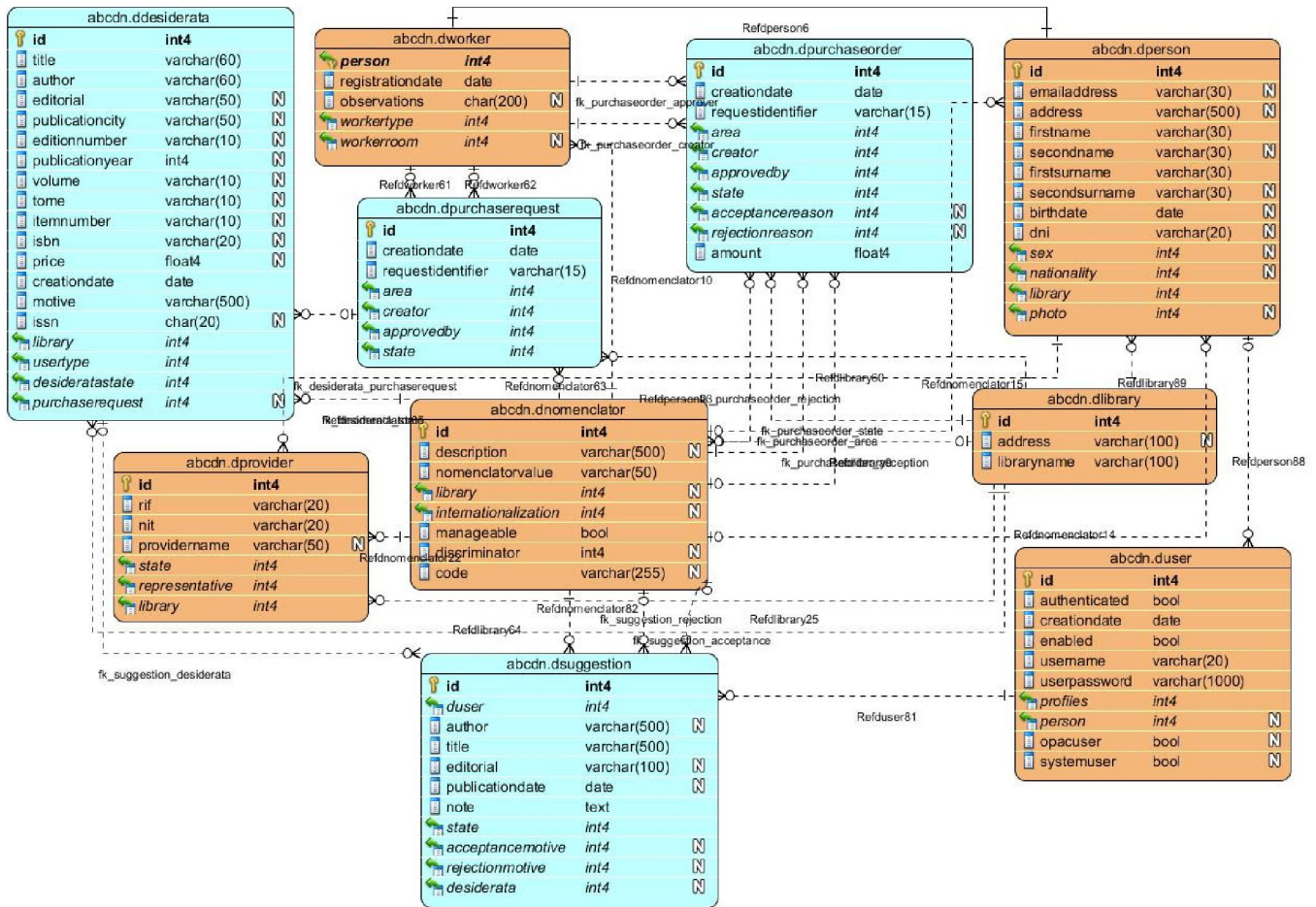


Figura 5 Modelado de la BD en PostgreSQL.

## 2.9.2 Base de Datos Adquisición (J-Isis)

Adquisición	
Atributos	Tipos de Datos
NumControl	integer
Título	varchar
Autor	varchar
TipoDeMaterial	varchar

## Capítulo 2: Características y diseño de la solución

SalaBiblioteca	varchar
Editor	varchar
AñoPublicación	integer

Tabla 5 Modelado de la Base de Datos Adquisición.

### 2.9.3 Base de Datos Ejemplar (J-Isis)

Ejemplar	
Atributos	Tipos de datos
AutorCorporativo	varchar
Volumen	varchar
Tomo	varchar
Ciudad	varchar
País	varchar
Edición	varchar
NotaPública	varchar
CódigoBarra	varchar
LocalizaciónPermanente	varchar
LocalizaciónEstantería	varchar
RestriccionesUso	varchar
Proveedor	varchar
TipoAdquisición	varchar
FechaAdquición	date
PrecioEstimado	double
Moneda	varchar

## Capítulo 2: Características y diseño de la solución

EstadoDelMaterial	varchar
-------------------	---------

Tabla 6 Modelado de la BD Ejemplar.

### Conclusiones del capítulo.

En el presente capítulo se mostró un resumen de los Casos de Uso del módulo Adquisición, lo que permitió obtener un mayor entendimiento del negocio a diseñar. A partir del estudio de los mismos, se definió la concepción del módulo en cuestión perteneciente al sistema ABCD versión 3.0. Además, se describió la arquitectura de software definida lo que facilitó el desarrollo del módulo Adquisición. Teniendo en cuenta esta arquitectura se procedió al diseño del módulo donde se realizaron los diagramas de clases con estereotipos web y de secuencia. Se describió, además, el modelo entidad-relación de la BD para finalizar el diseño.



## Capítulo 3: Implementación y prueba de la solución

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DE LA SOLUCIÓN

#### Introducción.

En el presente capítulo se hace referencia a la implementación del módulo Adquisición, basándose en el resultado del diseño realizado en el capítulo anterior. Como parte de la implementación se definen las clases significativas de cada *bundle* y se muestran los diagramas de componentes y despliegue. Además se incluye el diseño de pruebas y los resultados obtenidos de la aplicación de las pruebas al módulo.

#### 3.1 Implementación.

El flujo de trabajo Implementación, de RUP, explica cómo desarrollar, organizar, realizar pruebas e integrar los componentes implementados basándose en las especificaciones de diseño. En este se implementan las clases y objetos en ficheros fuente, binarios y ejecutables. Todos los elementos implementados forman el modelo de implementación y se integran de forma incremental para en caso de que ocurran fallos sean más fáciles de detectar. El resultado final de este flujo de trabajo en la presente investigación es un sistema web (41).

#### 3.2 Diagramas de componentes.

Un Diagrama de Componentes, representa la separación de un sistema de software en componentes físicos, (por ejemplo, archivos, módulos, paquetes, etc.) y muestra las dependencias entre estos. Los Diagramas de Componentes ilustran las piezas del software, controladores que conformarán un sistema. Además tienen un nivel más alto de abstracción que un diagrama de clase. Usualmente, un componente se implementa por una o más clases en tiempo de ejecución (42).

La Figura 9 muestra el Diagrama de Componentes del CU Registrar Registro de Adquisición. Este evidencia las dependencias que existen entre los *bundles* del módulo Adquisición y el componente *IJisisDataProvider*, *bundle* donde se encuentran declarados los servicios para el trabajo con la BD no relacional, dichos servicios son consumidos por las clases *RegisterLogAcquisition* y *RegisterLogSampleAcquisition*. Además, se visualiza el uso de elementos de la arquitectura ABOS para la construcción de interfaces (cuya representación se realiza con un paquete nombrado "cu.uci.abos"), dichos elementos no se detallan por ser varios, pero entre ellos se encuentran la reutilización de componentes como: reportes, *CRUDTreeTable*,

## Capítulo 3: Implementación y prueba de la solución

*validator*, internacionalización, entre otros. En el Anexo 4 se encuentran los Diagramas de Componentes referentes al módulo de Adquisición.

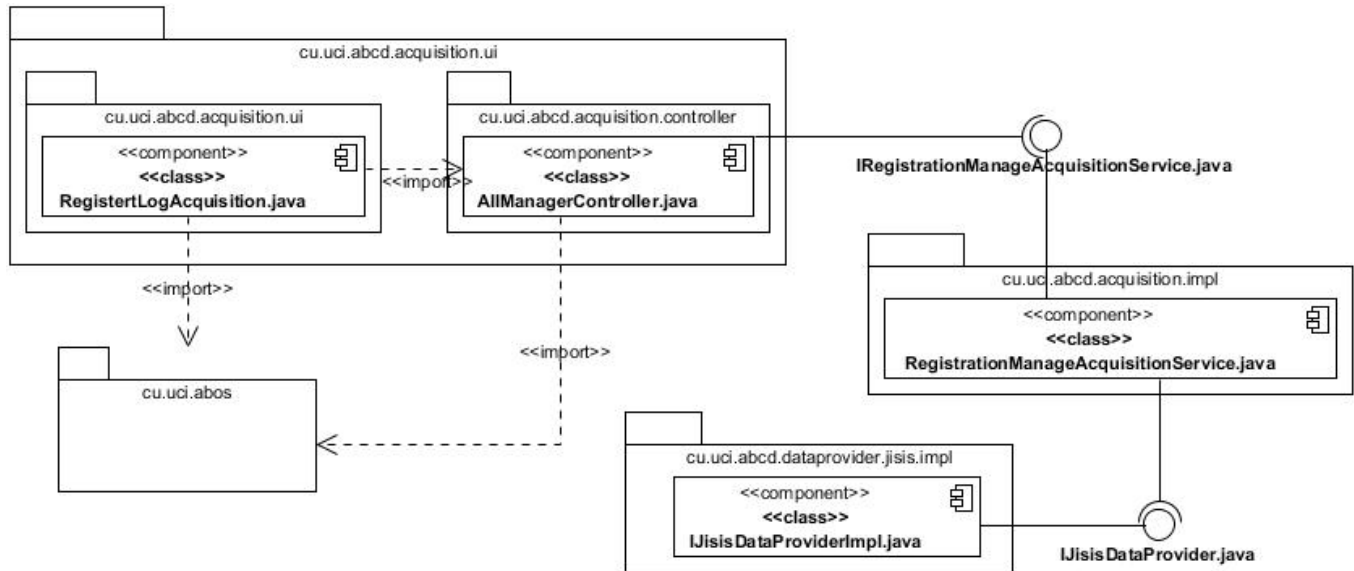


Figura 6 Diagrama de Componentes del CU Gestionar Registro de Adquisición, perteneciente al Módulo Adquisición del sistema ABCD versión 3.0.

### 3.2 Diagrama de despliegue.

El Diagrama de Despliegue, es un tipo de diagrama del Lenguaje Unificado de Modelado (UML), que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Además, describe la arquitectura física del sistema durante la ejecución, en términos de:

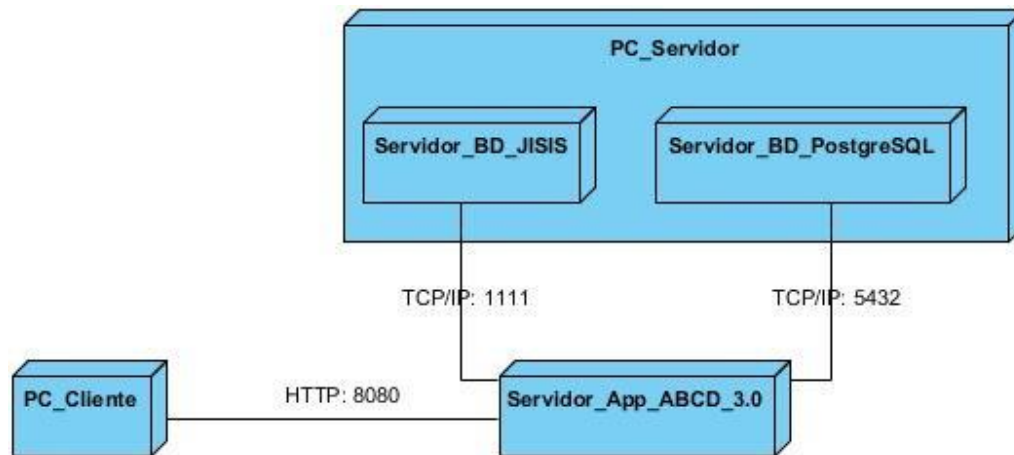
- Procesadores.
- Dispositivos.
- Componentes de software.

También describen la topología del sistema: la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos (1).

La presente vista de despliegue muestra las relaciones físicas de los distintos nodos que componen el sistema y la distribución de los componentes sobre dichos nodos.

A continuación se muestra el diagrama de despliegue:

## Capítulo 3: Implementación y prueba de la solución



Para el despliegue de la aplicación se necesita una PC cliente, que debe tener instalado un navegador web, este hace peticiones al servidor de aplicaciones Eclipse Virgo versión 3.6.3, donde se encuentra la aplicación incluyendo los *bundles* *cu.uci.abcd.acquisition.ui*, *cu.uci.abcd.acquisition* y *cu.uci.abcd.acquisition.impl*, usando el protocolo de comunicación (HTTP) por el puerto 8080. El sistema establece comunicación con los servidores de Bases de Datos PostgreSQL versión 9.3 utilizando el protocolo TCP/IP por el puerto 5432 y J-Isis versión Septiembre 2014 utilizando el protocolo TCP/IP por el puerto 1111.

### 3.3 Pruebas.

Las pruebas de software consisten en la verificación del comportamiento de un programa con un conjunto finito de casos de prueba. Son una serie de actividades dentro del desarrollo de software que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa u ordenador; probando el comportamiento del mismo (43).

#### 3.3.1 Método de Caja Negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, o sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene (43).

Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en

## Capítulo 3: Implementación y prueba de la solución

los requisitos funcionales del sistema y ejercitándolos. Se considera que estas pruebas permiten encontrar (43):

1. Funciones incorrectas o ausentes.
2. Errores de interfaz.
3. Errores en estructuras de datos o en accesos a las Bases de Datos externas.
4. Errores de rendimiento.
5. Errores de inicialización y terminación.

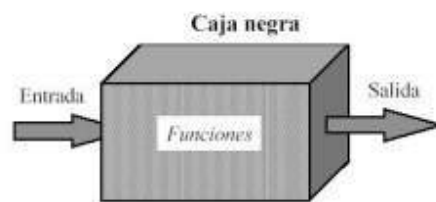


Figura 7 Método de Prueba, Caja Negra.

Para preparar los casos de pruebas hicieron falta un número de datos que ayudaron a la ejecución de los casos y que permitieron que el sistema se ejecute en todas sus variantes. Además, se utilizaron datos válidos e inválidos para el programa según si lo que se deseaba es hallar un error o probar una funcionalidad. Los datos se escogieron atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el mismo funcione correctamente.

Para desarrollar la prueba de caja negra se utilizó la técnica de Partición de Equivalencia, esta técnica es una de las más efectivas, pues permite examinar los valores válidos e inválidos de las entradas existentes en el software (44).

### 3.3.3.2 Diseño de Casos de Prueba.

A continuación se muestra la Sección 1 (Registrar Registro de Adquisición) del Caso de Prueba correspondiente al CU Gestionar Registro de Adquisición. Las celdas de las tablas que se muestran contienen las letras V, I o N/A. V indica válido, I indica inválido y N/A que no es necesario proporcionar un valor de la variable en este caso, ya que es irrelevante.

## *Capítulo 3: Implementación y prueba de la solución*

### SC1 Registrar Registro de Adquisición.

Escenario	Descripción	Título	Autor	Tipo de material	Sala de la biblioteca	Editor	Año de publicación	Editorial	Respuesta del sistema
EC1.1 Agregar un Registro de Adquisición	Permite al usuario agregar un Registro de Adquisición	V La Edad de Oro.	V José Martí.	V Revista	V Sala 1	V A. Dacosta Gómez.	V 1889	V Gente Nueva	<p>Permite introducir de forma obligatoria los datos del Registro de Adquisición :</p> <ul style="list-style-type: none"> <li>• Título</li> <li>• Autor</li> </ul> <p>Seleccionar:</p> <ul style="list-style-type: none"> <li>• Tipo de material</li> <li>• Sala de la biblioteca</li> </ul> <p>De forma opcional el usuario puede tener configurada la hoja de trabajo del Registro de Adquisición con algunos de los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Editor</li> <li>• Año de publicación</li> </ul>

### Capítulo 3: Implementación y prueba de la solución

									<p>Y permite:</p> <ul style="list-style-type: none"><li>• Aceptar la operación</li><li>• Cancelar la operación Valida los datos.</li></ul> <p>Verifica que no exista un Registro de Adquisición con título y autor coincidente.</p> <p>Le asigna de forma automática al Registro de Adquisición un número de control y lo guarda en la BD Catalográfica con estado: "Adquirido".</p> <p>Registra los datos del Registro de Adquisición y los datos del usuario que actúa en el sistema:</p> <ul style="list-style-type: none"><li>• Nombre y Apellidos</li></ul>
--	--	--	--	--	--	--	--	--	--

### Capítulo 3: Implementación y prueba de la solución

									<ul style="list-style-type: none"> <li>• Usuario</li> <li>• Fecha</li> <li>• Hora</li> </ul> <p>Muestra los datos del Registro de Adquisición. Ver SC 2: “Visualizar Registro de Adquisición”.</p> <p>Permite introducir los datos del Registro de un Ejemplar del Registro. (Adicionar Ejemplares) Ver: CP Gestionar Registro de Adquisición de Ejemplares.</p> <p>Muestra un mensaje de Información: “Se ha creado un Elemento”.</p>
EC1.2	Permite al usuario cancelar la operación.	NA	NA	NA	NA	NA	NA	NA	Cancela las operaciones realizadas sobre el Registro de Adquisición.
		V	I	V	V	V	I	V	

### Capítulo 3: Implementación y prueba de la solución

EC1.3 Datos son incorrectos.	Acción que realiza el sistema cuando el usuario introduce datos incorrectos.	La Edad de Oro.	123	Revista	Sala 1	A. Dacosta Gómez.	1889.	Gente Nueva	Muestra un mensaje de error: "Existen datos incorrectos, por favor rectifíquelos".  Muestra un indicador sobre el/los campos escritos incorrectamente.  Regresa al EC1.2
EC1.4 Campos vacíos.	Acción que realiza el sistema cuando el usuario deja campos vacíos.	I (vacío)	V (José Martí)	V (Revista)	V (Sala 1)	V (A. Dacosta Gómez)	V (1889)	V (Gente Nueva)	Muestra un mensaje de error: "Existen campos vacíos que son obligatorios, por favor complete estos campos".  Muestra un indicador sobre el/los campos vacíos.  Regresa al EC 1.2
EC1.5 Autor coincidente.	Acción que realiza el sistema cuando encuentra título y/o autor	V (La Edad de Oro)	V (José Martí)	V (Revista)	V (Sala 1)	V (A. Dacosta Gómez)	V (1889)	V (Gente Nueva)	Muestra un listado con los títulos y autores coincidentes con los siguientes datos:  • Número de control



## Capítulo 3: Implementación y prueba de la solución

	coincidente.								<ul style="list-style-type: none"> <li>• Título</li> <li>• Autor</li> <li>• Editor</li> <li>• Año de publicación</li> <li>• Tipo de material</li> <li>• Sala de la biblioteca</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>• Ver los detalles del Registro de Adquisición</li> <li>• Editar un Registro de Adquisición existente</li> <li>• Registrarlo como un nuevo material</li> </ul> <p>Registra el material como un nuevo registro.</p> <p>Regresa al EC 1.2</p>
EC1.6 Ver Detalles	Permite ver los detalles del	NA	NA	NA	NA	NA	NA	NA	Permite ver los detalles del Registro de Adquisición seleccionado. Ver

## Capítulo 3: Implementación y prueba de la solución

	Registro de Adquisición seleccionado.								SC2: “Visualizar Registro de Adquisición”.
EC1.7	Permite editar el Registro de Adquisición seleccionado.	V	V	V	V	V	V	V	Permite editar el Registro de Adquisición seleccionado. Ver SC3: “Editar Registro de Adquisición”.
Editar un Registro de Adquisición existente.	editar el Registro de Adquisición seleccionado.	La Edad de Oro.	José Martí.	Revista	Sala 1	A. Dacosta Gómez.	1889	Gente Nueva	

Tabla 7 Caso de Prueba Gestionar Registro de Adquisición, SC1 Registrar Registro de Adquisición.

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
1	Título	Campo de texto	No	Solo admite letras
2	Autor	Campo de texto	No	Solo admite letras
3	Tipo de material	Campo de texto	No	Solo admite letras
4	Sala de la biblioteca	Campo de texto	No	Admite letras y números
5	Editor	Campo de texto	No	Solo admite letras
6	Año de publicación	Campo de texto	No	Solo admite números
7	Editorial	Campo de texto	No	Solo admite letras

Tabla 8 Descripción de las variables.

## Capítulo 3: Implementación y prueba de la solución

### Resultado de las pruebas

A continuación en la Tabla 7 se muestran los resultados de las pruebas de caja negra que fueron realizadas al módulo Adquisición, en las que se puede observar el número de iteraciones que se le aplicaron los casos de pruebas, el total de no conformidades<sup>34</sup> encontradas y las que fueron resueltas, e igualmente se muestra la cantidad que no procedieron.

No. De iteraciones	Total de no conformidades	No procede	Resueltas
1ra Iteración	20	2	18
2da Iteración	2	0	2
3ra Iteración	0	0	0

Tabla 9 Resultados de las pruebas de Caja Negra aplicadas al módulo Adquisición.

De esta forma quedaron resueltas todas las no conformidades que se detectaron durante las pruebas realizadas al sistema. Se consiguieron resultados satisfactorios y se obtuvo un sistema que responde a las especificaciones de calidad de software. Las no conformidades encontradas se nombran a continuación:

Errores de implementación de las funcionalidades.

#### Gestionar Sugerencias

- Al consultar las Sugerencias aprobadas, se listaban las Sugerencias aprobadas y rechazadas.
- La interfaz Asociar/Disociar Sugerencia, no mostraba los datos pertenecientes a dicho CU.
- Las Sugerencias asociadas a una Desiderata no se añadían correctamente.

#### Gestionar Desiderata

- El estado de la Desiderata no quedaba bien definido al asociarse a la Orden de Compra.
- Se tenían que disociar todas las Sugerencias de la Desiderata para poder eliminarlo.
- Al exportar un PDF<sup>35</sup>, mostraba, solo parte del contenido.

<sup>34</sup> no conformidades: problemas detectados al probar la aplicación.

<sup>35</sup> PDF (sigla del inglés *Portable Document Format*, "formato de documento portátil") es un formato de almacenamiento para documentos digitales independiente de plataformas de software o hardware.

## Capítulo 3: Implementación y prueba de la solución

De modo general también se encontraron:

- Las fechas no se encontraban validadas ya que la fecha inicial podía ser mayor que la fecha final.
- Las fechas no se encontraban con el formato definido para el sistema ABCD versión 3.0.
- Al consultar, la estrategia de búsqueda no era sensible a minúsculas y mayúsculas.
- Varios campos donde solo debía admitir letras, permitía números y símbolos.
- No contenía el mismo formato de fecha en todas las interfaces.
- El campo “Precio de ejemplar” no permitía entrar números decimales.
- Falta de ortografía en varias interfaces.
- Al eliminar determinado elemento no mostraba un mensaje de confirmación.
- Después de realizar una consulta los campos mantenían los valores.
- Los mensajes que se muestran no estaban internacionalizados.
- Al exportar el documento no se mostraba internacionalizado.
- El formato de los componentes “*Label*” no se encontraba estandarizado para la aplicación.
- Los números de control no se encontraban validados.
- No existía concordancia entre el singular y el plural de los listados.

### 3.3.2 Pruebas de Integración

La prueba de integración es ejecutada para asegurar que los componentes en el modelo de implementación operen correctamente cuando son combinados con el sistema completo. Se prueba un paquete o un conjunto de paquetes del modelo de implementación. Estas pruebas descubren errores o especificaciones incompletas de las interfaces de los paquetes (43).

La utilización de Spring DM para la creación de *beans* es una forma de integración que permite la configuración de los mismos. Estos hacen referencia a los servicios con el objetivo de recibir notificaciones cuando están configurados, que puede ser de forma dependiente o independiente por Spring DM. De los métodos proporcionados por Spring DM para la configuración de *beans*, el utilizado por los desarrolladores del sistema ABCD versión 3.0 consiste en especificar en la etiqueta de referencia una configuración de escucha de dicha referencia del servicio. Este enfoque permite utilizar un POJO<sup>36</sup> como *listener* (oyente). La asociación entre eventos y métodos de devolución de llamada se especifica a través de la configuración

---

<sup>36</sup> Un **POJO** (acrónimo de *Plain Old Java Object*), enfatiza el uso de clases simples y que no dependen de un framework en especial.

## Capítulo 3: Implementación y prueba de la solución

de escucha de la referencia del servicio solicitado. A continuación se muestra una interfaz en la que se visualiza la utilización de *beans*.

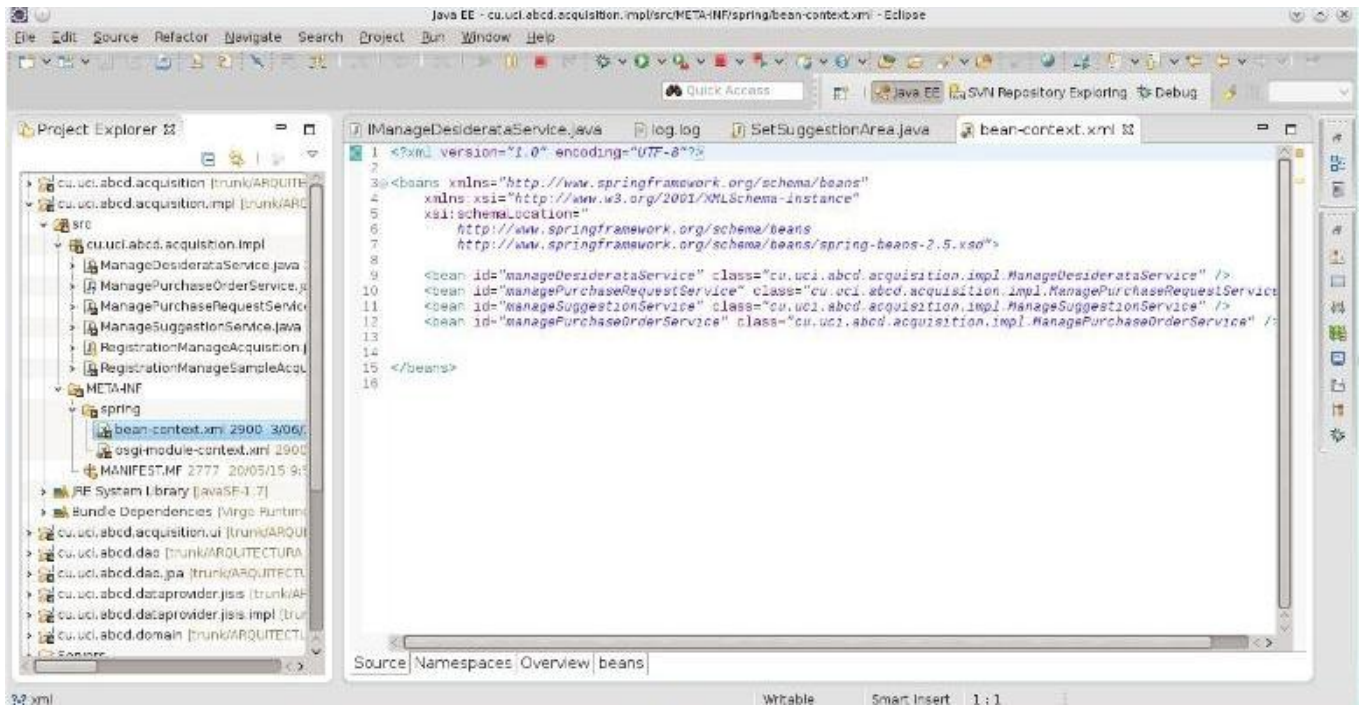


Figura 8 Ejemplo de utilización de beans

Las pruebas de integración fueron realizadas verificando que:

## Capítulo 3: Implementación y prueba de la solución

1. En la carpeta “resources” encontrada en el directorio `servidor\abcdconfig\resources`, se localizan las imágenes e íconos utilizados para la construcción del SIGB ABCD versión 3.0. La Figura 10 muestra parte de los elementos contenidos en dicha carpeta.

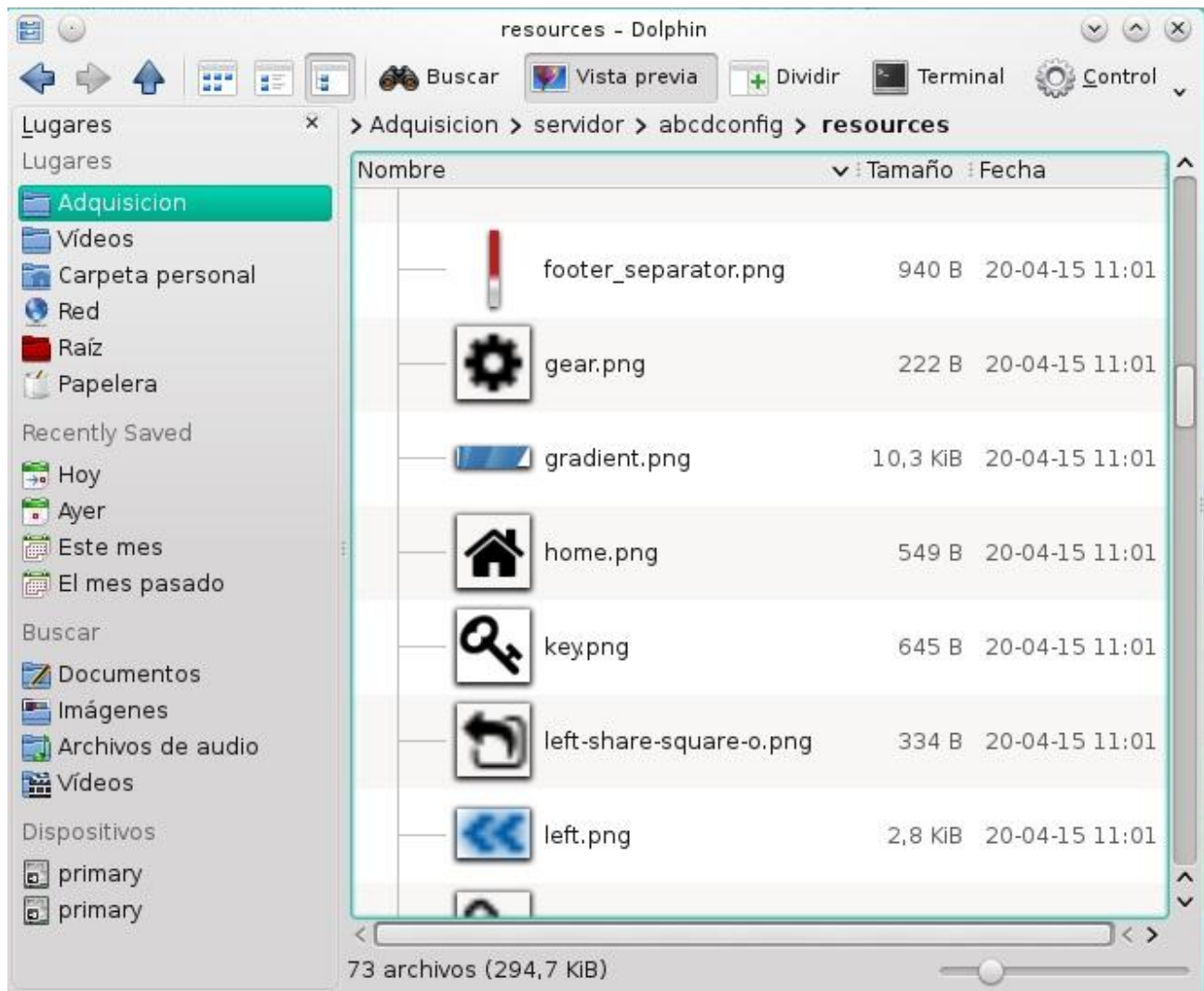


Figura 9 Carpeta de resources del sistema ABCD versión 3.0.

2. En la carpeta “usr” encontrada en el directorio `servidor\repository\usr`, se sitúan los *bundles* y bibliotecas (internas y externas) necesarias para que el sistema ABCD versión 3.0 funcione correctamente. A continuación se muestran de los *bundles* pertenecientes al módulo Adquisición, los paquetes específicos que no contiene el servidor Eclipse Virgo y que son requerimientos mínimos para que el sistema levante, además la Figura 11 ilustra parte del contenido de la carpeta *usr*.

## Capítulo 3: Implementación y prueba de la solución

### ***cu.uci.abcd.adquisition***

- org.eclipse.rap.rwt
- org.springframework.data.domain
- org.springframework.jpa.domain.repository
- org.unesco.jisis.corelib.exceptions
- org.unesco.jisis.corelib.record

### ***cu.uci.abcd.adquisition.impl***

- org.springframework.data.core
- com.adobe.xmp
- javax.persistence
- javax.persistence.criteria

### ***cu.uci.abcd.adquisition.impl***

- org.eclipse.rap.rwt
- javax.persistence.criteria
- javax.servlet.http
- org.eclipse.rap.rwt
- org.springframework.data.domain
- org.springframework.data.domain.jaxb

## Capítulo 3: Implementación y prueba de la solución

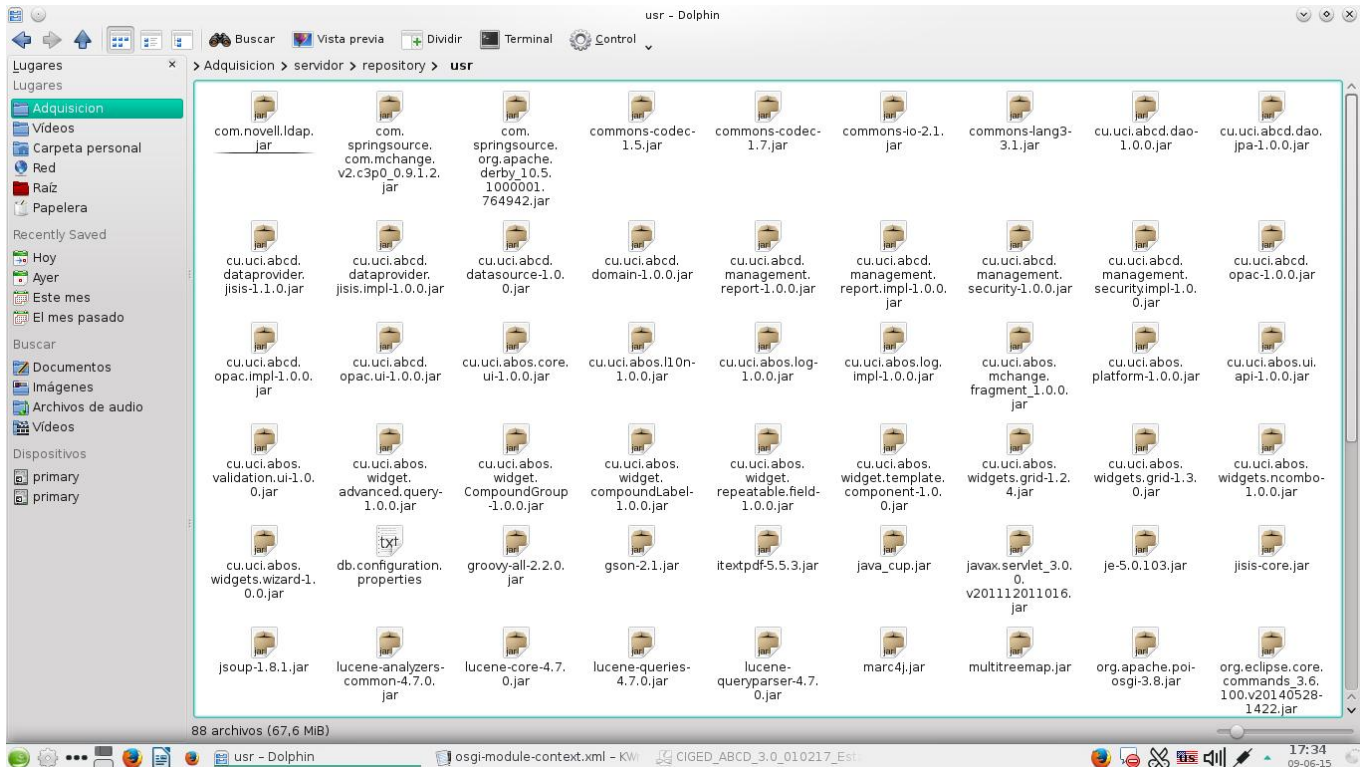


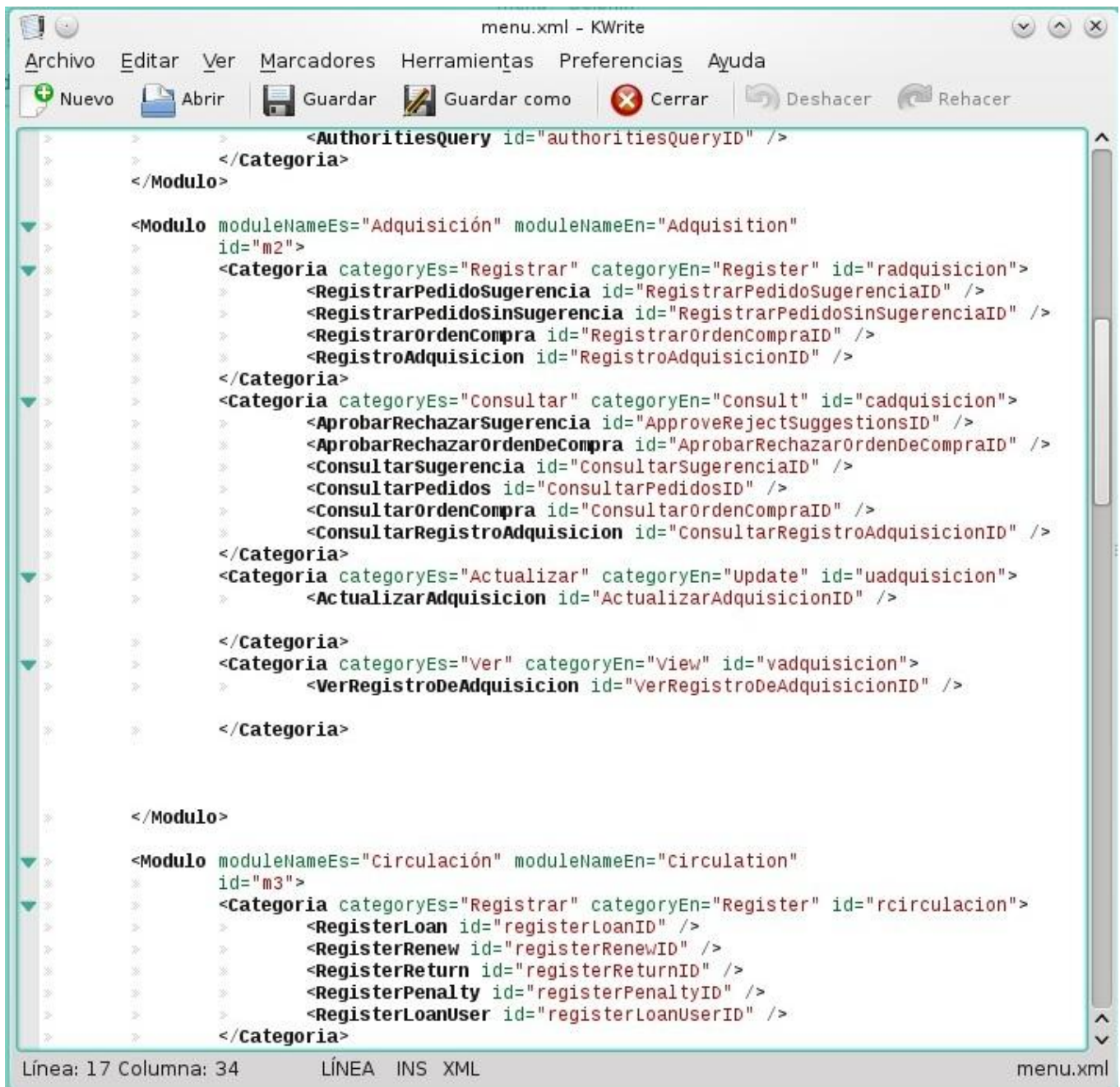
Figura 10 Contenido de la carpeta *usr* del sistema ABCD versión 3.0

El módulo Adquisición del sistema ABCD versión 3.0 también se integra con el módulo OPAC, en donde los usuarios realizan sugerencias de materiales bibliográficos que luego permitirán ser aprobadas o rechazarlas por el Gestor de Sugerencias. Adquisición, se integra además con Administración de Bibliotecas, al ser aprobada una *Solicitud de Orden de Compra*, la misma pasará a ser una *Orden de Compra*, donde el Gestor de Órdenes de Compra escogerá el proveedor (gestionado en Administración de bibliotecas) y podrá ajustar los precios y cantidades de materiales bibliográficos a comprar.

3. En el archivo `.xml` encontrado en el directorio `servidor\abcdconfig\menú\menu.xml`, estén definidos los identificadores asociados a las interfaces agrupadas en el menú del módulo Adquisición. Para que una interface sea mostrada correctamente, en la implementación se debe utilizar el mismo nombre definido para los identificadores en dicho archivo. A continuación se muestra parte del contenido del mismo como elemento integrador para visualizar lo expresado anteriormente.



## Capítulo 3: Implementación y prueba de la solución



```
<AuthoritiesQuery id="authoritiesQueryID" />
</Categoria>
</Modulo>

<Modulo moduleNameEs="Adquisición" moduleNameEn="Acquisition"
id="m2">
  <Categoria categoryEs="Registrar" categoryEn="Register" id="radquisicion">
    <RegistrarPedidoSugerencia id="RegistrarPedidoSugerenciaID" />
    <RegistrarPedidoSinSugerencia id="RegistrarPedidoSinSugerenciaID" />
    <RegistrarOrdenCompra id="RegistrarOrdenCompraID" />
    <RegistroAdquisicion id="RegistroAdquisicionID" />
  </Categoria>
  <Categoria categoryEs="Consultar" categoryEn="Consult" id="cadquisicion">
    <AprobarRechazarSugerencia id="ApproveRejectSuggestionsID" />
    <AprobarRechazarOrdenDeCompra id="AprobarRechazarOrdenDeCompraID" />
    <ConsultarSugerencia id="ConsultarSugerenciaID" />
    <ConsultarPedidos id="ConsultarPedidosID" />
    <ConsultarOrdenCompra id="ConsultarOrdenCompraID" />
    <ConsultarRegistroAdquisicion id="ConsultarRegistroAdquisicionID" />
  </Categoria>
  <Categoria categoryEs="Actualizar" categoryEn="Update" id="uadquisicion">
    <ActualizarAdquisicion id="ActualizarAdquisicionID" />
  </Categoria>
  <Categoria categoryEs="Ver" categoryEn="View" id="vadquisicion">
    <VerRegistroDeAdquisicion id="VerRegistroDeAdquisicionID" />
  </Categoria>
</Modulo>

<Modulo moduleNameEs="Circulación" moduleNameEn="Circulation"
id="m3">
  <Categoria categoryEs="Registrar" categoryEn="Register" id="rcirculacion">
    <RegisterLoan id="registerLoanID" />
    <RegisterRenew id="registerRenewID" />
    <RegisterReturn id="registerReturnID" />
    <RegisterPenalty id="registerPenaltyID" />
    <RegisterLoanUser id="registerLoanUserID" />
  </Categoria>
</Modulo>
```

Línea: 17 Columna: 34      LÍNEA INS XML      menu.xml

Figura 11 Archivo .xml encontrado en el directorio servidor\abcdconfig\menú\menu.xml.

De esta manera los módulos del sistema ABCD realizan correctamente su función por separados, y al probarse conjuntamente se evita que uno de ellos pueda tener un efecto adverso o inadvertido sobre otro módulo. Además, las subfunciones cuando se combinan, producen la función requerida para el módulo.

## *Capítulo 3: Implementación y prueba de la solución*

### **Conclusiones del capítulo**

En el presente capítulo se obtuvo como resultado los diagramas de componentes (utilizando como base el CU Gestionar Registro de Adquisición), lo que permitió describir los elementos físicos del sistema y sus relaciones. Además, se muestra el diagrama de despliegue del módulo Adquisición, el mismo facilita mostrar la disposición de las particiones físicas del sistema y la asignación de los componentes software a estas particiones. Por último, se aplicaron pruebas de Caja Negra e Integración utilizando para realizar las pruebas de Caja Negra la técnica de Partición de Equivalencia. Se efectuaron tres iteraciones, encontrando en cada una de ellas varias no conformidades pero siendo todas corregidas, también se aplicó la prueba de integración, comprobando de esta forma que el módulo implementado se integre correctamente con los demás módulos correspondientes al sistema ABCD versión 3.0 y siendo satisfactorios los resultados arrojados.

## *Capítulo 3: Implementación y prueba de la solución*

### CONCLUSIONES

Una vez concluido el desarrollo del presente trabajo, se puede afirmar que los objetivos propuestos para la realización del mismo se cumplieron exitosamente.

- A partir de la elaboración del marco teórico de la investigación se enriqueció el conocimiento acerca de los procesos de adquisición en Sistemas Integrados de Gestión Bibliotecaria.
- Se realizó el diseño del sistema generando cada uno de los artefactos como, diagramas de clases, diagramas de interacción, modelado de la BD, lo que permitió dar paso a la implementación del módulo de adquisición.
- Se implementó el módulo Adquisición del sistema ABCD versión 3.0 a partir de los requerimientos del software establecidos, haciendo uso de la arquitectura y las tecnologías definidas por el equipo de desarrollo de software de dicho sistema. También se diseñaron y aplicaron las pruebas definidas, probando cada una de las funcionalidades del módulo para verificar el correcto funcionamiento del mismo.

### RECOMENDACIÓN

- Implementar versiones posteriores que perfeccionen el módulo Adquisición del sistema ABCD versión 3.0.

### REFERENCIAS BIBLIOGRAFICAS

1. CRESPO, Luis Bermello. Bibliotecas digitales y actividad bibliotecaria. *Ciencias de la Información*,. 2001. 1 p. 57-68.
2. Gethin, Peter. El profesional de la información. *El profesional de la información*. [En línea] 2001. [Citado el: 18 de Junio de 2015.] <http://elprofesionaldelainformacion.com/contenidos/2001/noviembre/7.pdf>.
3. Española, Real Academia. Diccionario de la lengua española. *Diccionario de la lengua española*. [En línea] Real Academia Española © Todos los derechos reservados, 2014. [Citado el: 7 de Diciembre de 2014.] <http://lema.rae.es/drae/?val=biblioteca>.
4. Ben White, Jefe de Propiedad Intelectual, British Library. OMPI. OMPI. [En línea] Agosto de 2012. [Citado el: 2 de Mayo de 2015.] [http://www.wipo.int/wipo\\_magazine/es/2012/04/article\\_0004.html](http://www.wipo.int/wipo_magazine/es/2012/04/article_0004.html).
5. Médicas, Infomed - Centro Nacional de Información de Ciencias. Infomed. Infomed. [En línea] 2015. [Citado el: 2 de Mayo de 2015.] <http://www.sld.cu/sitios/bmn/temas.php?idv=2991>.
6. Zurita Sanchez, Juan M. academia.edu. academia.edu. [En línea] [Citado el: 30 de Abril de 2015.] [http://www.academia.edu/3642366/Software\\_libre\\_una\\_alternativa\\_para\\_la\\_gesti%C3%B3n\\_de\\_Informaci%C3%B3n\\_en\\_bibliotecas](http://www.academia.edu/3642366/Software_libre_una_alternativa_para_la_gesti%C3%B3n_de_Informaci%C3%B3n_en_bibliotecas).
7. WALSH, José Antonio Magán. Editorial Complutense. s.l. : Tratado básico de biblioteconomía, 2004.
8. Gavilán, César Martín. Criterios para la formación, mantenimiento y evaluación de la colección bibliográfica. s.l. : Selección y adquisición de materiales, 2008.
9. Biblioteca Nacional de Cuba "José Martí". *Manual Procesos Técnicos en las Bibliotecas Públicas*. Ciudad de La Habana : Biblioteca Nacional de Cuba "José Martí", 2010. 978-959-7137-45-0.
10. Gavilán, Martín Cesar. eprints. eprints. [En línea] 26 de Septiembre de 2008. [Citado el: 18 de Junio de 2015.] <http://eprints.rclis.org/13188/1/sigb.pdf>.
11. *Revista cubana de los profesionales de la información y la comunicación en salud*. Navarrete, Óscar Arriola. 2014.
12. Gutierrez, Coral Luis Alberto. eprints. eprints. [En línea] 2011. [Citado el: 30 de Abril de 2015.] <http://eprints.rclis.org/17913/1/Luis-alberto%20gutierrez-coral.PDF>.
13. Smet, Egbert. *The abc of ABCD Central :the Reference Manual*. 2013.
14. Romero, Hemeregildo. *Metodologías de Desarrollo*. [Aplicación Electrónica] 2009.
15. LOUDEN, Kenneth C. *Lenguajes de programación: Principios y práctica*. Cengage Learning Latin America. 2004.

## *Glosario de términos*

16. RUEDA, Julio. Aplicación de la metodología RUP para el desarrollo rápido de aplicaciones basado en el estándar j2ee. Trabajo de grado. Facultad de Ingeniería. Universidad de San Carlos de Guatemala. Guatemala : s.n., 2006.
17. Larman, Craig. UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al. 2002.
18. NAUGHTON, Patrick y SCHILDT, Herbert. Java. s.l. : McGraw-Hill, 1997.
19. LÓPEZ, P. Ingeniería del software I. Práctica 1. Herramienta CASE Visual Paradigm. 2012.
20. The Eclipse Foundation. Eclipse. [En línea] 2014. <http://www.eclipse.org/>.
21. SÁNCHEZ, Gregorio Cabrera. Sistemas gestores de bases de datos: administración de sistemas informáticos. Paraninfo/Thomson Learning. s.l. : Paraninfo/Thomson Learning, 2001.
22. GONGORA. L. Arquitectura de Software de un Sistema Integrado para la Biblioteca Nacional José Martí. Ciudad Habana : Universidad de las Ciencias Informática, 2007. 120. p.
23. [Isis-users] Report on the 2nd online presentation of ISIS & JISIS. [Isis-users] Report on the 2nd online presentation of ISIS & JISIS. [En línea] Mayo de 2012. [Citado el: 4 de Marzo de 2014.] <http://lists.iccisis.org/pipermail/isis-users/2012-May/001818.html>.
24. Fundación Eclipse. Eclipse. Eclipse. [En línea] 9 de Abril de 2014. [Citado el: 19 de Marzo de 2015.] <http://www.eclipse.org/virgo/>.
25. Copyright © 2015 The Eclipse Foundation. All Rights Reserved. Remote Application Platform. Remote Application Platform. [En línea] 2015. [Citado el: 11 de Mayo de 2015.] <http://eclipse.org/rap/>.
26. The Eclipse Foundation. Remote Application Platform. Remote Application Platform. [En línea] 2015. [Citado el: 20 de Junio de 2015.] <http://www.eclipse.org/rap/>.
27. Miguel, Roberto Montero. OSGi. OSGi. [En línea] [Citado el: 20 de Junio de 2015.] [http://static1.1.sqspcdn.com/static/f/923743/15025178/1320739253190/OSGI\\_Roberto\\_Montero.pdf?token=Eq1hEgb78PYxvuaSJJ35DvCnhDs%3D](http://static1.1.sqspcdn.com/static/f/923743/15025178/1320739253190/OSGI_Roberto_Montero.pdf?token=Eq1hEgb78PYxvuaSJJ35DvCnhDs%3D).
28. Torres, Calas Abraham, Alfonzo, Pérez Damian, Abreu, Peña Marieta. ComuMat. ComuMat. [En línea] 2015. [Citado el: 20 de Junio de 2015.] [https://compumat.uci.cu/sites/default/files/public/p909-ponencia-3624\\_0.pdf](https://compumat.uci.cu/sites/default/files/public/p909-ponencia-3624_0.pdf).
29. Sagaya Aurelia1, Xavier Patrick Kishore, Osmer Saleh. Inase. Inase. [En línea] [Citado el: 23 de Abril de 2015.] <http://www.europment.org/library/2014/santorini/bypaper/COMPUTERS/COMPUTERS2-29.pdf>.
30. Gary Mak, Daniel Rubio. Pro Spring Sourse DM Server. 2009.
31. © 2015 The Eclipse Foundation. All Rights Reserved. . EclipseLink. EclipseLink. [En línea] 2015. [Citado el: 22 de Junio de 2015.] <https://eclipse.org/eclipselink/>.

## *Glosario de términos*

32. Vega, Miguel. Departamento de lenguajes y Sistemas Informáticos. Departamento de lenguajes y Sistemas Informáticos. [En línea] Octubre de 2010. [Citado el: 30 de Abril de 2015.] <http://lsi.ugr.es/~mvega/docis/casos%20de%20uso.pdf>.
33. Medina, Ing. Andy Cabrera. 0114\_ESPECIFICACIÓN DE CASOS DE USO. Ciudad Habana : s.n., 2015.
34. Martín, Eva Leonart. [En línea] [Citado el: 30 de Abril de 2015.] [users.dsic.upv.es/asignaturas/facultad/lsi/trabajos/142000.doc](http://users.dsic.upv.es/asignaturas/facultad/lsi/trabajos/142000.doc).
35. Peláez, Juan. Arquitectura basada en capas. Arquitectura basada en capas. [En línea] 26 de Mayo de 2009. [Citado el: 3 de Mayo de 2015.] <http://www.juanpelaez.com/geek-stuff/arquitectura/arquitectura-basada-en-capas/>.
36. Departamento de Informatica Universidad Técnica Federico Santa María. Departamento de Informatica Universidad Técnica Federico Santa María. [En línea] [Citado el: 10 de Marzo de 2015.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.
37. REYES, Ricardo Armando Machorro. Los Patrones como un Medio del Diseño Orientado a Objetos.
38. CECHICH, Alejandra y MOORE, Richard. Un modelo formal de patrones orientados a objetos. En II Workshop de Investigadores en Ciencias de la Computación. 2000.
39. CUADRADO GALLEGO, JUAN JOSE. Análisis y diseño estructurado y orientado a objetos de sistemas informáticos. 2003.
40. Torossi, Marcelo Gustavo. UNIVERSIDAD NACIONAL DEL NORDESTE. Facultad de Ciencias Exactas y Naturales y Agrimensura. [En línea] [Citado el: 27 de Mayo de 2015.] [http://exa.unne.edu.ar/informatica/anasistem2/public\\_html/apuntes/de1.pdf](http://exa.unne.edu.ar/informatica/anasistem2/public_html/apuntes/de1.pdf).
41. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Madrid : s.n., 2000. 84-7829-036-2.
42. TORRES, Patricio Letelier. Desarrollo de Software Orientado a Objeto usando UML. Universidad Politecnica de Valencia (UPV)–España. 2004.
43. Pressman, Roger. Ingeniería de Software "Un Enfoque Práctico". s.l. : Mc Graw Hill, 2005.
44. PÉREZ, Beatriz. Proceso de Testing Funcional Independiente. 2006. Tesis Doctoral. Tesis de Maestría en Informática, Directora: Szasz, N., PEDECIBA Informática, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay. [En línea]
45. Online Language Dictionaries. wordreference.com. wordreference.com. [En línea] [Citado el: 28 de Marzo de 2015.] <http://www.wordreference.com/definicion/desiderata>.

## *Glosario de términos*

46. © 2015 WordReference.com. WordReference.com. WordReference.com. [En línea] 2015. [Citado el: 10 de Mayo de 2015.] <http://www.wordreference.com/definicion/transgresi%C3%B3n>.
47. eclipse. eclipse. [En línea] Eclipse documentation - Current Release, 2014. [Citado el: 28 de Mayo de 2015.]  
<http://help.eclipse.org/luna/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Freference%2Fapi%2Forg%2F eclipse%2Fcore%2Fruntime%2Fclass-use%2FContributor.html>.
48. SearchSoa. SearchSoa. [En línea] TechTarget, 2015. [Citado el: 5 de Junio de 2015.]  
<http://searchsoa.techtarget.com/definition/Rich-Internet-Application-RIA>.
49. Copyright © 2015 The Eclipse Foundation. Eclipse. Eclipse. [En línea] 2015. [Citado el: 5 de Junio de 2015.] <https://www.eclipse.org/swt/>.
50. CAL, Wilson Alba y PARRA, Yamila Estrada. Sistema de administración de plugins & plugin de integración con centrales telefónicas. Serie Científica, 2012, vol. 5, no 12. [En línea]
51. © Fundación Universitaria San Pablo CEU - Red de Bibliotecas . Bibliotecas Universitarias CEUNET. Bibliotecas Universitarias CEUNET. [En línea] 25 de Abril de 2007. [Citado el: 19 de Junio de 2015.]  
<http://www.bibliotecaceu.es/servicios/adquisiciones/tramite-pedidos>.
52. www.itec-sde.eu 2014 . SDE. SDE. [En línea] 2014. [Citado el: 19 de Junio de 2015.] [http://www.itec-sde.info/es/search\\_results?utf8=%E2%9C%93&search=qooxdoo&commit=Buscar](http://www.itec-sde.info/es/search_results?utf8=%E2%9C%93&search=qooxdoo&commit=Buscar).
53. © 2005-2015 Mozilla Developer Network and individual contributors. Mozilla Developer Network . Mozilla Developer Network . [En línea] 2005. [Citado el: 19 de Junio de 2015.]  
<https://developer.mozilla.org/es/docs/Web/HTML>.
54. Copyright www.itec-sde.eu 2014 . SDE. SDE. [En línea] [Citado el: 20 de Junio de 2015.] [http://www.itec-sde.info/es/search\\_results?utf8=%E2%9C%93&search=qooxdoo&commit=Buscar](http://www.itec-sde.info/es/search_results?utf8=%E2%9C%93&search=qooxdoo&commit=Buscar).
55. Copyright © 1993, 2014, Oracle and/or its affiliates. All rights reserved. Oracle Help Center. Oracle Help Center. [En línea] 2014. [Citado el: 20 de Junio de 2015.]  
<http://docs.oracle.com/javase/7/docs/api/java/lang/ClassLoader.html>.
56. Copyright 1999-2015 by Refsnes Data. All Rights Reserved. W3schools.com. W3schools.com. [En línea] 2015. [Citado el: 20 de Junio de 2015.] [http://www.w3schools.com/xml/xml\\_what.asp](http://www.w3schools.com/xml/xml_what.asp).
57. Creado por vBulletin© Copyright © 2015 Soluciones vBulletin Inc. © IGTNetwork.net - 1998 vBulletin Skin Design By: Dkdexter. programaciondesarrollo.es. programaciondesarrollo.es. [En línea] 15 de Febrero de 2011. [Citado el: 19 de Junio de 2015.] <http://programaciondesarrollo.es/que-es-un-entorno-de-desarrollo-integrado-ide/>.



GLOSARIO DE TÉRMINOS

TIC: Tecnologías de la información y la comunicación.

SIGB: Sistemas Integrados de Gestión Bibliotecaria.

OPAC: Catálogo de Acceso Público en Línea.

ABCD: Automatización de Bibliotecas y Centros de Documentación.

UCI: Universidad de las Ciencias Informáticas.

CIGED: Centro de Informatización de la Gestión Documental.

RUP: Proceso Unificado de Racional.

GNU: Licencia General Pública.

UML: Lenguaje Unificado de Modelado.

CASE: Ingeniería de Software Asistida por Computadora.

JSF: Java Server Faces.

IU: Interfaz de usuario.

API: Interfaz de Programación de Aplicaciones.

JSP: Java Server Pages.

IDE: Entornos Integrados de Desarrollo.

## ANEXOS

### **Anexo 1: Preguntas de la Entrevista al Cliente.**

**Andy Cabrera Medina:** Especialista "B" en Ciencias Informáticas perteneciente al Centro de Informatización de la Gestión Documental del Departamento de Desarrollo de Componentes ABCD versión 1.2.

1. ¿Cuál es el objetivo principal del ABCD versión 1.2, y sus funciones principales?
2. ¿Cuáles son los procesos que se realizan actualmente dentro del flujo de trabajo del módulo Adquisición del sistema ABCD versión 1.2?
3. ¿Qué tipo de materiales se pueden tratar en el módulo de adquisición?
4. ¿Cuáles son las vías de adquisición que se le añadirán al módulo de adquisición en la versión 3.0?
5. ¿Cómo se realiza el proceso de adquisición para las diferentes vías de adquisición existentes?
6. ¿Cuáles son los principales cambios en las funcionalidades que se desea hacer con la nueva versión del módulo Adquisición del sistema ABCD?
7. ¿Con que frecuencia se inspecciona la lista de nuevas Sugerencias?
8. ¿Recibe notificación el usuario cuando su Sugerencia fue aceptada o rechazada?
9. ¿Cuáles son los principales procesos del sistema?
10. ¿Cuál es el objetivo principal del módulo Adquisición del sistema ABCD versión 3.0, y sus funciones principales?

Respuestas del Especialista "B" en Ciencias Informáticas perteneciente al Centro de Informatización de la Gestión Documental del Departamento de Desarrollo de Componentes ABCD versión 1.2:

1. El objetivo principal de ABCD versión 1.2 es informatizar el proceso de gestión de materiales bibliográficos en las bibliotecas y centros de documentación. Sus funciones principales son: adquisición, catalogación, circulación y consulta de materiales.
2. Los procesos que se realizan actualmente dentro del flujo de trabajo del módulo Adquisición del sistema ABCD versión 1.2 son: Sugerir un material bibliográfico, realizar un pedido, crear una orden de compra y registrar el material bibliográfico.

3. Los tipos de materiales se pueden tratar en el módulo de adquisición son: libros, revistas, periódicos, documentos, diccionarios, mapas, etc.
4. La vía de adquisición que se le añadirá al módulo de adquisición en la versión 3.0 es: Depósito Legal.
5. Existen tres vías de adquisición, estas son: Donación, Canje, Préstamo y Compra, para las tres primeras solo se registra el material y el proceso termina; para la Compra, primeramente se sugiere el material, luego se crea la *Desiderata*, se crea la Solicitud de Compra, se crea la Orden de Compra y por último se registra el material bibliográfico.
6. Los principales cambios en las funcionalidades que se desea hacer con la nueva versión del módulo Adquisición del sistema ABCD son:
  - Realizar las sugerencias de materiales bibliográficos desde el módulo OPAC.
  - Cambiar el término *Pedido* por *Desiderata* y varios de los valores que se registran en la misma al crearla.
  - Realizar una Solicitud de Compra antes de la Orden de Compra.
7. Se inspecciona la lista de nuevas Sugerencias cada vez que el usuario lo desee.
8. Si, el usuario recibe una notificación cuando su Sugerencia fue aceptada o rechazada, y el motivo por el cual se tomó determinada decisión.
9. Los principales procesos del sistema ABCD versión 3.0 son: Sugerir material, Crear la *Desiderata*, Crear la Solicitud de Compra, Crear la Orden de Compra y Registrar el material bibliográfico.
10. El objetivo principal de ABCD versión 3.0 es perfeccionar el proceso de adquisición de materiales bibliográficos en las bibliotecas