

Universidad de las Ciencias Informáticas

Facultad 1



**Desarrollo del módulo de administración de recursos  
digitales para el Sistema Integrado de Gestión Bibliotecaria  
bibliotheX.**

**Trabajo de Diploma para optar por el Título de  
Ingeniero en Ciencias Informáticas**


**Autor:** Lissandra Diego Cisnero.

**Tutor:** Ing. Raudel González Echenique.

**Co Tutor:** Ing. Adnier Roselló Carrazana.

**Ciudad de La Habana, 2011.**

**“Año 53 de la Revolución.”**



*“El futuro de nuestra Patria tiene que ser necesariamente un futuro de hombres de ciencia, de hombres de pensamiento, porque precisamente es lo que más estamos sembrando, lo que más estamos sembrando son oportunidades a la inteligencia”.*

*Fidel Castro Ruz.*

## DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Lissandra Diego Cisnero

---

Ing. Raudel González Echenique

---

Ing. Adnier Roselló Carrazana

## **DATOS DE CONTACTO**

### **Tutores:**

#### **Tutor principal:**

**Ing. Raudel González Echenique**

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el año 2008. En la actualidad pertenece al departamento de Sistemas Digitales de la Facultad 1 y se desempeña como profesor adjunto a la producción en el departamento Gestión Documental y Archivística perteneciente al centro de producción GEITEL de la facultad 1.

#### **Co tutor:**

**Ing. Adnier Roselló Carrazana**

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el año 2010. En la actualidad pertenece al departamento Gestión Documental y Archivística perteneciente al centro de producción GEITEL de la facultad 1.

## **DEDICATORIA**

*Dedico este trabajo de diploma a mi mamá (Belkys), mi papá (Felipe) y mi abuela (Dilvia) por ser las personas más importantes en mi vida y a las que más quiero porque me han dado lo necesario para llegar hasta aquí.*

## **AGRADECIMIENTOS**

*A mi mamá, mi abuela y mi papá por darme amor, brindarme el soporte que necesitaba y ayudarme a hacer realidad este sueño.*

*A mi tío Jorge, por estar ahí conmigo para apoyarme en todo momento casi como un padre.*

*A Tatiana y Mandy que durante siete años han sido parte de la familia y me han apoyado, aunque los sigo queriendo de lejos.*

*A demás familiares que se preocuparon por mis resultados docentes y por mí en sentido general. En especial a mi tía Doris.*

*Al grupito de bachatas en mi casa, Gelsys, Yanet, Haniel, Edito, Julio, Serguey y Raydel que aunque nos abandonó a última hora me ayudó muchísimo, dándome fuerzas para seguir adelante cada día durante los años anteriores. Por último agradecer a Puente y José que aunque se incorporaron al final al grupito, se han convertido en grandes amigos para mí.*

*Voy a referirme en particular a Gelsys, que ha sido y es la mejor amiga que he tenido y que nos hemos apoyado una a la otra durante estos cinco años, cada vez que atravesábamos algún momento difícil y claro para el bonche también.*

*A Yenisleydi Careaga Cristo que más que una de las mejores profesoras que he tenido, ha sido una especie de publicista ya que gracias a ella me decidí a bailar español en los festivales y demás actividades a las que me envió.*

*A mis tutores, por ayudarme a superar este difícil momento que significó la tesis para mí.*

*A los compañeros con que compartí el aula a lo largo de los cinco años.*

*A las compañeras de apartamento.*

*A los profesores que colaboraron en estos años con mi formación profesional.*

*A la UCI, a la Revolución, por haberme permitido vivir la mejor experiencia de mi vida al poder compartir durante estos cinco años con grandes amigos.*

*A todos aquellos que hicieron que de esta experiencia surgiera una mejor Lissandra.*

## RESUMEN

El Sistema Integrado de Gestión Bibliotecaria (SIGB) bibliotheX es un sistema automatizado desarrollado en la Universidad de las Ciencias Informáticas por el proyecto Gestión Bibliotecaria de la Facultad 1, basado en el SIGB Koha, de distribución libre para agilizar los procesos y servicios que puede brindar una biblioteca digital. Este sistema cuenta con diversos módulos, entre los que destacan los siguientes: Adquisición, Circulación, Catalogación, Reportes y OPAC (Catálogo en Línea). Actualmente este sistema no cuenta con un módulo que se encargue de administrar los recursos digitales de que dispone la biblioteca.

En el presente trabajo de diploma se propone llevar a cabo el desarrollo de un módulo que se integre al SIGB bibliotheX para que agilice la administración de los recursos digitales que el mismo manipula y satisfacer las necesidades de los usuarios. Este módulo permitirá principalmente que el usuario pueda cargar o descargar recursos digitales y a su vez permitirá que la biblioteca pueda controlar estos procesos.

Para el desarrollo de este subsistema se utiliza la metodología de desarrollo RUP, el lenguaje de modelado UML y la herramienta CASE Visual Paradigm para el diseño del mismo. Para la implementación, serán utilizados el servidor de aplicaciones Web Apache, el lenguaje de programación Perl y MySQL como gestor de base de datos relacional.

### **Palabras clave:**

SIGB bibliotheX, administración de recursos digitales, descargar recursos digitales, cargar recursos digitales, módulos.

## ÍNDICE

Introducción.....	1
Capítulo 1. Fundamentación Teórica .....	5
1.1. Administración de recursos .....	5
1.1.1. Definiciones de recursos digitales.....	5
1.2. Administración de recursos digitales Open Source .....	6
1.3. MySQL y administración de recursos digitales .....	9
1.4. Manejo de recursos digitales con el tipo de datos BLOB .....	10
1.5. Sistemas de archivos y almacenamiento de recursos en el módulo de administración de recursos digitales .....	10
1.6. Lenguajes del sistema .....	10
1.6.1. Lenguaje del lado del servidor .....	11
1.6.2. Lenguajes del lado del cliente .....	11
1.6.3. Lenguaje de modelado .....	12
1.7. Herramientas y tecnologías a utilizar .....	14
1.8. Metodologías de desarrollo de software.....	18
Conclusiones del capítulo .....	20
Capítulo 2. Características del Sistema .....	21
2.1. Análisis crítico de los procesos de gestión y organización de los recursos digitales.....	21
2.2. Objeto de automatización .....	21
2.2.1. Descripción de los procesos que serán objeto de automatización .....	21
2.2.2. Descripción de los sistemas automatizados que existen en la entidad y que están vinculados con el campo de acción del proyecto.....	22



2.3. Información que se maneja .....	22
2.4. Propuesta de sistema .....	23
2.4.1. Descripción general de la propuesta de sistema, como debe funcionar .....	23
2.5. Modelamiento del Negocio .....	23
2.6. Especificación de los requisitos de software.....	25
2.6.1. Requisitos funcionales .....	25
2.6.2. Requisitos no funcionales .....	26
2.7. Modelado del Sistema .....	27
2.7.1. Definición de los actores del sistema.....	27
2.7.2. Listado de casos de uso.....	27
2.7.3. Diagrama de casos de uso.....	28
2.7.4. Casos de uso expandidos .....	29
Conclusiones del capítulo .....	35
Capítulo 3. Análisis y diseño del sistema.....	36
3.1 Análisis .....	36
3.2. Diseño .....	39
3.2.1. Diagramas de interacción.....	39
3.2.2. Diagramas de clase del diseño .....	43
3.2.3. Diagrama Entidad-Relación de la Base de datos .....	46
3.2.4. Descripción de las tablas de la base de datos .....	47
3.3. Arquitectura .....	51
3.4. Tratamiento de errores .....	53
3.5. Seguridad .....	53

3.6. Concepción de la ayuda .....	53
Conclusiones del capítulo .....	53
4.1. Implementación.....	54
4.1.1. Diagrama de despliegue .....	54
4.1.2. Diagrama de componentes .....	55
4.2. Modelo de prueba .....	58
Conclusiones del capítulo .....	60
Conclusiones generales .....	61
Recomendaciones .....	62
Referencias bibliográficas.....	63
Bibliografía .....	66

## INTRODUCCIÓN

La biblioteca surgió en principio con una función conservadora de almacén de información, con el paso del tiempo fue jugando un papel cada vez más importante dentro de la sociedad ya que en varias etapas históricas significó el advenimiento de un notable desarrollo social, económico y cultural. Para la conservación de la información se han empleado diversas técnicas, desde las tablillas utilizadas en la antigua Mesopotamia hasta las fichas y registros bibliográficos.

Según el Diccionario de la Real Academia Española la biblioteca es: *“la institución cuya finalidad consiste en la adquisición, conservación, estudio y exposición de libros y documentos.”* (1)

Con la llegada de la era de la información y el desarrollo de las telecomunicaciones e Internet, las bibliotecas empiezan a tener acceso a documentos que nunca poseerán físicamente debido a esto el volumen de información ha aumentado considerablemente, siendo necesario utilizar las tecnologías de la información para automatizar los procesos que se llevan a cabo en dichas instituciones. En este contexto surgen y se desarrollan los Sistemas Integrados de Gestión Bibliotecaria (SIGB).

La Universidad de las Ciencias Informáticas (UCI) no se ha quedado ajena a esta realidad; por tal razón fue creado el proyecto Gestión Bibliotecaria de la Facultad 1, su principal propósito es automatizar los procesos bibliotecarios de varios centros como la Biblioteca Nacional José Martí o la Biblioteca de la UCI, uno de los productos que este proyecto ofrece es el SIGB bibliotheX cuyo desarrollo está basado en el SIGB Koha y que se distribuye bajo licencia libre.

Sin embargo, este sistema está diseñado principalmente para almacenar las descripciones de los materiales bibliográficos y no bibliográficos que están en formato duro y contiene pocas funcionalidades para administrar la información que está en formato digital permitiendo solamente catalogarla o guardarla en el sistema como un material digital pero no le permite a los usuarios disponer de dicho material.

La producción cada vez mayor de libros electrónicos y otras fuentes de conocimiento en formato digital, evidencia la necesidad de añadir al sistema nuevas funcionalidades que satisfagan las exigencias de los usuarios actuales, en entornos en los cuales la información digital es cada vez más usada.

A partir de lo antes expuesto se plantea el siguiente **problema a resolver**:

¿Cómo facilitar la administración de los recursos digitales en el SIGB bibliotheX para ampliar y mejorar los servicios brindados por la biblioteca?

De acuerdo con la problemática planteada se propone como **objetivo general** el desarrollo del módulo de administración de recursos digitales para el Sistema Integrado de Gestión Bibliotecaria bibliotheX, que permita realizar una mejor gestión de los recursos digitales con que cuentan las bibliotecas.

El **objeto de estudio** de la investigación está enmarcado en los procesos de gestión y organización de los recursos digitales.

Se especifica como **campo de acción** los procesos de gestión y organización de los recursos digitales en la biblioteca.

Durante el transcurso de la investigación se **defiende la idea**: el desarrollo del módulo de administración de recursos digitales para el Sistema Integrado de Gestión Bibliotecaria bibliotheX permitirá realizar una mejor gestión de los recursos digitales.

Del objetivo antes descrito se derivan los siguientes **objetivos específicos**:

- Estudiar herramientas, lenguajes, tecnologías y metodologías utilizadas en el desarrollo de sistemas de administración de recursos digitales.
- Identificar los requisitos funcionales y no funcionales del módulo de administración de recursos digitales para el SIGB bibliotheX.
- Realizar el análisis y diseño del módulo de administración de recursos digitales para el SIGB bibliotheX.
- Verificar mediante pruebas la correcta implementación del módulo de administración de recursos digitales para el SIGB bibliotheX.

Con el propósito de dar cumplimiento a los objetivos se proponen las siguientes **tareas de investigación**:

- Analizar los procesos asociados a la gestión de los recursos digitales para el SIGB bibliotheX.
- Especificar los requisitos del módulo de administración de recursos digitales para el SIGB bibliotheX.
- Modelar los diagramas correspondientes al análisis y diseño del módulo de administración de recursos digitales para el SIGB bibliotheX.

- Implementar la propuesta del módulo de administración de recursos digitales para el SIGB bibliothex.
- Realizar pruebas de caja negra para lograr el correcto funcionamiento del módulo de administración de recursos digitales para el SIGB bibliothex.

Los **métodos teóricos** que son utilizados para darle cumplimiento a las tareas anteriormente expuestas son el **analítico sintético** que permite determinar cuáles serán las tecnologías y elementos necesarios para la implementación del módulo luego de analizar la bibliografía consultada; el **análisis histórico lógico** para el estudio del desarrollo y evolución de las bibliotecas, específicamente en la administración de materiales, incluyendo los digitales, permitiendo comprender los procesos de administración de recursos que tienen lugar en dichas instituciones y el método **modelación** utilizado para la creación de los distintos modelos que realizarán una representación estructurada de la solución del software.

También se utiliza el **método empírico de la observación** el cual posibilitó obtener el conocimiento acerca del comportamiento de los Sistemas Integrados de Gestión Bibliotecaria que ayude al correcto desarrollo del módulo de administración de recursos digitales.

En el presente trabajo se exponen de forma detallada en cuatro capítulos, los elementos de la investigación, entre los que se incluyen, el desarrollo del módulo de administración de recursos digitales, y las pruebas a realizar al mismo que permitan evaluar la calidad del mismo.

**Capítulo 1: Fundamentación teórica.** En este capítulo se presentan los conceptos fundamentales para la comprensión del término administración de recursos digitales y lo referente a las principales características del Sistema Integrado de Gestión Bibliotecaria Koha como sistema base para el desarrollo del módulo de administración de recursos digitales. Además, se describen las principales herramientas y tecnologías a utilizar para el desarrollo del módulo.

**Capítulo 2: Características del sistema.** En este capítulo se describen los objetivos fundamentales de la organización y el flujo actual de los procesos involucrados en el campo de acción. Abarca los elementos relacionados con la automatización de la cual serán objeto dichos procesos. Se describe de forma general la propuesta del sistema y cómo debe funcionar, además de los diferentes requisitos con los que debe contar.

**Capítulo 3: Análisis y diseño del sistema.** Se describe la propuesta del análisis y diseño del sistema y se muestran los diagramas que reflejan dichas descripciones.

**Capítulo 4: Implementación y prueba.** En este capítulo se detalla el modelo de despliegue del sistema y los componentes que lo conforman. Se describen las pruebas de unidades realizadas que permitan validar la solución y se exponen ejemplos de pruebas ejecutadas al sistema que demuestren la aceptabilidad del mismo.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se realiza un estudio sobre el surgimiento, las características y conceptos fundamentales de la administración de recursos. Se muestran algunos Sistemas de Gestión de Activos Digitales utilizados a nivel mundial. Aquí se describen los lenguajes, tecnologías y metodología a utilizar para desarrollar el módulo administración de recursos digitales escogidos para la solución.

### 1.1. Administración de recursos

La **administración** está vinculada al rendimiento y funcionamiento de las organizaciones. El término tiene su origen en el idioma latín: ad-ministrare (“servir”) o ad manus trahere (“manejar” o “gestionar”). Un **recurso**, por otra parte, es un medio de cualquier clase que contribuye a lograr aquello que se pretende.

Las personas, el dinero, la tecnología y hasta el tiempo pueden ser considerados, según el contexto, como recursos susceptibles de ser administrados. La correcta asignación de funciones de cada uno de estos recursos ayudará a que la operatoria de un conjunto sea más eficiente. (2)

La **administración de recursos**, por lo tanto, consiste en el manejo eficiente de estos medios, que pueden ser tanto tangibles como intangibles. El objetivo de la administración de recursos es que éstos permitan la satisfacción de los intereses de la institución. En el caso de esta investigación, que se realiza enfocada en los procesos de administración de recursos digitales en la biblioteca, los intereses o beneficios van enfocados a la institución y a los usuarios. Para la institución los mismos pueden ser: lograr un mayor control y organización de los recursos y contar con una mayor variedad de los mismos y para el usuario los mismos pueden ser: encontrar los recursos con mayor rapidez, tener a su disposición un gran número de recursos a los cuales antes no tenía acceso y poder compartir sus propios recursos con la institución y así se beneficiarán todos, biblioteca y usuarios.

#### 1.1.1. Definiciones de recursos digitales

De acuerdo con una definición formal del ISBD (ER) (1997), se entiende por **recurso digital** “*todo material codificado para ser manipulado por una computadora y consultado de manera directa o por acceso electrónico remoto*”. Para las comunidades académicas los **recursos digitales** más importantes son las revistas y los libros electrónicos, así como las bases de datos en línea, pero se coincide con Codina

(2000) al considerar también de valor informativo otros recursos digitales: los directorios, los motores de búsqueda y los sitios Web de instituciones académicas, corporativas o comerciales, entre otros.

Codina (2001) afirma que los **recursos digitales** poseen propiedades distintas a las de los recursos analógicos y es por eso que se habla de parámetros e indicadores de evaluación particulares.

La primera diferencia entre el documento digital y el analógico es la versatilidad en el uso de formatos, lo que significa que el primero permite incorporar imágenes, sonido, animación o multimedia. Además de lo anterior, el uso del hipertexto da como resultado un recurso que amplía y diversifica de una manera extraordinaria las posibilidades de recuperación de datos. (3)

En los últimos años los recursos digitales también son conocidos como activos digitales los cuales incluyen documentos, correos electrónicos, imágenes, archivos de audio y video y cualquier otro contenido que pueda ser almacenado en un ordenador.

Un sistema de gestión de activos digitales es una solución que combina un programa de catalogación, como es el caso de Cumulus, y una infraestructura informática de cliente-servidor con almacenamiento que juntos ayudan en el proceso de administración de activos digitales.

El término "administración de activos digitales" (DAM) también hace referencia a la descarga, copias de seguridad, clasificación, agrupación, archivado, optimización, mantenimiento, adelgazamiento y exportación de archivos. Y con la evolución y sofisticación de los sistemas DAM hoy en día se permiten la automatización de tareas, flujos de trabajo, anotación y aprobación que pueden ser creados en torno a un activo digital. (4)

Luego de haber analizado las definiciones y demás comentarios brindados sobre los recursos digitales se puede concluir diciendo que los recursos digitales son aquellos materiales que pueden ser procesados por una computadora y que aportan algún valor informativo a la sociedad.

### **1.2. Administración de recursos digitales Open Source**

Debido a la necesidad que ha surgido de controlar y manipular distintos recursos digitales se han creado una serie de sistemas de código abierto utilizados para la administración de recursos digitales. En este epígrafe se mencionan las principales características que describen a los sistemas más importantes de este tipo.



## **DSpace**

DSpace es un software de código abierto diseñado por el Massachusetts Institute of Technology (MIT) y los laboratorios de Hewlett-Packard (HP) para gestionar repositorios de ficheros (textuales, audio, vídeo, etc.), facilitando su depósito, organizándolos en comunidades, asignándoles metadatos y permitiendo su difusión a recolectores o agregadores. Estas características han hecho que, junto con EPrints, sea uno de los programas preferidos por las instituciones académicas para gestionar el repositorio dónde los investigadores depositan sus publicaciones y materiales de búsqueda con objeto de darles una mayor visibilidad.

DSpace se una aplicación cliente/servidor que se gestiona vía web, es decir, que la mayor parte de procesos pueden llevarse a cabo con un navegador estándar como Internet Explorer, Firefox u Opera.

Desde el punto de vista del servidor, DSpace puede instalarse tanto en entornos GNU/Linux como en un servidor Windows, siempre que reúna una serie de prerequisites:

- Entorno de desarrollo Java 1.4 o superior.
- Herramienta de compilación Apache Ante 1.6.2 o superior.
- Sistema de gestión de bases de datos PostgreSQL 7.3 u Oracle 9 o superiores con el juego de caracteres Unicode activados.
- Servidor de aplicaciones Jakarta TomCat 4.x o 5.x.

Por defecto, DSpace está configurado con el esquema de metadatos Dublin Core.

DSpace utiliza una aplicación denominada Checksum Checker que se ejecuta dentro de una tarea programada con objeto de detectar cambios en los ficheros para preservar dichos ficheros. (5)

## **Fedora**

Fedora es el acrónimo de Flexible Extensible Digital Object Repository Architecture (Arquitectura digital de repositorio de objetos digitales flexible y extensible). El repositorio de documentos digitales Fedora requiere de la instalación previa del programa Java en el servidor en el que funcionará; además, Fedora incluye en su paquete de instalación una base de datos que en caso de que los especialistas decidan no

utilizarla, puede sustituirse por otras como Oracle y MySQL. Este software funciona con los sistemas operativos Windows y Unix y sobre servidores Web Apache y Tomcat.

Fedora, es un sistema *open source* gratuito que ofrece a los programadores su código fuente. Se distribuye bajo la Licencia de la Comunidad Educativa y permite que el programa se propague a todos los que lo requieran y que se hagan modificaciones, siempre que se coloquen en forma visible los términos de esta licencia para que otras personas puedan conocer sobre ella.

Para el procesamiento de los documentos, los especialistas son los encargados de asignar los metadatos, los cuales son descritos en el formato Dublin Core. Fedora cuenta con interfaces de presentación distinta para cada una de sus sesiones, una para el procesamiento de los documentos y otra para los usuarios finales. Ellos tendrán la posibilidad de recuperar los contenidos mediante búsquedas en varios índices, previamente declarados por los procesadores o mediante la navegación por las listas de las colecciones.

Este sistema brinda facilidades como la de crear colecciones digitales en varios formatos de documentos, como son: texto, imagen, sonido, etcétera. A pesar de que originalmente el programa es en inglés, es posible configurar por medio de la agregación de aplicaciones adicionales que ofrece el sistema, el programa en varios idiomas.

### **Greenstone**

Greenstone fue creado por la Universidad de Waikato de Nueva Zelanda y se estableció en el año 2000 como parte de su proyecto de la biblioteca digital. Esta herramienta es *open source* y se desarrolló bajo la licencia pública GPL; su distribución es gratuita y en el paquete de instalación se ofrece el código fuente. Esto permite que se pueda adaptar a las necesidades de las instituciones que decidan utilizarlo como repositorio para sus colecciones digitales.

Greenstone se soporta en Windows y Unix. Además, puede desarrollarse perfectamente sobre servidores Web Apache, Personal Web Server (PWS) o Internet Information Server (IIS) de Microsoft y requiere la aplicación Java para su correcto funcionamiento. Las computadoras clientes podrán mostrar las colecciones en ambiente Web mediante sus navegadores, Internet Explorer u otros. Ofrece la posibilidad de instalar la herramienta en varios idiomas, entre los que se encuentra el español.

Greenstone tiene dos posibilidades para instalarse: biblioteca local y biblioteca Web. La biblioteca local es para computadoras autónomas. Tiene incluido un servidor Web, limitado, para poder utilizarlo por otras

computadoras de la red. No necesita ningún tipo de configuración por parte de sus usuarios. La biblioteca Web puede instalarse en cualquier computadora con un servidor Web para distribuir las colecciones con mayor amplitud.

El procesamiento de documentos con Greenstone, que acepta todo tipo de formatos, como pueden ser: texto, imágenes, bases de datos en Isis, etc., se realiza de forma fácil y ágil por medio de la interfaz del bibliotecario. En este ambiente de trabajo se asignan metadatos a los documentos mediante una plantilla, según el modelo de metadatos escogido, entre los que está Dublin Core y un formato propio que ofrece Greenstone. (6)

El estudio de los sistemas antes descritos ha servido de apoyo a la investigación, pues permite conocer cómo trabajan estos y así poder entender mejor como será el funcionamiento del nuevo sistema que se propone en el presente trabajo.

### **1.3. MySQL y administración de recursos digitales**

Dentro de los tipos de cadena que maneja MySQL se pueden distinguir otros dos subtipos, los tipo Text y los tipo BLOB (Binary Large Object), estos últimos son utilizados para guardar información referente a recursos digitales.

La diferencia entre un tipo y otro es el tratamiento que reciben a la hora de realizar ordenamientos y comparaciones. Mientras que el tipo text se ordena sin tener en cuenta las mayúsculas y las minúsculas, el tipo BLOB se ordena teniéndolas en cuenta.

Los tipos BLOB se utilizan para almacenar datos binarios como pueden ser ficheros.

**TinyText y TinyBlob:** Columna con una longitud máxima de 255 caracteres.

**Blob y Text:** un texto con un máximo de 65535 caracteres.

**MediumBlob y MediumText:** un texto con un máximo de 16.777.215 caracteres.

**LongBlob y LongText:** un texto con un máximo de 4.294.967.295 caracteres. Hay que tener en cuenta que debido a los protocolos de comunicación los paquetes pueden tener un máximo de 16 Mb. (7)

## **1.4. Manejo de recursos digitales con el tipo de datos BLOB**

Ante la creciente digitalización de imágenes, planillas, videos, así como todo tipo de documentos, las aplicaciones requieren cada vez mayor organización y mantener guardada esta información que en muchos casos es crítica para la organización.

El tipo de datos Blob permite almacenar esta información en la propia base de datos, aprovechando así los diferentes mecanismos de integridad y control que proveen los Sistemas Gestores de Bases de Datos. Es decir, permite ver esta información como un dato más a manejar. (8)

## **1.5. Sistemas de archivos y almacenamiento de recursos en el módulo de administración de recursos digitales**

El SIGB Koha, sistema en el que está basado el SIGB bibliotheX, internamente está conformado por una serie de carpetas que contienen cada uno de los módulos que trae por defecto el sistema y otros que se han ido modificando y agregando.

Para el desarrollo del módulo de administración de recursos digitales se dispondrá de una carpeta dentro del sistema de archivos de Koha, la cual contendrá, a su vez, una serie de carpetas agrupadas por año.

Dentro de esta carpeta organizada por año, se agruparán los archivos entonces por meses; es decir, estará integrada por una carpeta correspondiente a cada mes, según la fecha en que el técnico procese dicho archivo o recurso.

Por último, se contará con una carpeta que será creada según el día en que se van a guardar los archivos.

En este sistema de archivos se guardarán los archivos con la fecha del sistema en el momento en que el técnico termina de procesar los mismos incluyendo minutos y segundos para evitar duplicaciones. Además esto facilitará una posterior búsqueda por parte del técnico al estar organizado por fechas.

## **1.6. Lenguajes del sistema**

El módulo administración de recursos digitales del SIGB bibliotheX ha sido conformado con un conjunto de lenguajes que incluyen un gran número de características ventajosas a la hora de desarrollar un Sistema Integrado de Gestión de Biblioteca con excepción del Lenguaje de Modelación Unificado (UML) que fue seleccionado con el fin de especificar, visualizar y documentar los artefactos a obtener del módulo administración de recursos digitales.

## **1.6.1. Lenguaje del lado del servidor**

### **Perl 5.10.1**

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI (Common Gateway Interface) para la Web. Perl es un acrónimo de Practical Extracting and Reporting Language, que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

Es un lenguaje de uso libre. Antes estaba muy asociado a la plataforma UNIX, pero en la actualidad está disponible en otros sistemas operativos como Windows.

La estructura completa de Perl deriva de lenguaje C. Perl es un lenguaje imperativo, con variables, expresiones, asignaciones y bloques de códigos delimitados por llaves. Perl tiene una gran potencia en la manipulación de textos debido a que incluye expresiones regulares que facilitan el trabajo con textos. (9)

## **1.6.2. Lenguajes del lado del cliente**

### **HTML**

HTML, Hyper Text Markup Language (Lenguaje de Marcas de Hipertexto) es el lenguaje con el que se definen las páginas Web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página Web. Es un lenguaje de marcación de elementos para la creación de documentos hipertexto, muy fácil de aprender.

HTML es un lenguaje de marca, o sea, una manera de expresar la información de un documento (por ejemplo: información sobre los vínculos del hipertexto y sobre formato) en el documento mismo. Los lenguajes de marca, usan etiquetas que son marcas que se ubican dentro del texto y que brindan información de despliegue. De este modo, el Lenguaje Marca de Hipertexto es una forma específica de usar etiquetas para ofrecer información sobre un documento. (10)

### **JavaScript**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas Web dinámicas. Una página Web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

Los programas JavaScript van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos. (11)

## **CSS**

CSS (Hojas de estilo en cascada o Cascading Style Sheets, en inglés) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación siendo imprescindible para crear páginas Web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, entre otros. (12)

### **1.6.3. Lenguaje de modelado**

## **UML**

UML (Lenguaje de Modelación Unificado) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Actualmente es el más estandarizado por la industria, debido a que ha sido elaborado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otra rama.

Entre sus principales ventajas se encuentran las siguientes:

1. UML se puede usar para modelar distintos tipos de sistemas debido a que ofrece nueve diagramas con los cuales modelar sistemas. A continuación se mencionan los diagramas que utiliza UML para el modelado:
  - Diagramas de Casos de Uso para modelar los procesos.
  - Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
  - Diagramas de Colaboración para modelar interacciones entre objetos.
  - Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
  - Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
  - Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
  - Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
  - Diagramas de Componentes para modelar componentes.
  - Diagramas de Implementación para modelar la distribución del sistema.
2. UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos.

Los creadores de UML se trazaron algunos objetivos, los más específicos son:

- Proporcionar a los usuarios un lenguaje de modelado visual expresivo y utilizable para el desarrollo e intercambio de modelos significativos.
- Proporcionar mecanismos de extensión y especialización.
- Ser independiente del proceso de desarrollo y de los lenguajes de programación.
- Proporcionar una base formal para entender el lenguaje de modelado.
- Fomentar el crecimiento del mercado de las herramientas Orientadas a Objeto (OO).
- Soportar conceptos de desarrollo de alto nivel como pueden ser colaboraciones, frameworks, patterns, y componentes.

- Integrar las mejores prácticas utilizadas hasta el momento. (13)

## 1.7. Herramientas y tecnologías a utilizar

### Quanta Plus 3.5.10

Quanta Plus presenta una rápida e intuitiva interfaz de desarrollo web, busca convertirse en la mejor herramienta de desarrollo web libre y sus funciones no tienen mucho que envidiar a varias aplicaciones comerciales.

Entre sus principales características se encuentran las siguientes:

- Ofrece una interfaz multi-documento, permite el desarrollo web en grupo.
- Ofrece soporte para plugins y debuggers PHP.
- Ofrece además resaltado de sintaxis para HTML, Javascript, CSS y otros.
- También es posible obtener una vista previa de los documentos creados e incluso procesarlos desde un servidor web antes de la vista previa. (14)

### Eclipse SDK 3.3.0

Eclipse es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java.

Se trata de un entorno de desarrollo integrado (IDE) en el que encontrarás todas las herramientas y funciones necesarias para tu trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar.

Cuenta con un editor de texto donde se puede ver el contenido del fichero en el que se está trabajando, una lista de tareas y otros módulos similares. Si bien las funciones de Eclipse son más bien de carácter general, las características del programa se pueden ampliar y mejorar mediante el uso de plug-ins. (15)

### EPIC

EPIC (Editor de Perl y Entorno de Desarrollo Integrado para Eclipse) es un Entorno de Desarrollo Integrado (IDE) de código abierto (con redactor y depuración de código incluidos) basado en la plataforma de Eclipse, compatible con Linux. Para la escritura de programas CGI o proyectos de Perl robustos con



gran cantidad de módulos, EPIC es el más recomendable y extensible IDE disponible actualmente, gracias a la integración con todas las características principales y las convenciones de la Interfaz de Usuario Gráfica de Eclipse.

Con el uso de EPIC se obtienen varias ventajas, las principales se muestran a continuación:

- Fácil instalación en la plataforma Eclipse.
  - Entre las principales funciones soportadas se encuentran el resaltado de sintaxis, autocompletado de funciones y métodos, soporte para documentación de Perl, plantillas de código, expresiones regulares.
  - Herramientas de documentación para funciones de Perl incorporadas.
  - Los errores de sintaxis son detectados y explicados mientras se edita un archivo en tiempo real.
- (16)

### **Visual Paradigm 6.4**

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (17)

### **MySQL 5.1.49**

MySQL es uno de los gestores de base de datos relacionales más usados actualmente para el desarrollo de aplicaciones web. En su uso destacan empresas como Google y Yahoo. Por otra parte, está liberado bajo la licencia GPL de GNU aunque también ofrece una versión comercial que posibilita la integración de este gestor a un software propietario. Además, es robusto, rápido y fácil de utilizar. A continuación se ilustrarán algunas de las características que hacen de MySQL un potente gestor de bases de datos:

- Flexibilidad: La flexibilidad de plataforma es una característica clásica de MySQL, soportando distintas versiones de Linux, UNIX y Windows y permite una personalización completa para aquellos que deseen añadir características al servidor.

- Robusto soporte transaccional: MySQL ofrece uno de los motores de bases de datos transaccionales más potentes del mercado. Las características incluyen un soporte completo de ACID (atómica, consistente, aislada, duradera), bloqueo a nivel de filas, posibilidad de transacciones distribuidas, y soporte de transacciones con múltiples versiones donde los lectores no bloquean a los escritores y viceversa. También se asegura una integridad completa de los datos mediante integridad referencial, niveles de aislamiento de transacciones especializados, y detección de deadlocks.
- Fortalezas en Web: MySQL es el estándar de-facto para sitios web de gran tráfico por su motor de consultas de alto rendimiento, su posibilidad de insertar datos a gran velocidad, y un buen soporte para funciones web especializadas, como las búsquedas fulltext. Otras características como las tablas en memoria, índices B-tree y hash, y tablas comprimidas hasta un 80% hacen de MySQL una buena opción para aplicaciones web.
- Fuerte protección de datos: MySQL ofrece características de seguridad que aseguran una protección absoluta de los datos. En cuanto a autenticación, MySQL ofrece potentes mecanismos para asegurar que sólo los usuarios autorizados tienen acceso al servidor. También se ofrece soporte SSH y SSL para asegurar conexiones seguras. Existe una estructura de privilegios que permite que los usuarios sólo puedan acceder a los datos que se les permite, así como potentes funciones de cifrado y descifrado para asegurarse de que los datos están protegidos. Finalmente, se ofrecen utilidades de backup y recuperación por parte de MySQL y terceros, que permiten copias completas, tanto lógicas como físicas, así como recuperación point-in-time. (18)

### **Subversion**

Subversion es un sistema de control de versiones libre y de código fuente abierto que maneja ficheros y directorios a través del tiempo ya que contiene un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. A cierto nivel, la capacidad para que varias personas puedan

modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. (19)

Subversion proporciona:

- **Versionado de directorios:** Implementa un sistema de ficheros versionado “virtual” que sigue los cambios sobre árboles de directorios completos a través del tiempo.
- **Verdadero historial de versiones:** Se puede añadir, borrar, copiar, y renombrar ficheros y directorios. Cada fichero nuevo añadido comienza con un historial nuevo, limpio y completamente suyo.
- **Elección de las capas de red:** Tiene una noción abstracta del acceso al repositorio, facilitando a las personas implementar nuevos mecanismos de red. Puede conectarse al servidor HTTP Apache como un módulo de extensión. También tiene disponible un servidor de Subversion independiente, y más ligero. (19)

**RapidSVN** es un cliente gráfico que permite manipular los repositorios de Subversion. Además es una de las alternativas más conocidas para los sistemas GNU/Linux, muy intuitivo y fácil de utilizar.(20)

### **Apache 2.2.16**

Apache es uno de los servidores web más utilizados actualmente, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

Sus principales características son:

- Es multiplataforma.
- Apache es una tecnología gratuita de código fuente abierto.
- Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y a las necesidades del usuario que lo utiliza.
- Apache utiliza varios lenguajes como: Perl, PHP y otros lenguajes de script. También trabaja con Java y páginas jsp. Teniendo todo el soporte que se necesita para tener páginas dinámicas.

- Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor. (21)

### **CGI**

El CGI por sus siglas en inglés 'Common Gateway Interface' es de las primeras formas de programación Web dinámica. Es un método para la transmisión de información hacia un compilador instalado en el servidor. Su función principal es la de añadir una mayor interacción a los documentos Web que por medio del HTML se presentan de forma estática.

El CGI es utilizado comúnmente para contadores, bases de datos, motores de búsqueda, formularios, generadores de e-mail automático, foros de discusión, chats, comercio electrónico, rotadores y mapas de imágenes, juegos en línea y otros.

Esta tecnología tiene la ventaja de correr en el servidor cuando el usuario lo solicita por lo que es dependiente del servidor y no de la computadora del usuario.

Los programas que maneja el CGI pueden estar compilados en diferentes lenguajes de programación. El más popular para el desarrollo de contenidos Web es el lenguaje Perl de distribución gratuita, aunque también podemos mencionar: C, C++ y Java. (22)

### **1.8. Metodologías de desarrollo de software**

El desarrollo de software es un área que ha ido en aumento con el transcurso de los años, debido a esto el proceso de desarrollo en sí se hace cada vez más complejo, de ahí que haya surgido la necesidad de crear una especie de guía para llevar a cabo esta tarea. Por esta razón surgieron las metodologías de desarrollo de software como el conjunto de procedimientos, técnicas y herramientas que ayudan a los desarrolladores a realizar un nuevo software.

Con el paso de los años fueron surgiendo un cúmulo de metodologías, unas enfocadas en una fuerte planificación durante todo el proceso de software y otras pensadas para proyectos donde el entorno es

volátil y los requisitos muy cambiantes. Ejemplos de estas son: RUP (Rational Unified Process), XP (Extreme Programming), Scrum y SXP.

A continuación se realizará una breve descripción de cada una de estas metodologías:

### **XP**

Es una metodología ágil creada por Kent Beck, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y cambiantes. (23)

### **Scrum**

Es una metodología desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Está basado en los principios de inspección continua, adaptación, auto-gestión e innovación. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de treinta días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto. Éstas son las verdaderas protagonistas. (24)

### **SXP**

Es un híbrido cubano que tiene como base las metodologías SCRUM y XP, que permiten actualizar los procesos de desarrollo de software para el mejoramiento de su producción. Consta de 4 fases: Planificación-Definición, Desarrollo, Entrega y Mantenimiento, cada una desglosada en flujos de trabajo y actividades que generan artefactos. Esta metodología está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, y permite además seguir de forma clara el avance del equipo de desarrollo por parte del cliente, de forma que los jefes pueden ver día a día cómo progresa el trabajo. (24)

### **RUP**

La metodología RUP (Proceso Unificado de Rational) es un proceso de desarrollo de software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Tiene como características esenciales que está dirigido por los Casos de Uso, está centrado en la arquitectura y es iterativo e incremental. (25)

RUP divide el proceso en cuatro fases: Inicio, Elaboración, Construcción y Transición, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor.

La metodología RUP se ha convertido rápidamente en el estándar de facto para el proceso de desarrollo en la industria de software. A diferencia de otras metodologías comerciales, la plataforma RUP hace que el proceso sea práctico con bases de conocimiento y guías para ayudar en el despegue de la planificación del proyecto, integrar rápidamente a los miembros del equipo y poner en acción el proceso personalizado. RUP proporciona un Framework de proceso configurable que permite seleccionar e implantar los componentes específicos de procesos necesarios para proporcionar un proceso consistente y customizado para cada equipo y proyecto. (26)

Teniendo en cuenta las características mencionadas anteriormente, se propone a RUP como la metodología indicada para modelar el desarrollo del módulo administración de recursos digitales y de esta forma darle solución al problema propuesto.

### **Conclusiones del capítulo**

En este capítulo se le da cumplimiento al objetivo planteado al principio del mismo debido a que se pudieron abordar y analizar las definiciones acerca de la administración de recursos digitales. Por otra parte, se determinaron las principales herramientas, tecnologías y metodología de desarrollo de software más propicias para el buen desarrollo del módulo de acuerdo con la política del proyecto Gestión Bibliotecaria que utiliza a RUP como la metodología de desarrollo y debido a que el sistema bibliotheX está basado en el SIGB Koha, se hizo necesario el empleo de las herramientas, lenguajes y demás tecnologías descritas anteriormente para el desarrollo del módulo de administración de recursos digitales, garantizando la correcta integración del mismo al SIGB bibliotheX y evitar posibles incompatibilidades que podrían surgir si no se utiliza el software requerido para dicha implementación.

## **CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA**

En el presente capítulo se realiza una propuesta del subsistema que se obtendrá con la posterior implementación del mismo. Se definen los requisitos funcionales y no funcionales necesarios para el desarrollo del módulo, además de los diferentes diagramas que corresponden a los flujos de trabajo de Levantamiento de Requerimientos y Modelación del Negocio.

### **2.1. Análisis crítico de los procesos de gestión y organización de los recursos digitales**

Actualmente el SIGB bibliotheX no cuenta con un sistema automatizado que se encargue de administrar y manipular los recursos digitales con que cuenta el mismo. Esto trae como consecuencia que no se realice una correcta administración de los recursos digitales en la aplicación. Además, al no existir un sistema que se dedique a estas funciones solamente se tiene control del material físico que posee la biblioteca no siendo así en el caso de los recursos digitales con que cuenta la biblioteca debido a que no están creados los registros bibliográficos o al menos no con toda la información asociada a cada uno de los recursos. Por otra parte, el usuario tampoco puede acceder a estos recursos digitales, pudiéndose en muchos casos evitar que el mismo tenga que acudir a la biblioteca en busca de un material que puede estar en formato digital ya que el sistema no le brinda la opción de buscar documentos en dicho formato.

Por lo antes expuesto se considera necesario crear un módulo que contenga un conjunto de funcionalidades que les permita tanto a los trabajadores de la biblioteca como a los usuarios realizar una serie de acciones que contribuyan a la mejor gestión y manipulación de los recursos digitales.

### **2.2. Objeto de automatización**

#### **2.2.1. Descripción de los procesos que serán objeto de automatización**

Serán objeto de automatización los procesos de gestión y organización de los recursos digitales en el SIGB bibliotheX. Como principales procesos seleccionados a automatizar se plantean los siguientes:

Controlar los recursos digitales: Este control incluye la organización de dichos recursos por año y luego por tipo de material, el tipo de material puede ser: libros, archivos de computadora, entre otros.

Visualizar los recursos digitales: Esta visualización permitirá acceder a los recursos, por ejemplo si es un documento el sistema deberá visualizar el contenido del documento.

Controlar la descarga y carga de recursos digitales: La descarga se hará de la forma habitual mostrando un link que contenga la ubicación del recurso y que el usuario pueda tenerlo en su poder siempre y cuando el recurso en cuestión tenga los permisos para poder ser descargados.

En el caso de la carga de algún recurso se le permitirá al usuario cargar uno o varios recursos que este considere de utilidad para la biblioteca y demás usuarios, aunque estos recursos que el usuario cargue tenga serán enviados primeramente a un directorio temporal y luego ser catalogado por el Técnico Bibliotecario para su correcta clasificación y evitar que existan duplicaciones o errores en la descripción de los recursos.

### **2.2.2. Descripción de los sistemas automatizados que existen en la entidad y que están vinculados con el campo de acción del proyecto**

La Biblioteca de la UCI, cuenta con un sistema que se encarga de almacenar y clasificar los recursos digitales que llegan a la institución. Este sistema también almacena dichos recursos mediante la utilización de un sistema de archivos al cual acceden los técnicos encargados de la catalogación de los mismos pero él mismo utilizaba el software Micro CDS/ISIS, (Computerized Documentation System - Integrated Set for Information System), también conocido como MicroISIS, para la creación de registros bibliográficos. El mismo incluye la definición de bases de datos utilizando diferentes estructuras de formatos (CEPAL, MARC, etc.) Se decidió utilizar el formato MARC 21, conjunto de estándares para el manejo, recuperación e intercambio de información porque mantiene la integridad de los datos ante posibles cambios de sistema y permite catalogar cualquier tipo de recurso. Por tal razón, al proyecto Gestión Bibliotecaria se le encomendó la tarea de migrar este sistema que se empleaba en la institución al SIGB Koha. Por otra parte, el SIGB bibliotheX cuenta con un sistema automatizado basado en el SIGB Koha pero este sistema no contiene un subsistema que se encargue de realizar la gestión de los recursos digitales. Este proceso se está llevando a cabo actualmente.

### **2.3. Información que se maneja**

Dentro de la información que se maneja se encuentran los materiales de entrada. Para este subsistema son: los libros, archivos de computadoras, materiales audiovisuales, materiales mixtos, música y mapas. Otro material de entrada lo constituye el registro bibliográfico asociado a cada uno de estos recursos mencionados anteriormente. Los materiales de salida serían los recursos digitales que se obtendrán al



interactuar con el subsistema, así como el registro bibliográfico asociado al mismo una vez culminada la catalogación.

## **2.4. Propuesta de sistema**

### **2.4.1. Descripción general de la propuesta de sistema, como debe funcionar**

Con el desarrollo del subsistema de administración de recursos digitales para el SIGB bibliotheX se facilitará el trabajo de los bibliotecólogos al colaborar con la mejor organización de los recursos con que cuenta el sistema y facilitará a los usuarios la búsqueda de diversos tipos de recursos digitales y la posterior manipulación de los mismos. Este módulo de administración se centraría primeramente en tener almacenados y clasificados los recursos digitales por año y luego poder acceder a los mismos o permitirle al usuario cargar algún material que no se encuentre en la biblioteca y que resulte interesante aunque este deba pasar por el proceso de catalogación previamente para verificar su autenticidad y calidad. Además, les permitiría a los trabajadores de la biblioteca crear o modificar los registros bibliográficos pertenecientes a los recursos digitales que maneja el sistema añadiéndole a los mismos la url del recurso a la cual el usuario podrá acceder posteriormente para disponer del archivo.

## **2.5. Modelamiento del Negocio**

El modelamiento del negocio es una de las acciones más importantes que se realizan durante el desarrollo de un sistema ya que permite entre otras cosas comprender la dinámica de la institución en la que se va a implantar la aplicación. En el caso específico del módulo de administración de recursos digitales se decidió que no es necesario un modelo completo del negocio y se realizará lo que se conoce como un modelo del dominio que brinda una vista más ajustada a la realidad del cliente y su entorno, enfocada a lo que el sistema debe hacer.

### **2.5.1. Modelo del Dominio**

Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. (27)

A continuación se muestra y describe el modelo de dominio elaborado para la mejor comprensión del sistema a desarrollar:

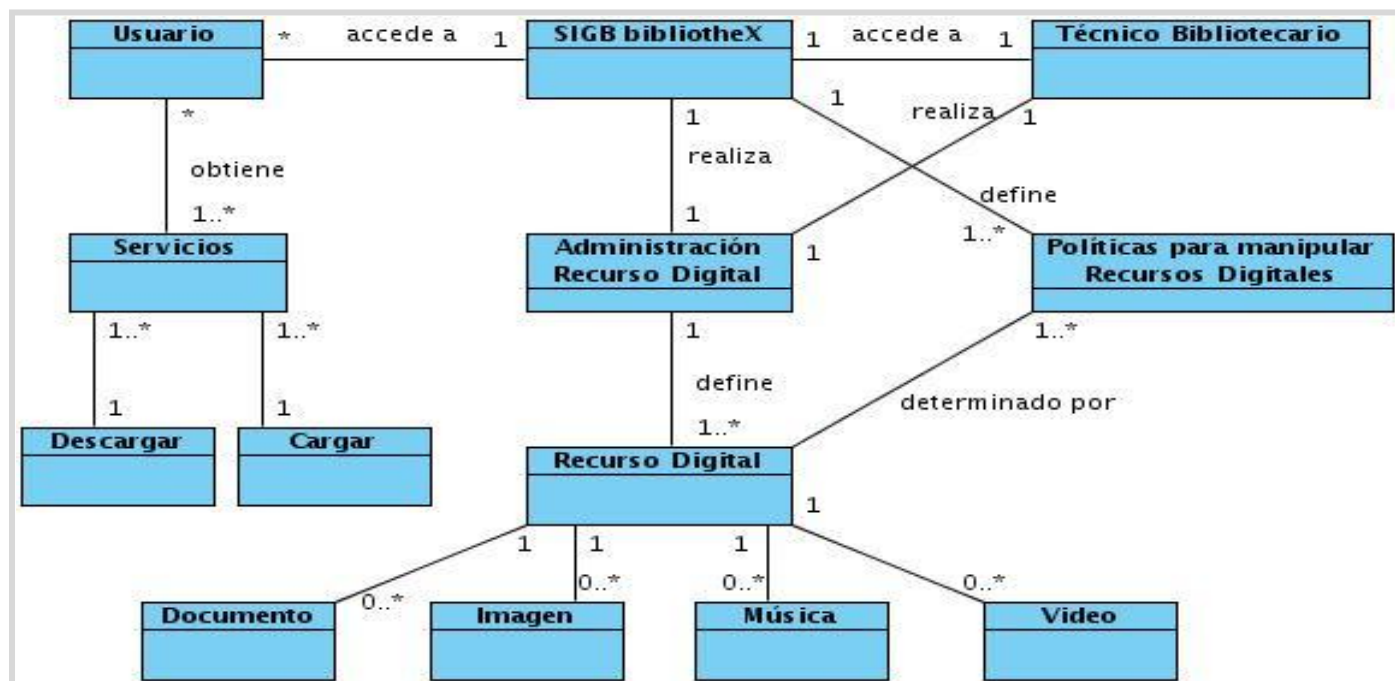


Figura 1. Modelo del Dominio.

## 2.5.2. Descripción de términos del dominio

**SIGB bibliothex:** Sistema al que acceden los usuarios para obtener los diversos servicios.

**Usuario:** Persona que obtiene servicios de Descargar y Cargar recursos digitales.

**Técnico Bibliotecario:** Persona que brinda servicios a los usuarios.

**Servicios:** Opciones o funcionalidades que brinda el sistema al usuario.

**Administración Recurso Digital:** El sistema se encarga de realizar la gestión de los recursos digitales siempre que exista.

**Políticas para manipular recursos digitales:** El sistema define las políticas que deben cumplir los recursos digitales para poder ser manipulados por el usuario.

**Descargar:** Servicio que brinda el sistema mediante el cual el usuario podrá descargar recursos digitales de acuerdo a lo establecido por la biblioteca.

**Cargar:** Servicio que brinda el sistema mediante el cual el usuario podrá cargar los recursos que sean de utilidad para los demás usuarios y la biblioteca en sentido general.

**Recurso digital:** Todo aquel archivo en formato digital que posea la biblioteca o el usuario que esté disponible en el sistema. Podrá ser de cuatro tipos: documentos, imágenes, música y videos.

**Documentos:** Podrán manipularse documentos con extensiones doc. o pdf.

**Imágenes:** Serán de formato jpeg, gif.

**Música:** Será en formato mp3.

**Videos:** Se encontrarán vídeos en cualquier formato siempre y cuando tengan la calidad requerida.

## **2.6. Especificación de los requisitos de software**

La especificación de requisitos captura los requisitos de software para el sistema completo o una porción del sistema.

### **2.6.1. Requisitos funcionales**

Los requisitos funcionales de un sistema describen la funcionalidad o los servicios que se espera que éste provea. Estos dependen del tipo de software y del sistema que se desarrolle y de los posibles usuarios del software. En fin, los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. A continuación se listarán los requisitos funcionales que empleará el sistema: (28)

RF 1: Autenticar usuario.

Descripción: El usuario debe autenticarse con su identificador y contraseña antes de acceder al servicio.

RF 2: Cargar recursos digitales.

Descripción: El sistema le brindará al usuario la opción cargar uno o varios recursos.

RF 3: Verificar que los recursos a cargar estén en los formatos adecuados.

Descripción: Una vez que el usuario carga un recurso, el sistema debe verificar que el formato del mismo sea uno de los establecidos por el sistema.

RF 4: Indexar recurso.

Descripción: Al usuario cargar un recurso, el sistema enviará el mismo a un directorio temporal, si el recurso cargado por el usuario no tiene un registro bibliográfico asociado, el Técnico se encargará de realizarle el proceso de catalogación, para lo cual el técnico deberá crear un nuevo registro bibliográfico

para dicho recurso. En caso contrario, el técnico podrá modificar el registro bibliográfico para agregarle al mismo la nueva dirección de donde el usuario obtendrá el recurso digital.

RF 5: Permitir descarga de recursos digitales.

Descripción: El sistema permitirá al usuario descargar los recursos digitales que dispone la biblioteca según los permisos que la misma establezca.

RF 6: Realizar Búsqueda.

Descripción: El usuario podrá realizar una búsqueda para encontrar los recursos de su interés.

RF 7: Restringir el tamaño de los recursos.

Descripción: El técnico tendrá la opción de restringir el tamaño de los recursos que el usuario vaya a cargar, de acuerdo a lo que la biblioteca indica al respecto.

### **2.6.2. Requisitos no funcionales**

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. (28)

A continuación se muestran las descripciones de los requisitos no funcionales definidos para este sistema:

#### **Seguridad**

Para acceder tanto a la interfaz administrativa como a la interfaz del OPAC, los usuarios y el Técnico Bibliotecario deberán autenticarse mediante el uso de su identificador y contraseña.

#### **Usabilidad**

El sistema deberá ser fácil de usar por los usuarios. La aplicación deberá poder ser usada por cualquier persona que posea conocimientos básicos de computación.

#### **Portabilidad**

Esta se refiere a la capacidad que deberá tener el sistema de poder ejecutarse sobre diferentes plataformas como Linux y Windows.

#### **Interfaz**

La interfaz deberá ser sencilla, amigable y de fácil uso para los usuarios finales. Los colores y el diseño de la misma deberán estar acorde con el del SIGB biblioteX.

## 2.7. Modelado del Sistema

En el presente epígrafe se realizará la modelación del sistema, pudiendo definirse los actores que interactuarán con el sistema, los casos de usos que lo conformarán, así como las descripciones de los mismos.

### 2.7.1. Definición de los actores del sistema

A continuación se listan los actores que intervienen en el sistema:

Actor del sistema	Justificación
Técnico Bibliotecario	Es el encargado de controlar la adjunción de recursos digitales y catalogarlos; además permitirá la descarga de los recursos digitales.
Usuario	Recibe las opciones de descargar o adjuntar algún recurso digital.
Usuario del sistema	Grupo de usuarios que accederán al sistema para recibir y realizar diferentes actividades de acuerdo con el rol que designa el sistema con previa autenticación.

Tabla 1. Actores del sistema.

### 2.7.2. Listado de casos de uso

CU-1. Autenticar usuario.

CU-2. Cargar archivos.

CU-3. Catalogar recursos sin registro asociado.

CU-4. Indexar archivos.

CU-5. Configurar permisos de descarga.

2.7.3. Diagrama de casos de uso

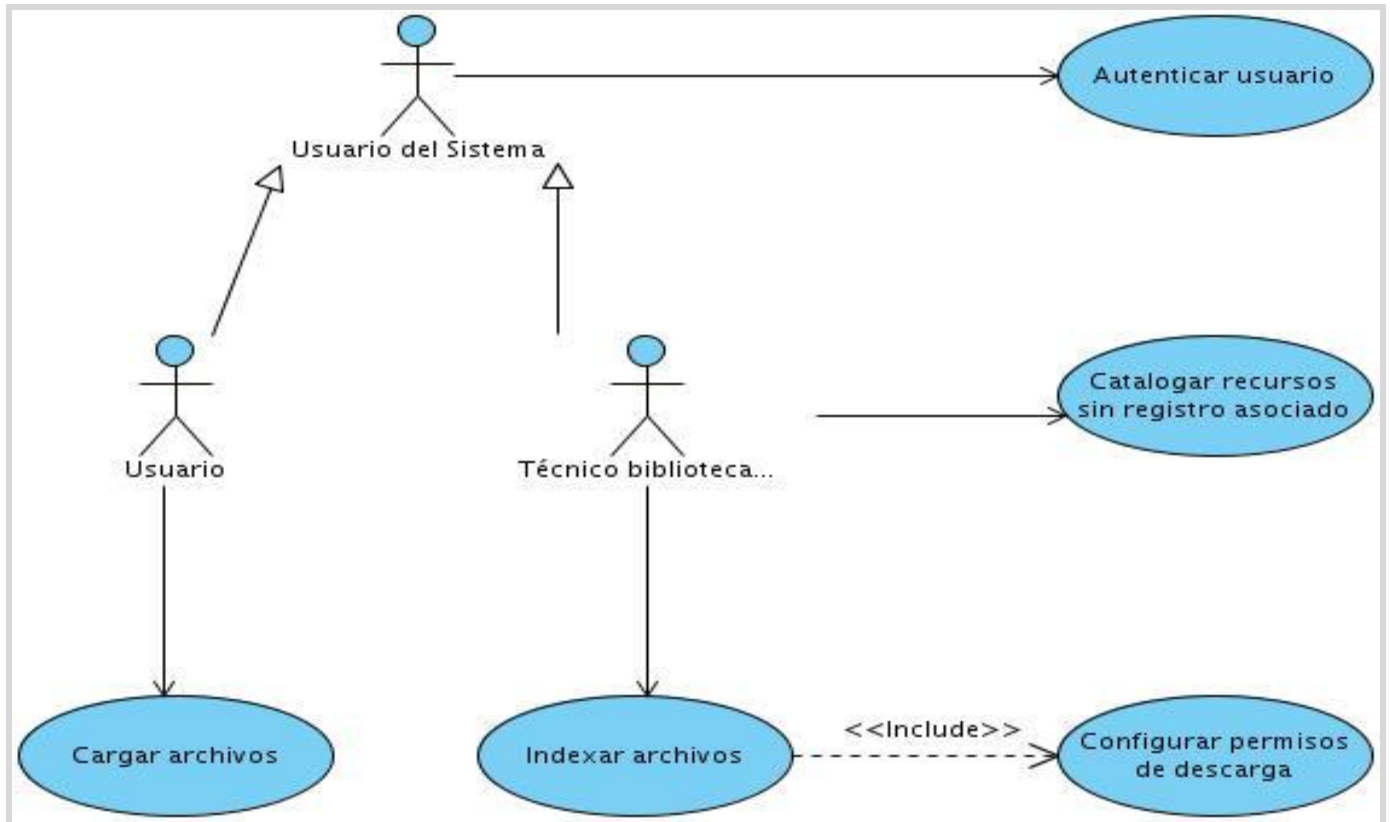


Figura 2. Diagrama de Casos de Uso del Sistema.

## 2.7.4. Casos de uso expandidos

Caso de uso	
CU-1	Autenticar usuario.
<b>Propósito</b>	Permitir a los usuarios la correcta autenticación para acceder al sistema según las restricciones del mismo.
<b>Actores</b>	Usuario del Sistema
<b>Resumen:</b>	El caso de uso inicia con la introducción del identificador y la contraseña por parte del Usuario. El caso de uso finaliza cuando el sistema verifica la autenticidad del Usuario, negando la entrada en caso de no introducirse datos válidos o mostrando las funciones y servicios de acuerdo a los permisos asignados al usuario.
<b>Precondiciones</b>	
<b>Referencias</b>	RF1
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Usuario introduce usuario y contraseña para acceder al sistema.	2. El sistema verifica que los datos introducidos coinciden con algún Usuario registrado.
	3. El sistema muestra todas las funciones y servicios de acuerdo a los permisos asignados al usuario. Termina el caso de uso.
<b>FLUJO ALTERNO</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	2.1. El sistema muestra mensaje de error, notificando que el usuario o la contraseña no son válidos. Muestra el formulario para autenticarse nuevamente.

<b>Caso de uso</b>	
CU-2	Cargar archivos.
<b>Propósito</b>	Los Usuarios tendrán la posibilidad de cargar archivos según las restricciones del sistema.
<b>Actores</b>	Usuario
<b>Resumen:</b>	El caso de uso inicia cuando el Usuario desea cargar algún archivo al sistema y finaliza cuando el sistema guarda el o los archivos cargados.
<b>Precondiciones</b>	El Usuario debe haberse autenticado correctamente.
<b>Referencias</b>	RF1, RF2, RF3, RF7.
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Usuario selecciona la opción de Cargar archivo.	2. El sistema le muestra un formulario donde el Usuario puede cargar un archivo. Permite al Usuario las opciones de Cargar o Cancelar.
3. El Usuario decide Cargar y llena un formulario con algunos datos que describan el archivo que va a cargar como son el Título y el Autor.	4. El sistema valida que este haya seleccionado un archivo. También valida que el archivo esté en alguno de los formatos establecidos por la biblioteca y valida que el tamaño del mismo cumpla con las restricciones del sistema. Además valida que el Usuario no haya dejado campos vacíos.
	5. El sistema guarda el archivo, se le notifica al Usuario que la carga del archivo se realizó exitosamente y finaliza el caso de uso.
<b>FLUJO ALTERNO</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>



3.1. El Usuario selecciona la opción Cancelar y finaliza el caso de uso.	
	<p>4.1. Si el Usuario no selecciona algún archivo, el sistema le muestra una notificación para que seleccione al menos un archivo y se remite al usuario a la acción 2.</p> <p>4.2. El sistema muestra una notificación que informa que el archivo en cuestión no tiene el formato establecido por la biblioteca y se remite al usuario a la acción 2.</p> <p>4.3. El sistema muestra una notificación que informa que el archivo excede el tamaño permitido por la biblioteca y se remite al usuario a la acción 2.</p> <p>4.4. El sistema muestra una notificación que informa que existen campos vacíos y se remite al usuario a la acción 2.</p>

<b>Caso de uso</b>	
CU-3	Catalogar recursos sin registro asociado.
<b>Propósito</b>	Catalogar los recursos digitales que llegan al sistema.
<b>Actores</b>	Técnico Bibliotecario
<b>Resumen:</b>	El caso de uso inicia cuando el Técnico recibe los recursos digitales y le realiza el proceso de catalogación al mismo. Termina una vez que el recurso tiene un registro bibliográfico asignado.
<b>Precondiciones</b>	El Técnico debe haberse autenticado correctamente. Búsqueda de registro bibliográfico realizada.
<b>Referencias</b>	RF1, RF4, RF6.
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Técnico accede a crear un nuevo registro.	2. El sistema muestra un formulario donde el técnico puede comprobar la existencia del material, por determinados criterios de búsqueda.
3. El Técnico selecciona los criterios de búsqueda, introduce los datos correspondientes y busca el registro.	4. El sistema muestra un mensaje informándole que el material no existe.
5. El Técnico selecciona crear un nuevo registro.	6. El sistema muestra un formulario con los campos necesarios para el registro.
	9. El sistema verifica que los datos insertados sean correctos.
	10. El sistema guarda el o los recursos asociados al registro creado.
	11. El sistema guarda el registro del material.

	12. El sistema emite un mensaje informando que el registro fue insertado satisfactoriamente, finalizando el caso de uso.
<b>FLUJO ALTERNO</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	4.1. El sistema emite un mensaje notificando que existe el registro del material, finalizando así el caso de uso.
	8.1. El sistema informa que los campos no están llenos correctamente.
	9.1. El sistema emite un mensaje para que se verifiquen los datos.

<b>Caso de uso</b>	
CU-4	Indexar archivo.
<b>Propósito</b>	Permitir al Técnico indexar un archivo a un registro bibliográfico, es decir, asociar un archivo digital al registro.
<b>Actores</b>	Técnico bibliotecario
<b>Resumen:</b>	El caso de uso inicia cuando el Técnico desea indexar algún archivo a un registro bibliográfico. Termina una vez que el archivo sea indexado al registro correspondiente.
<b>Precondiciones</b>	El Técnico debe haberse autenticado correctamente. Registro bibliográfico creado. Búsqueda de registro bibliográfico realizada.
<b>Referencias</b>	RF1, RF4, RF6.
<b>Sección “Buscar en Sistema Local de Ficheros”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

1. El Técnico selecciona la opción de Adjuntar archivo.	2. El sistema le muestra una ventana donde el Técnico puede adjuntar un archivo al registro.
4. El Técnico selecciona el o los archivos que desea adjuntar al registro en el sistema local de su computadora.	3. El sistema muestra una ventana donde el Técnico puede realizar una búsqueda y elegir los archivos a adjuntar guardados previamente en el sistema local de archivos de su computadora.
5. El Técnico presiona el botón Adjuntar.	6. El sistema guarda los archivos en el repositorio con que dispone el mismo para el control y almacenaje de los ficheros. Finaliza el caso de uso.
<b>Sección “Buscar en Repositorio Temporal”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
2. Una vez que el Técnico seleccionó la opción Adjuntar archivo elige alguno de los archivos que se muestra en el listado mencionado.	1. El sistema muestra una ventana donde aparecen los datos de los diez primeros archivos cargados previamente por los usuarios en el repositorio temporal y el Técnico puede escoger de aquí los que desee en caso de que estos se encuentren en este.
3. El Técnico presiona el botón Adjuntar.	3. El sistema guarda el archivo y lo adjunta al registro. Finaliza el caso de uso.
<b>FLUJO ALTERNO</b>	
2.1. El Técnico realiza la búsqueda de los archivos que desea en el repositorio, escoge los archivos que necesita. Se repite la Acción 3.	1.1. El sistema ofrece la posibilidad de realizar una búsqueda de los archivos en el repositorio temporal mediante los parámetros Título, Autor y Usuario.

Caso de uso	
CU-5	Configurar permisos de descarga
<b>Propósito</b>	Permitir a los usuarios la descarga de recursos digitales según las restricciones del sistema.
<b>Actores</b>	Técnico Bibliotecario
<b>Resumen:</b>	El caso de uso inicia cuando el Técnico determina los permisos que va a tener cada recurso digital para ser descargado o ser de sólo lectura según lo que establece la biblioteca. Termina una vez que el Técnico haya otorgado los permisos correspondientes.
<b>Precondiciones</b>	El Técnico debe haberse autenticado correctamente. Búsqueda de registro bibliográfico realizada.
<b>Referencias</b>	RF1, RF6, RF7.
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Técnico selecciona un registro al que está asociado un recurso digital.	2. El sistema mostrará el registro bibliográfico. Permite establecer los permisos para que el Usuario pueda descargar el recurso.
3. El Técnico selecciona la opción que la biblioteca establezca.	4. El sistema guardará los cambios, notificará que los cambios se han realizado correctamente y finaliza el caso de uso.
<b>FLUJO ALTERNO</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

### Conclusiones del capítulo

En este capítulo se realizó la propuesta del sistema a implementar, se definió el modelo del dominio y se construyó el diagrama de casos de uso del sistema para comprender la dinámica del sistema. También se enumeraron los requisitos funcionales y no funcionales con los que debe contar la aplicación y finalmente se llegó a la descripción detallada de cada caso de uso.

## **CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA**

En el presente capítulo se abordará el flujo de trabajo de Análisis y Diseño, para su modelado se emplean los diagramas de clases del análisis, de interacción y el desarrollo de otras actividades que facilitarán el entendimiento del sistema. También se modelarán los diagramas de clase del diseño con estereotipos web para poder apreciar la relación que existe entre las diferentes clases que conforman el sistema y la base de datos y por último se definirá la arquitectura del mismo.

### **3.1 Análisis**

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales.

Para realizar el modelo del análisis se emplean las siguientes clases:

**Clases Interfaz o CI\_< Nombre de la clase >:** Estas clases permiten modelar la interacción entre el sistema y sus actores, es decir, ventanas, sistemas externos, etc.

**Clases de Control o CC\_< Nombre de la clase >:** Las clases de control coordinan el trabajo de uno o unos pocos casos de uso, por lo que definen el flujo de control y las transacciones dentro de un caso de uso delegando el trabajo a otros objetos.

**Clases Entidad o CE\_< Nombre de la clase >:** Estas clases modelan información que posee una larga vida y que a menudo es persistente. (29)

A continuación se muestran los diagramas de clases de análisis definidos para cada caso de uso:

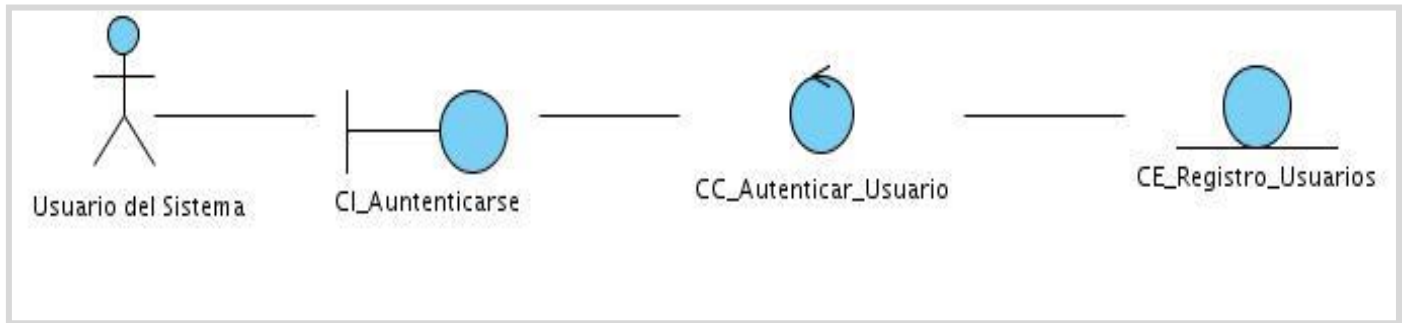


Figura 3. Diagrama de Clase de Análisis del CU Autenticar usuario.

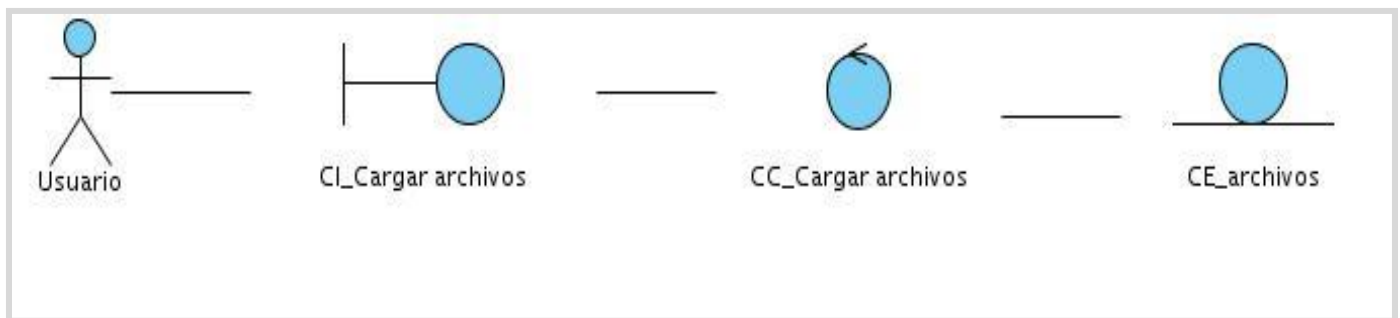


Figura 4. Diagrama de Clase de Análisis del CU Cargar archivos.

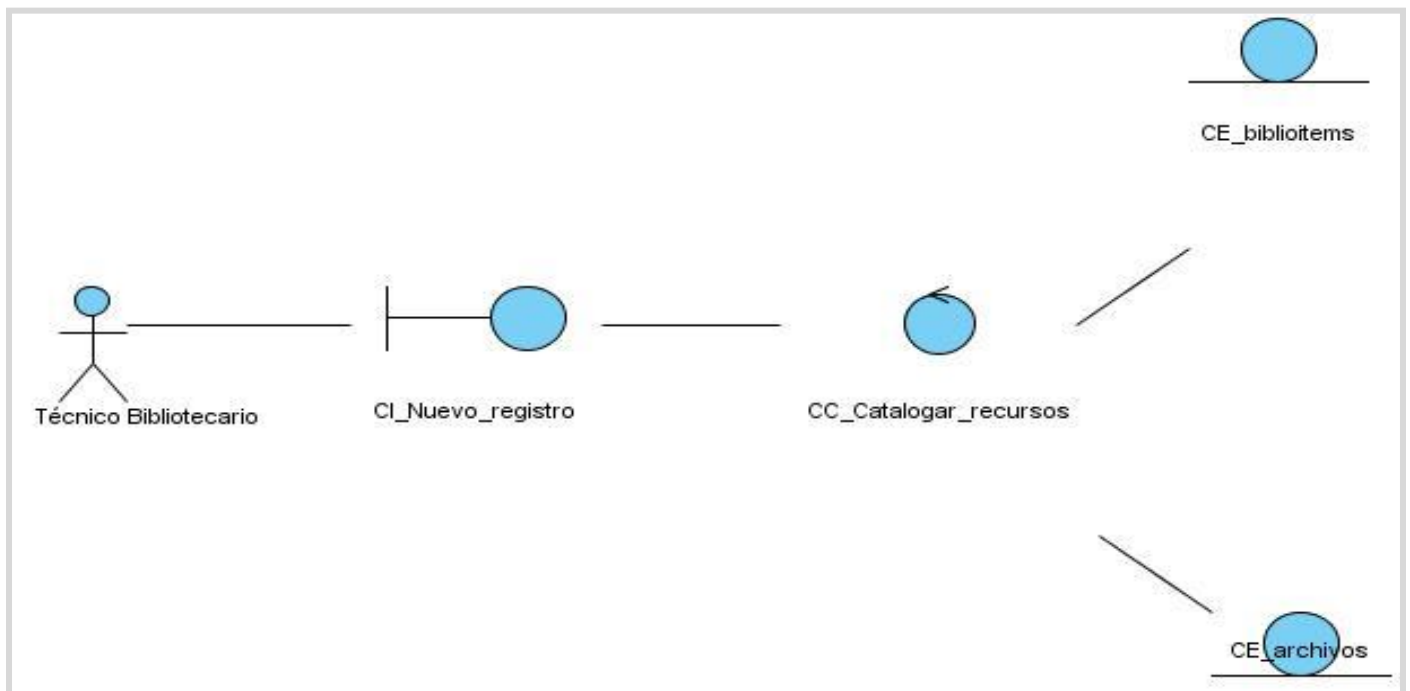


Figura 5. Diagrama de Clase de Análisis del CU Catalogar recursos sin registro asociado.

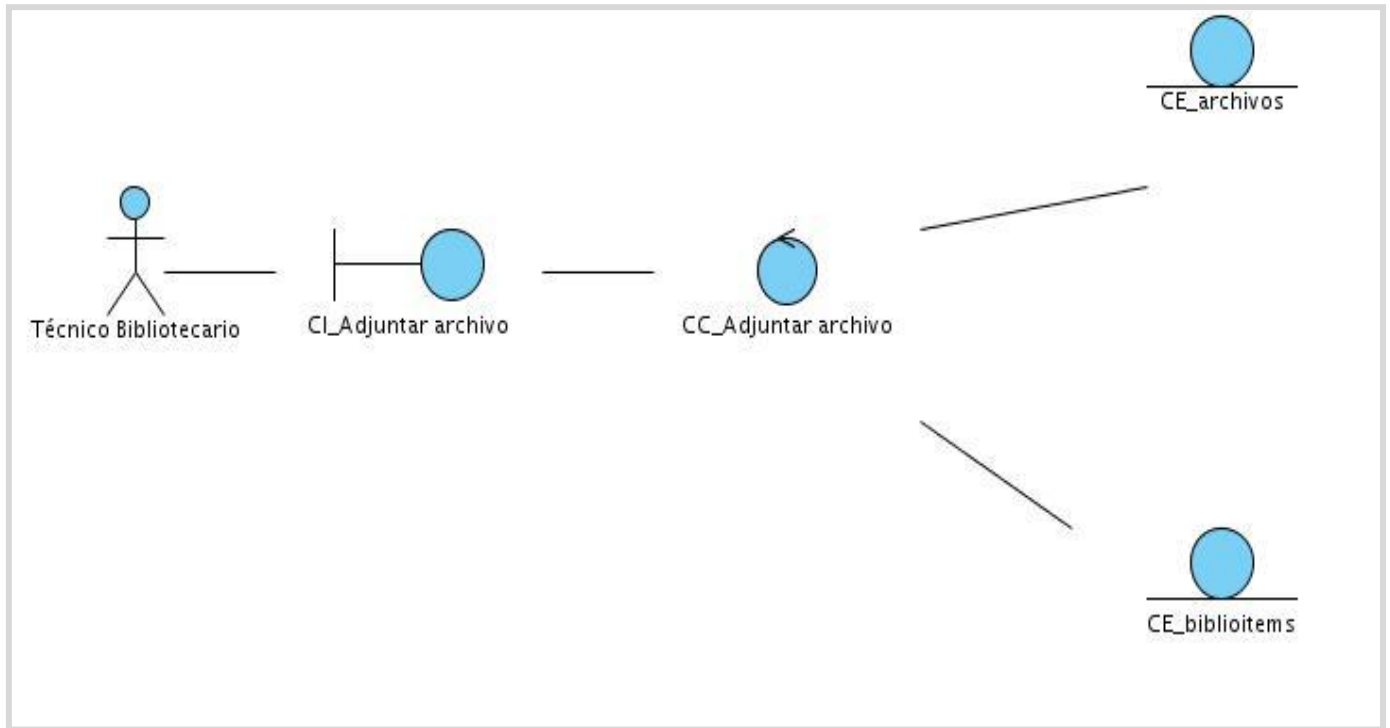


Figura 6. Diagrama de Clase de Análisis del CU Indexar archivo.

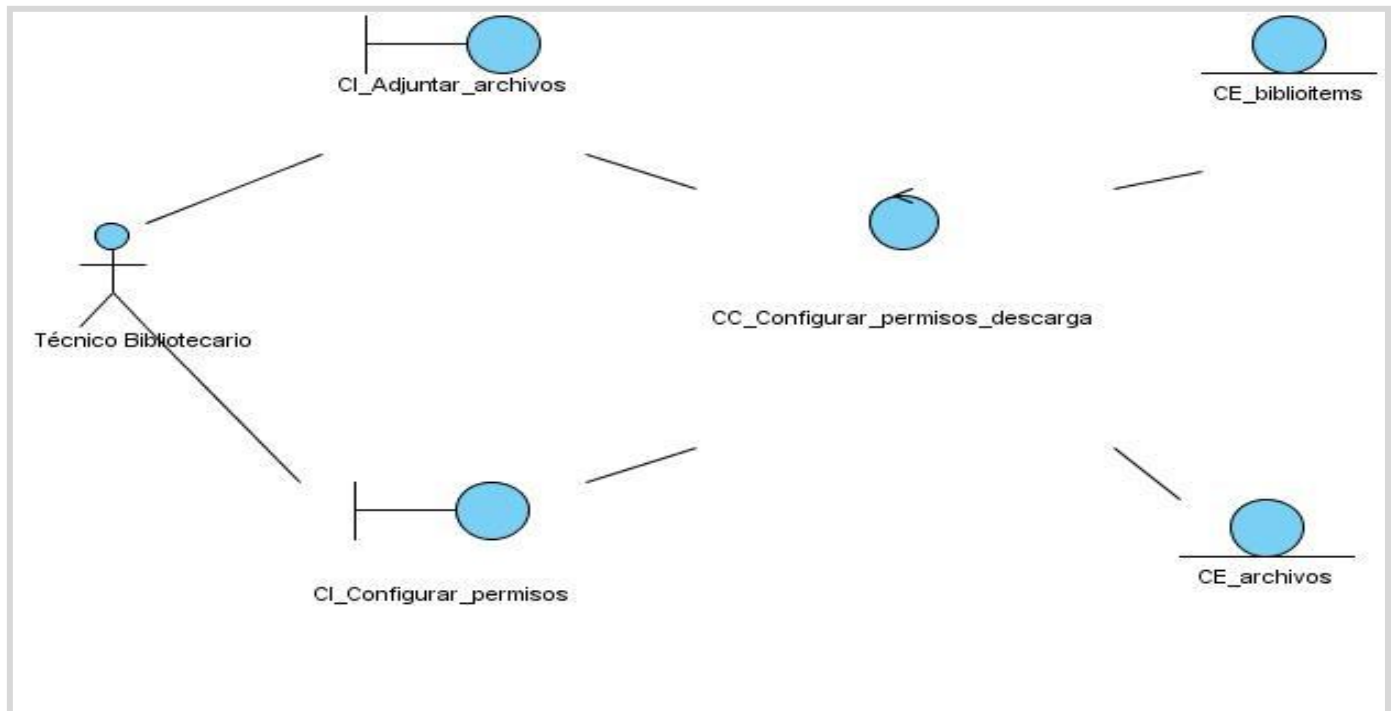


Figura 7. Diagrama de Clase de Análisis del CU Configurar permisos de descarga.



## 3.2. Diseño

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. De hecho, cuando la precisión del diseño es muy grande, la implementación puede ser hecha por un generador automático de código. (29)

A continuación se muestran los diagramas de interacción y clases del diseño definidos para cada caso de uso.

### 3.2.1. Diagramas de interacción

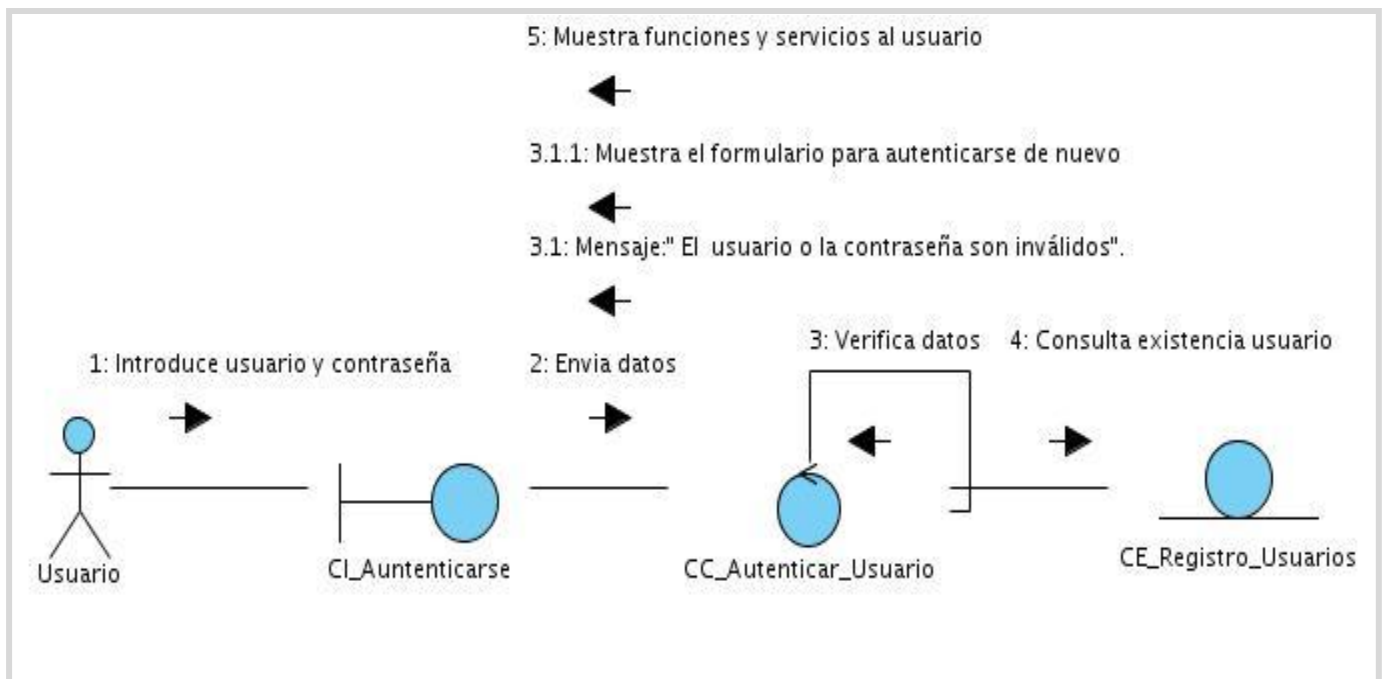


Figura 8. Diagrama de Colaboración del CU Autenticar usuario.

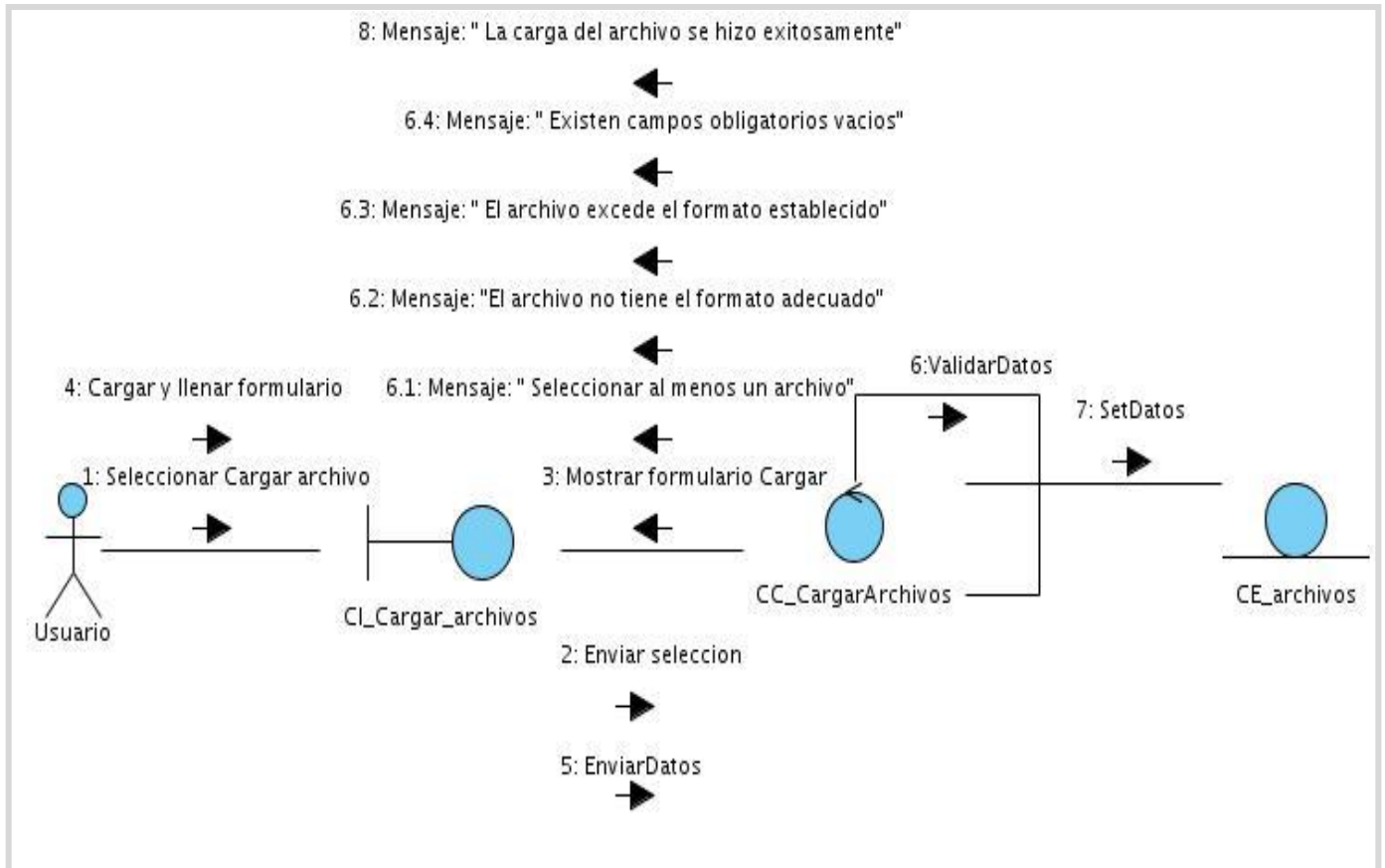


Figura 9. Diagrama de Colaboración del CU Cargar archivos.

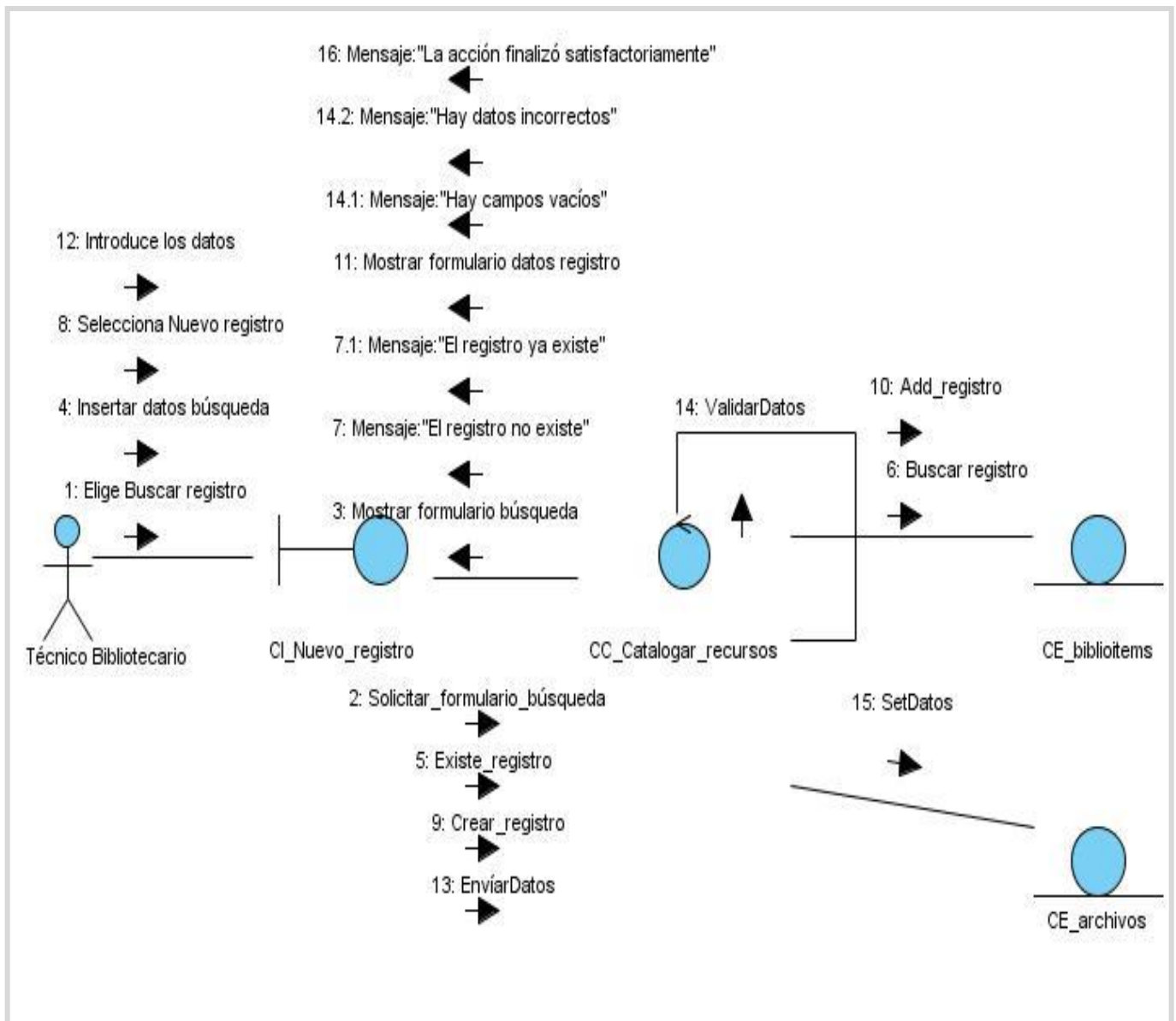


Figura 10. Diagrama de Colaboración del CU Catalogar recursos sin registro a asociado.

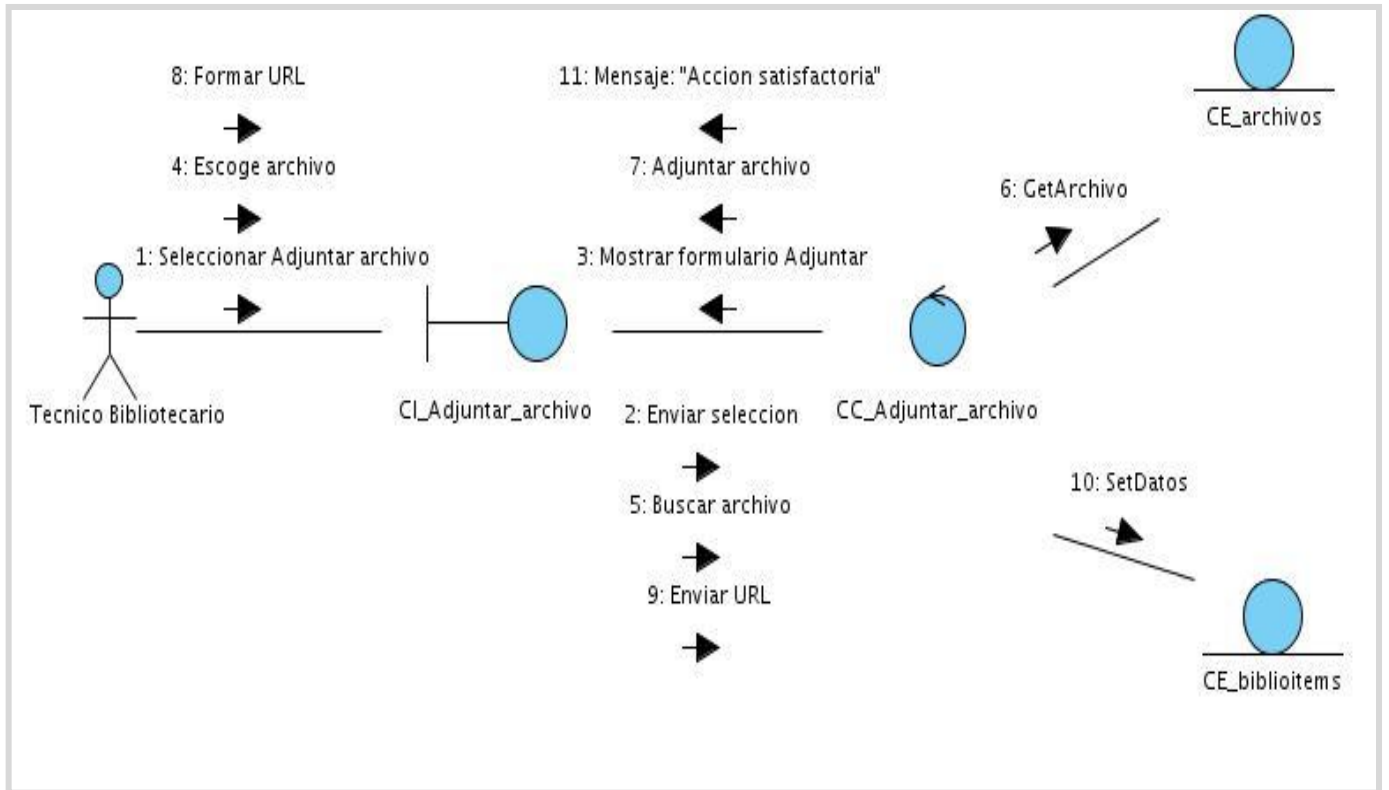


Figura 11. Diagrama de Colaboración del CU Adjuntar archivo.

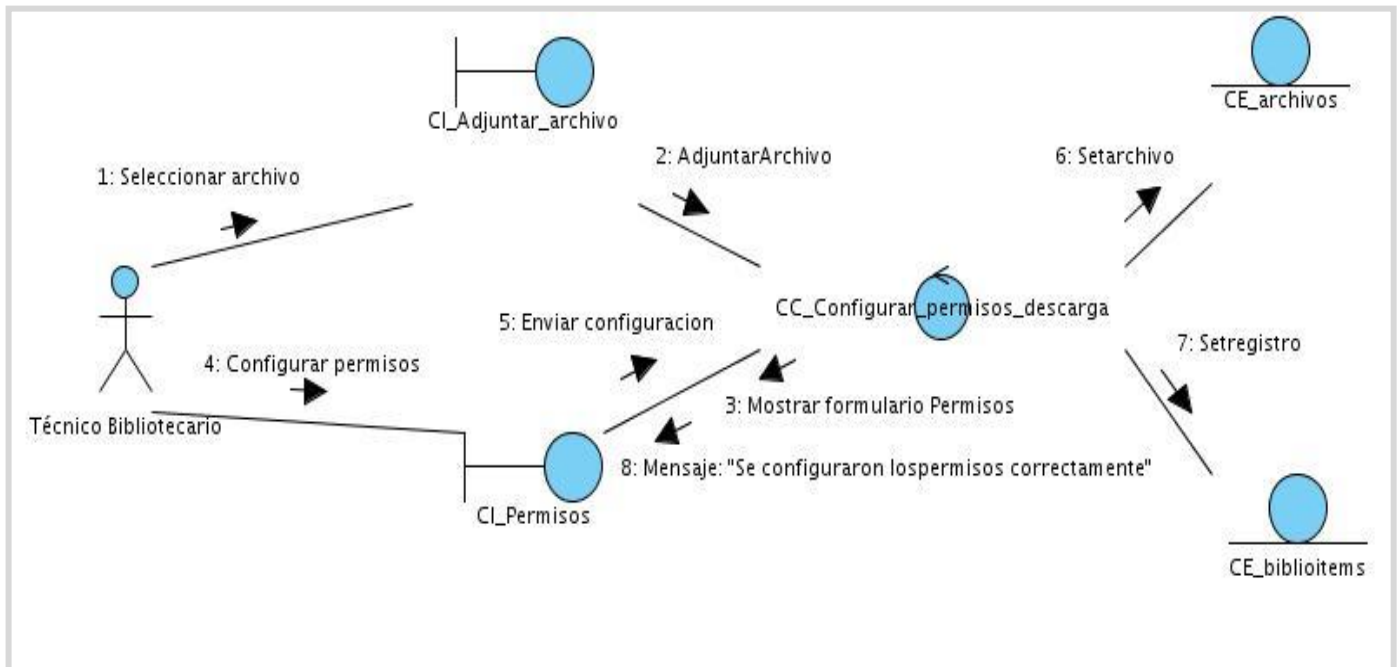


Figura 12. Diagrama de Colaboración del CU Configurar permisos de descarga.

3.2.2. Diagramas de clase del diseño

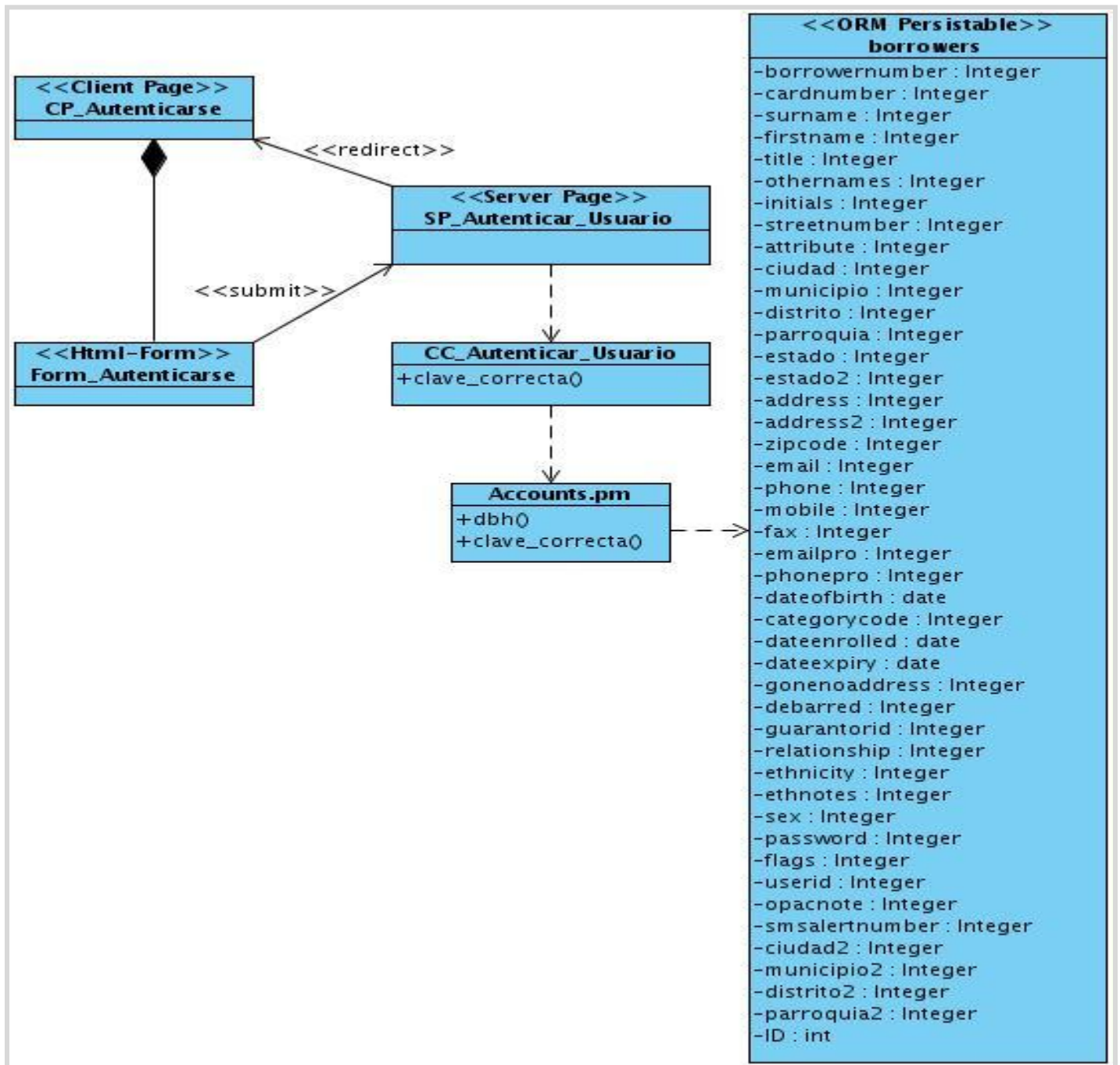


Figura 13. Diagrama de clases del diseño (CU Autenticar usuario).

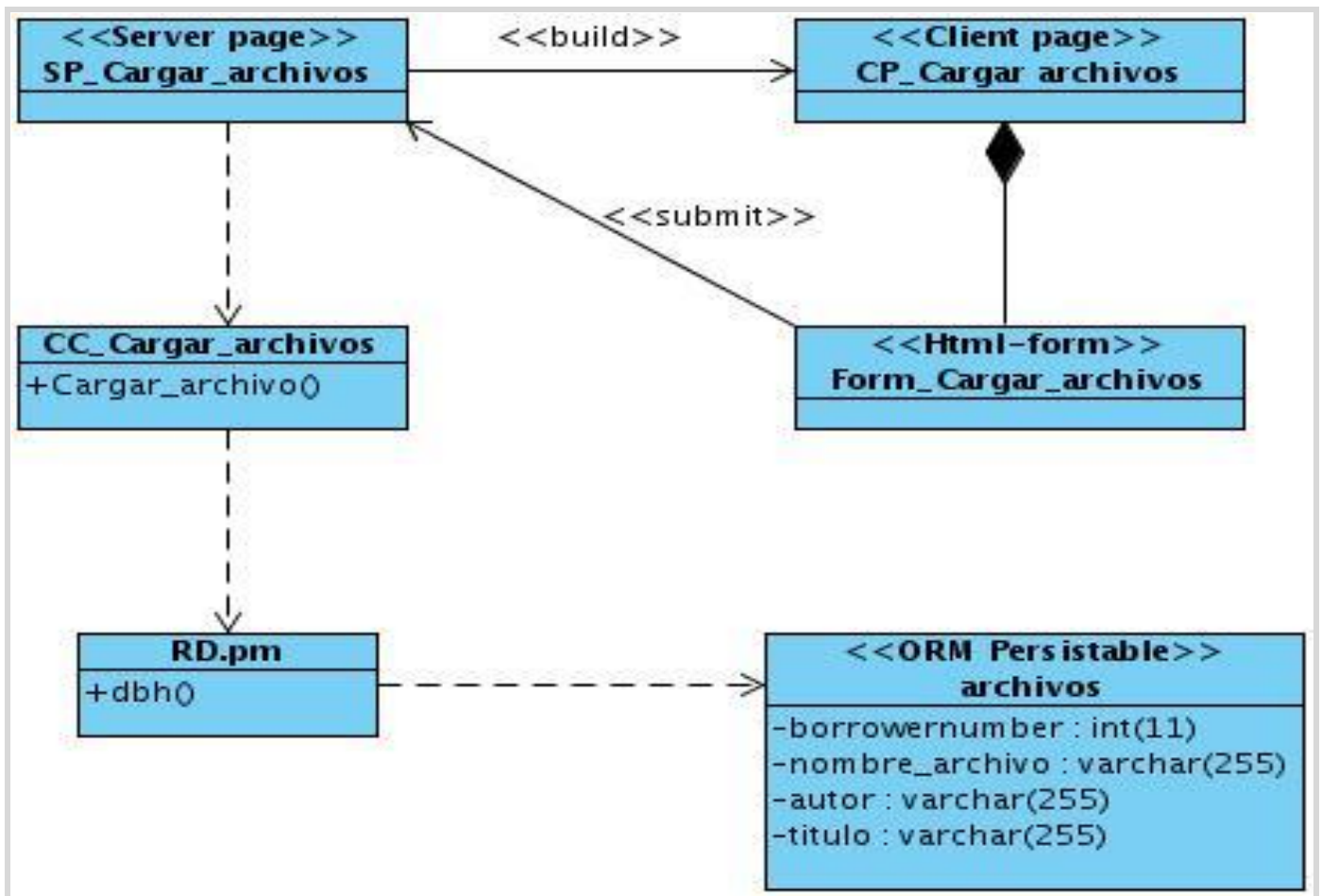


Figura 14. Diagrama de clases del diseño (CU Cargar archivos).



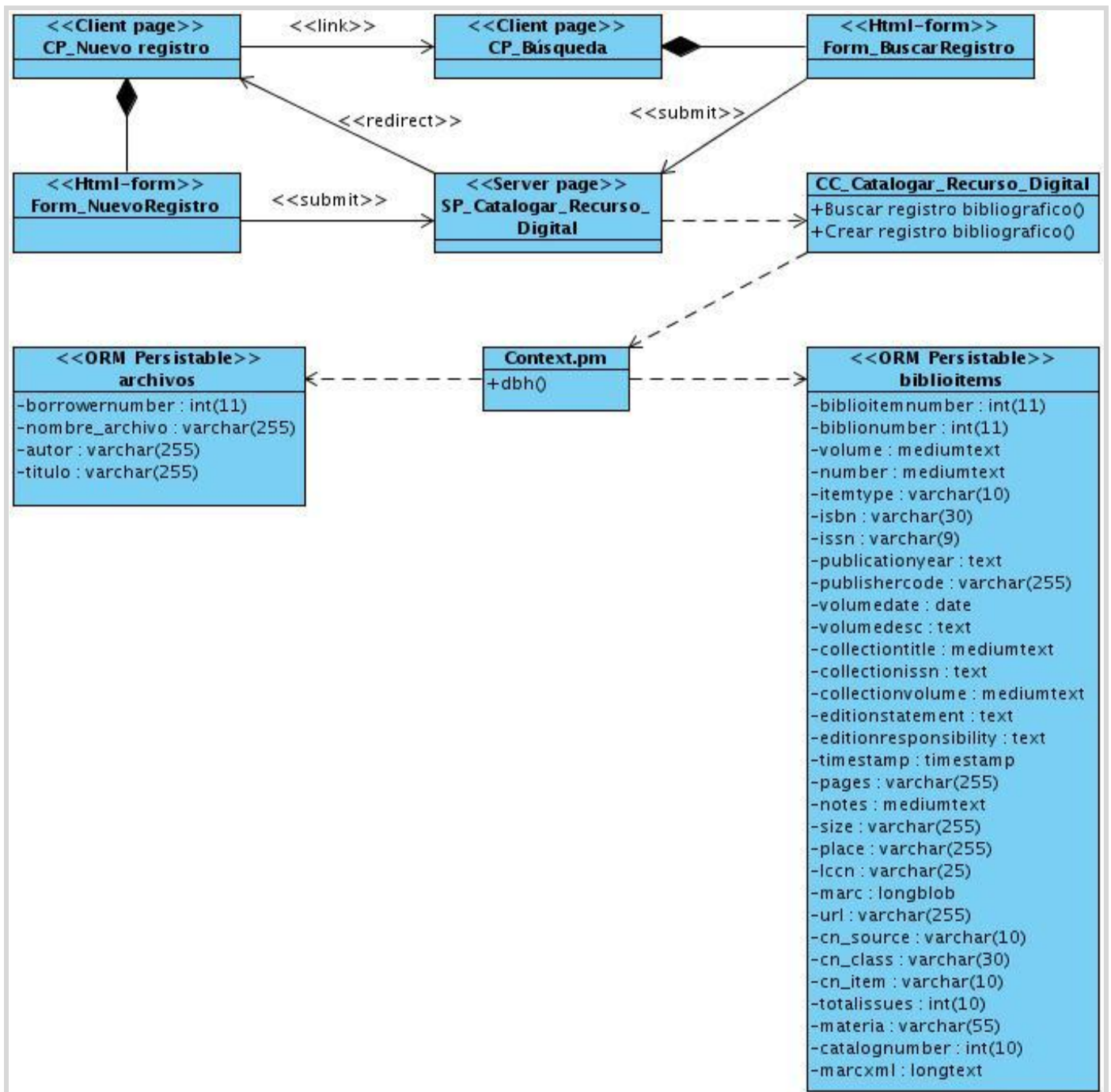


Figura 15. Diagrama de clases del diseño (CU Catalogar recursos sin registro bibliográfico asociado).

## 3.2.3. Diagrama Entidad-Relación de la Base de datos

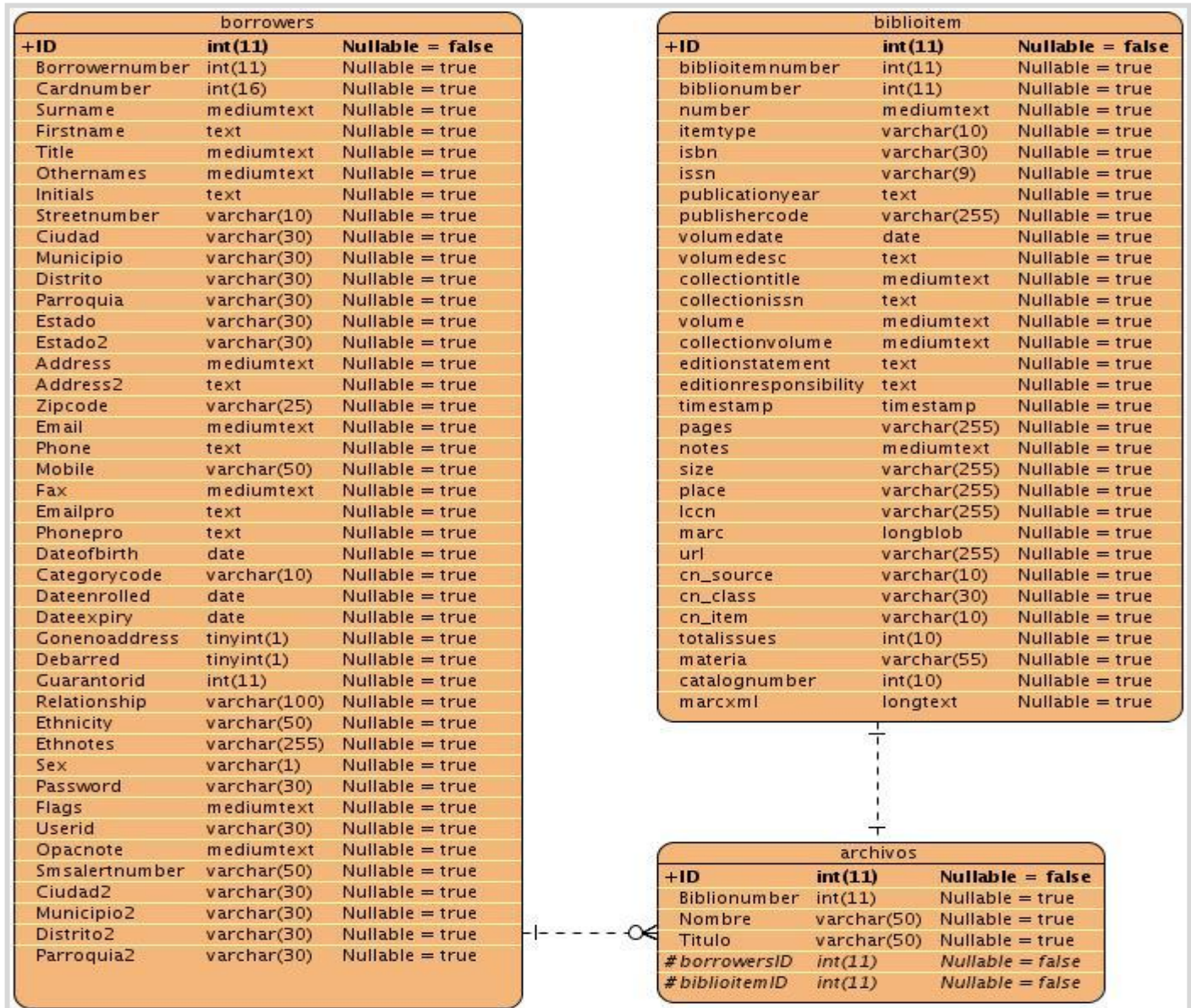


Figura 16. Diagrama Entidad-Relación.



## 3.2.4. Descripción de las tablas de la base de datos

<b>Nombre:</b> biblioitems		
<b>Descripción:</b> Contiene parte de la información de los registros bibliográficos generados por los catalogadores. Permite relacionar los datos del registro bibliográfico con los ejemplares asociados a cada registro.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<u>biblioitemnumber</u>	int(11)	Identificador de la relación entre el registro y el ejemplar
bibliumber	int(11)	Número de registro bibliográfico
volume	mediumtext	Volumen
number	mediumtext	Número
itemtype	varchar(10)	Tipo de material
isbn	varchar(30)	Número estándar internacional de libro
issn	varchar(9)	Número Internacional Normalizado de Publicaciones Seriadas
publicationyear	text	Año de publicación
publishercode	varchar(255)	Código de editor
volumedate	date	Fecha de volumen
volumedesc	text	Descripción de volumen
collectiontitle	mediumtext	Título de colección
collectionissn	text	ISSN de la colección
collectionvolume	mediumtext	Volumen de la colección
editionstatement	text	Enunciado de edición
editionresponsibility	text	Responsable de edición

timestamp	timestamp	Fecha y hora
pages	varchar(255)	Cantidad de páginas
notes	mediumtext	Notas
size	varchar(255)	Tamaño
place	varchar(255)	Lugar
lccn	varchar(25)	Número de clasificación de la Biblioteca del Congreso
marc	longblob	Registro bibliográfico en archivo binario
url	varchar(255)	Url asociada
cn_source	varchar(10)	Fuente de clasificación
cn_class	varchar(30)	Número de clasificación
cn_item	varchar(10)	Cutter
totalissues	int(10)	Total de préstamos
materia	varchar(55)	Materia
catalognumber	int(10)	Número del catalogador
marcxml	longtext	Registro bibliográfico en formato XML

**Tabla 2. Descripción de la tabla biblioitems.**

<b>Nombre:</b> archivos		
<b>Descripción:</b> Contiene la información de los recursos que cuya información se guardará en la base de datos repositorio.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
biblionumber	varchar(11)	Número de registro bibliográfico
nombre	varchar(50)	Nombre del archivo
contenido	longblob	En este campo se guarda la información del archivo
tipo	varchar(20)	Tipo de archivo

**Tabla 3. Descripción de la tabla archivos.**

<b>Nombre:</b> borrowers		
<b>Descripción:</b> Contiene la información de los usuarios de la Biblioteca.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<u>borrowernumber</u>	int(11)	Número de usuario
cardnumber	varchar(16)	Número de carné del usuario (Único)
surname	mediumtext	Apellidos
firstname	text	Primer nombre
title	mediumtext	Título
othernames	mediumtext	Otros nombres
initials	text	Iniciales
streetnumber	varchar(10)	Número de calle
streettype	varchar(50)	Tipo de calle
ciudad	varchar(30)	Ciudad

municipio	varchar(30)	Municipio
distrito	varchar(30)	Distrito
parroquia	varchar(30)	Parroquia
estado	varchar(30)	Estado
estado2	varchar(30)	Estado 2
address	mediumtext	Dirección
address2	text	Dirección 2
zipcode	varchar(25)	Código postal
email	mediumtext	Correo electrónico
phone	text	Teléfono
mobile	varchar(50)	Móvil
fax	mediumtext	Fax
emailpro	text	Correo del centro de trabajo o estudio
phonepro	text	Teléfono del centro de trabajo o estudio
dateofbirth	date	Fecha de nacimiento
categorycode	varchar(10)	Código de categoría
dateenrolled	date	Fecha de inscripción del usuario
dateexpiry	date	Fecha de vencimiento de la inscripción
gonenoaddress	tinyint(1)	Especifica si el usuario tiene dirección dudosa
debarred	tinyint(1)	Especifica si el usuario está excluido
guarantorid	int(11)	Identificación del garante
relationship	varchar(100)	Relación

ethnicity	varchar(50)	Origen étnico
ethnotes	varchar(255)	Notas de origen étnico
sex	varchar(1)	Sexo
password	varchar(30)	Contraseña
flags	int(11)	Número de permiso en el sistema
userid	varchar(30)	Id de usuario
opacnote	mediumtext	Nota de Opac
smsalertnumber	varchar(50)	Número de alerta SMS
ciudad2	varchar(30)	Ciudad 2
municipio2	varchar(30)	Municipio 2
distrito2	varchar(30)	Distrito 2
parroquia2	varchar(30)	Parroquia 2

**Tabla 4. Descripción de la tabla borrowers.**

### 3.3. Arquitectura

La definición oficial de Arquitectura del Software de la **IEEE Std 1471-2000** propone que: *“La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”.* (30)

Existen varios estilos de arquitectura que se utilizan en el desarrollo del software tales como: arquitectura basada en servicios (SOA), arquitectura orientada a objetos y el modelo vista controlador (MVC); este último es el que se decidió utilizar debido a que su principal objetivo es separar la lógica del negocio de la lógica de presentación o interfaz.

## Definición de las partes

El *Modelo* es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. Responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

La *Vista* es el objeto que maneja la presentación visual de los datos representados por el **Modelo**. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

El *Controlador* es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el **Modelo**. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo. (31)

## Procesamiento del patrón MVC

El procesamiento se lleva a cabo entre sus tres componentes de la siguiente manera: El controlador recibe una orden y decide quién la lleva a cabo en el modelo. Una vez que el modelo (la lógica de negocio) termina sus operaciones devuelve el flujo al controlador y este envía el resultado a la vista. (32)

La siguiente figura describe este proceso:



Figura 17. Patrón Modelo-Vista-Controlador.

## **3.4. Tratamiento de errores**

El tratamiento de errores es una tarea muy importante dentro del desarrollo del software debido a que de estos depende en buena medida la calidad externa del sistema. Se utilizaron técnicas de validación con el fin de evitar el envío de información incorrecta por parte del usuario. También se utilizaron mensajes de texto en la página donde se ejecuta la acción, explicando en qué consiste el error y permitiéndole al usuario corregirlo y continuar navegando fácilmente.

## **3.5. Seguridad**

Para el uso del sistema el técnico bibliotecario deberá estar adecuadamente autenticado con su identificador y contraseña correspondientes al igual que los usuarios que accedan al sistema para recibir los servicios que ofrece el mismo. Además de esta medida, las contraseñas viajan encriptadas por la red para evitar que alguien ajeno a la biblioteca se apodere de una cuenta y realice algún cambio que vaya en contra del sistema. También el sistema está preparado contra ataques de Inyección SQL, realizando solamente las conexiones necesarias a la base de datos y utilizando para ello un usuario y contraseña.

## **3.6. Concepción de la ayuda**

El sistema debe contar con diferentes temas de ayuda que le ayuden al usuario a conocer su funcionamiento durante la interacción con el mismo. En fin, la ayuda debe estar disponible y accesible en todas las páginas a las que acceda el usuario para que él pueda consultar la información que necesita en ese instante. Cada página muestra cómo realizar aquellas operaciones que estén relacionadas con la posición donde se encuentre el usuario en dicho momento.

## **Conclusiones del capítulo**

En el presente capítulo se desarrollaron las actividades relacionadas con el análisis y el diseño del sistema. Para ello se representa la lógica del negocio mediante los diagramas de análisis, diagramas de colaboración y de clases del diseño. Además se realizan las descripciones de las tablas de la base de datos que va a utilizar el sistema. De esta forma quedan las condiciones creadas para comenzar la implementación del sistema.

## Capítulo 4. Implementación y prueba

### 4.1. Implementación

#### 4.1.1. Diagrama de despliegue

La Fase de Construcción tiene como propósito clarificar los requisitos restantes y completar el desarrollo del sistema basándose en la línea base de la arquitectura. La implementación es el centro durante las iteraciones de la Fase de Construcción, por lo tanto el flujo de trabajo Implementación describe cómo los elementos del Modelo de Diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el Modelo de Despliegue. Una de las principales actividades de la Disciplina de Implementación es:

- Estructurar el Modelo de Implementación: En esta actividad se establece la estructura de los elementos de implementación, basándose en las responsabilidades asignadas a los subsistemas de implementación y su contenido. (33)

Uno de los artefactos de esta actividad es el modelo de despliegue el cual es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar. Los nodos poseen relaciones que representan medios de comunicación entre ellos. (34)

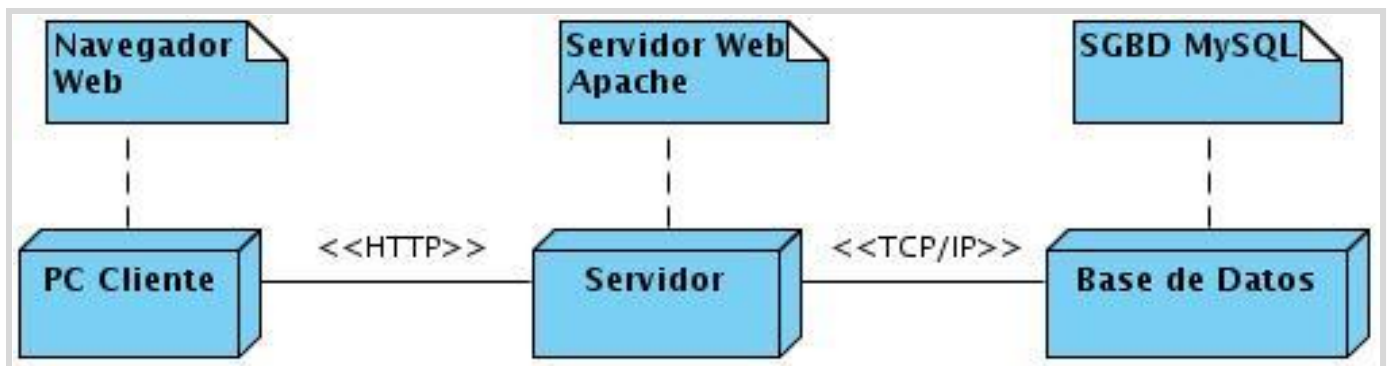


Figura 18. Diagrama de despliegue.



## **Nodo PC Cliente**

Representa las computadoras que los usuarios utilizarán para interactuar con el sistema. Se comunica con el Servidor a través del protocolo HTTP.

## **Nodo Servidor**

En este nodo se encuentran los scripts de la aplicación.

## **Nodo Base de Datos**

En este nodo se encuentra el Servidor de Base de Datos del SIGB bibliotheX.

### **4.1.2. Diagrama de componentes**

- **Componente:** es la parte modular de un sistema, desplegable y reemplazable que encapsula implementación, un conjunto de interfaces y proporciona la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos (ficheros ejecutables, binarios, etc.). Son las piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas.
- **Algunos estereotipos estándar de componentes son:**
- **Ejecutable:** Es un programa que se puede ejecutar en un nodo.
- **Biblioteca:** Es una biblioteca de objetos estática o dinámica.
- **Tabla:** Es una tabla de una Base de Datos.
- **Archivo:** Es un fichero que contiene código fuente o datos.
- **Documento:** Es un documento.
- **Página Web:** Es una página que se obtiene de la ejecución del sistema.

Los Diagramas de Componentes modelan la vista estática de un sistema. Se representa como un grafo de componentes de software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes, por lo cual cada diagrama describe un apartado del sistema. (33)

El subsistema Admon Recursos Digitales está dividido en otros tres subsistemas: el subsistema Controladoras, el subsistema Modelos y el subsistema Vistas. A continuación se mostrará el diagrama de componentes de forma global:

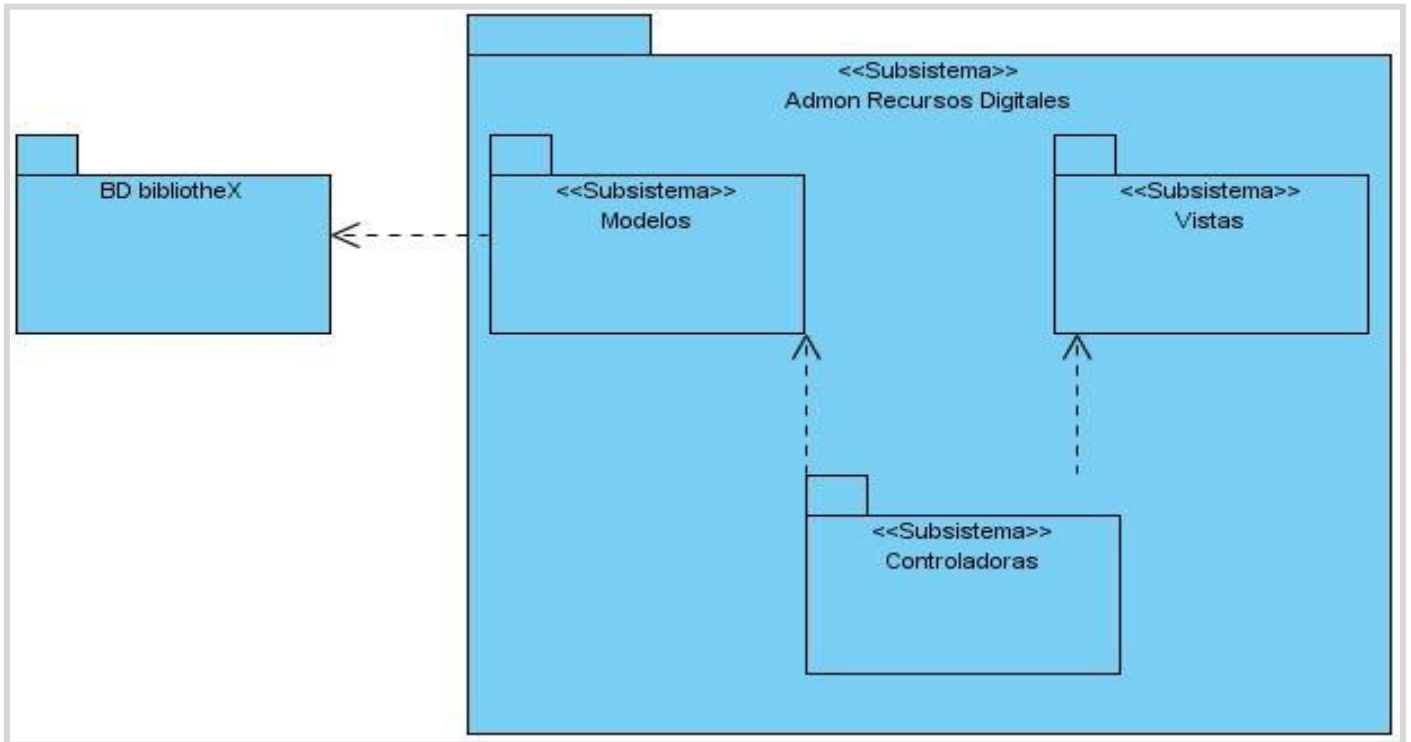


Figura 19. Diagrama de componentes.

Para ilustrar de una manera más clara el funcionamiento del patrón modelo vista controlador, a continuación se mostrará el Subsistema Modelos que se encarga de la conexión con la base de datos para de esta forma gestionar los datos que maneja el sistema.

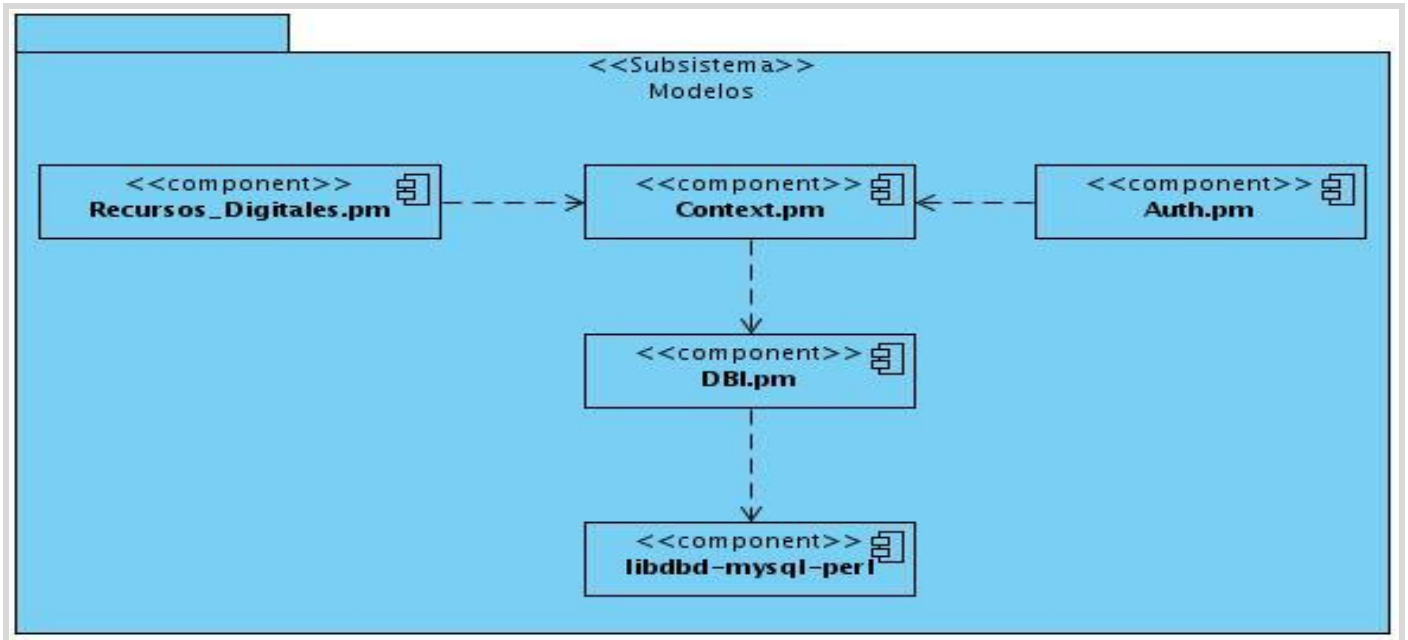


Figura 20. Diagrama de componentes. Subsistema Modelos.

El subsistema Vistas garantiza la interacción del usuario con el sistema. A continuación se representan de forma detallada los componentes de dicho subsistema.

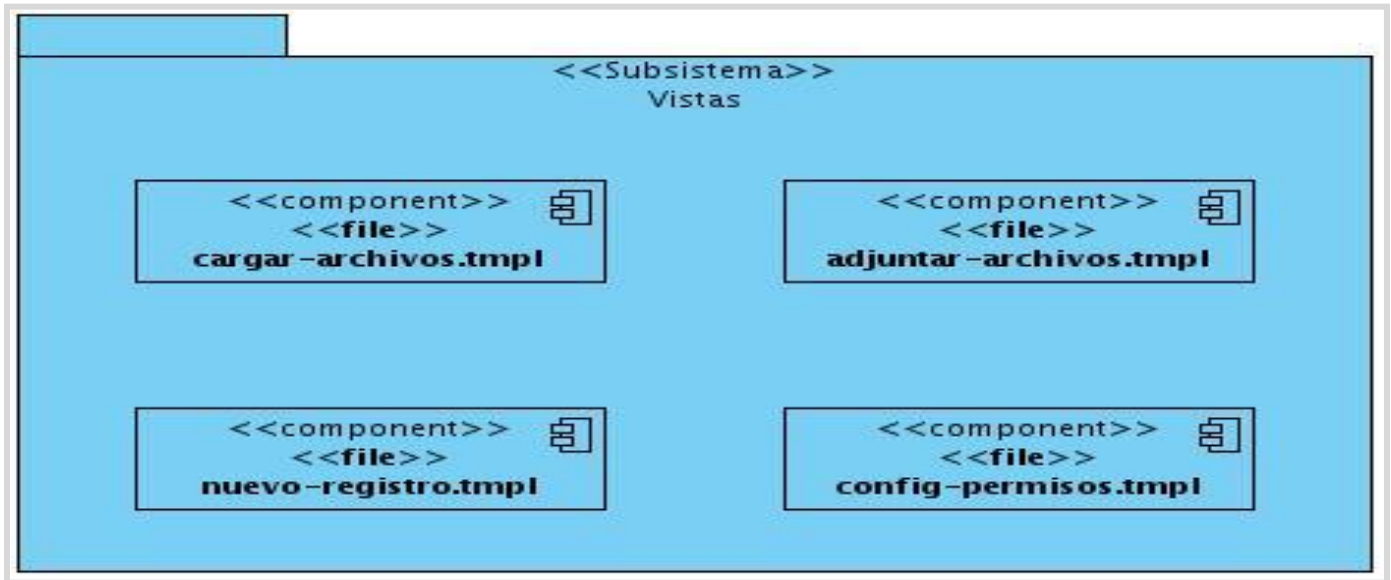


Figura 21. Diagrama de componentes. Subsistema Vistas.

El subsistema Controladoras, dirige las actividades de la aplicación, este contiene los ficheros de código fuente, los cuales interactúan con los demás subsistemas para coordinar las acciones del software. Aquí se representan los detalles del mismo.

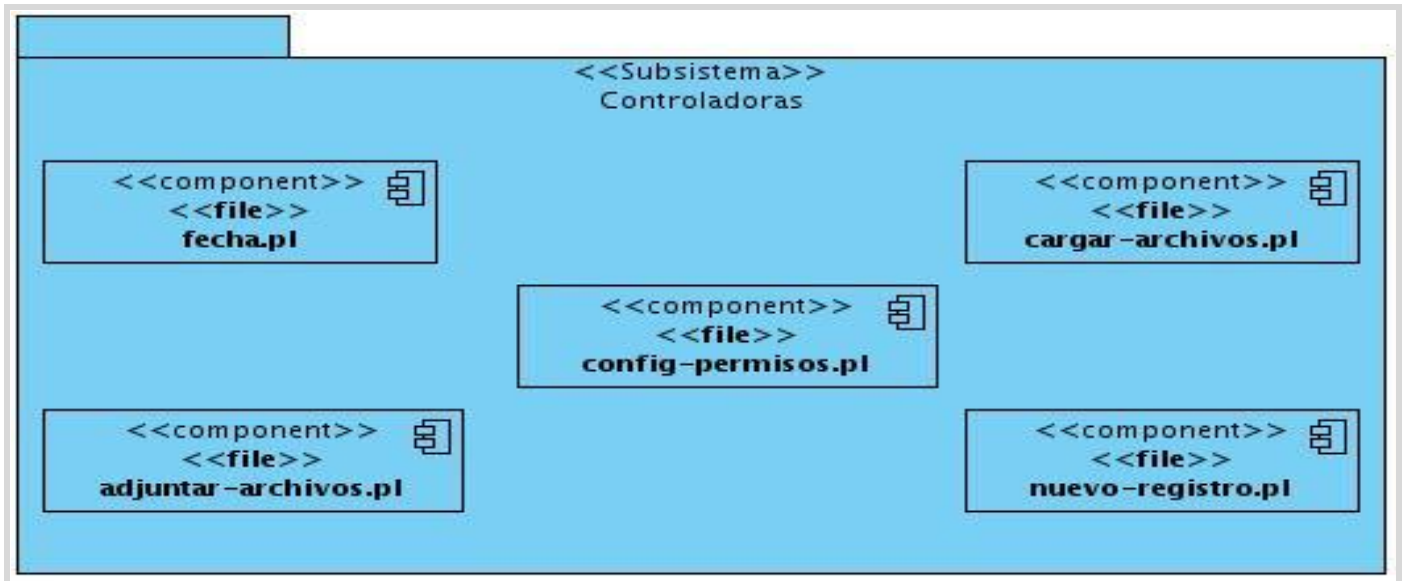


Figura 22. Diagrama de componentes. Subsistema Controladoras.

## 4.2. Modelo de prueba

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

Existen dos métodos de prueba fundamentales: el método de la caja negra y de la caja blanca. 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.

La prueba de la caja blanca del software comprueba los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles.

En este caso se va a utilizar el método de caja negra aplicando la técnica de la Partición de Equivalencia ya que esta técnica es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. (35)

A continuación se describen los casos de prueba realizados a los casos de uso definidos anteriormente:

<b>Nombre del Caso de Uso</b>	<b>Autenticar usuario</b>	
<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El Usuario introduce un nombre de usuario o contraseña incorrectos.	El sistema muestra una notificación: ha introducido un usuario o contraseña incorrecta. Por favor inténtelo nuevamente.	La operación se repite hasta que el Usuario introduce el nombre de usuario o contraseña correcta.
El Usuario introduce correctamente su nombre de usuario y contraseña.	El sistema muestra al usuario las funcionalidades del sistema a las cuales tiene acceso de acuerdo a los permisos concedidos.	El usuario previamente debió pasar por la Administración de la Biblioteca para que se le asocie a la misma y asignarle una contraseña.

**Tabla 5. Modelo de prueba del CU Autenticar usuario.**

<b>Nombre del Caso de Uso</b>	<b>Cargar archivos</b>	
<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El Usuario deja campos obligatorios del formulario vacíos (Título, Autor).	El sistema especifica que dichos campos deben ser llenados por el Usuario para poder cargar archivos.	La operación se repite hasta que el Usuario llene los campos.
El Usuario no selecciona un archivo para cargar.	El sistema especifica que el Usuario debe escoger un archivo.	La operación se repite hasta que el Usuario seleccione un archivo.
El Usuario introduce	El sistema muestra un mensaje:	La operación se repite hasta que

incorrectamente los datos en el campo Título.	El campo Título es incorrecto.	el Usuario no llene correctamente los datos.
El Usuario introduce incorrectamente los datos en el campo Autor.	El sistema muestra un mensaje: El campo Autor es incorrecto.	La operación se repite hasta que el usuario introduzca los datos correctamente.
El Usuario introduce correctamente los datos del formulario y selecciona un archivo a cargar.	El sistema realiza la carga del archivo satisfactoriamente.	El archivo debe estar en los formatos permitidos por el sistema.

Tabla 6. Modelo de prueba del CU Cargar archivos.

### Conclusiones del capítulo

En este capítulo se abordaron aspectos relacionados con los Flujos de Trabajo Implementación y Prueba. Realizándose el diagrama de despliegue que muestra la disposición física del sistema representándose a través de nodos. En el flujo de Prueba se describieron los casos de prueba de cada uno de los casos de uso presentados en el trabajo. Estas pruebas fueron realizadas obteniéndose resultados satisfactorios que validaron el funcionamiento del sistema.

### **CONCLUSIONES GENERALES**

En una sociedad cada vez más interesada en la información y el conocimiento, contar con una biblioteca que pueda acceder a recursos digitales es de gran importancia. Con el desarrollo de este sistema para la administración de recursos digitales se garantiza la agilidad en los procesos de gestión de dichos recursos por parte de los trabajadores de la biblioteca y se satisfacen las necesidades de los usuarios.

Al concluir el presente trabajo, se obtuvieron los siguientes resultados:

- Se seleccionaron las herramientas, lenguajes y tecnologías necesarias para garantizar la correcta integración del módulo de administración de recursos digitales al SIGB bibliotheX.
- Con la investigación realizada en torno al tema de administración de recursos digitales, se pudo destacar la necesidad real que existe en los SIGB de administrar dichos recursos para mejorar los servicios brindados por la biblioteca.
- Se realizó el análisis y diseño del sistema para la administración de recursos digitales para el SIGB bibliotheX.
- Las pruebas realizadas validan la solución propuesta, brindando resultados satisfactorios que demuestran el correcto funcionamiento del módulo de administración de recursos digitales.

## **RECOMENDACIONES**

Para ampliar y mejorar el sistema de administración de recursos digitales propuesto anteriormente, se recomienda:

- Incorporar nuevas funcionalidades al módulo para continuar cumpliendo con las exigencias de los usuarios y continuar la investigación sobre la gestión de los recursos digitales en los SIGB y las necesidades de los usuarios para garantizar mejoras en futuras versiones del mismo.
- Incluir al módulo de administración de recursos digitales en el manual de usuarios con que cuenta el SIGB bibliotheX para que estos puedan navegar con mayor facilidad por el sistema.



## REFERENCIAS BIBLIOGRÁFICAS

1. **Real Academia Española.** rae. [En línea] 2009. [Citado el: 5 de Junio de 2011.] <http://www.rae.es>.
2. **definicion.de.** definicion.de. *Definición de administración de recursos - Qué es, Significado y Concepto.* [En línea] 2010. [Citado el: 2 de Diciembre de 2010.] <http://definicion.de/administracion-de-recursos>.
3. **Lugo, Margarita y Hernandez, Adriana.** Revista Digital Universitaria unam. *Evaluacion de recursos digitales: analisis de algunos parametros.* [En línea] 11 de Julio de 2004. [Citado el: 2 de Diciembre de 2010.] 1607-6079.
4. **microgestio.es.** microgestio.es. *Gestion de Activos Digitales.* [En línea] 2008. [Citado el: 18 de Marzo de 2011.] <http://www.microgestio.es/es/gestion-contenidos-digitales.php>.
5. **Rodríguez-Gairín, Josep Manuel y Sulé Duesa, Andreu.** *DSpace: un manual específico para gestores de la información y la documentación.* [En línea] 20 de Junio de 2008. [Citado el: 4 de Mayo de 2011.] <http://www.ub.edu/bid/20rodri2.htm>.
6. **Sarduy Domínguez, MSc. Yanetsys y Urra González, Lic. Pedro.** *Herramientas para la creación de colecciones digitales.* 5, La Habana : s.n., 19 de Mayo de 2006, Acimed, Vol. 14.
7. **desarrolloweb.** desarrolloweb.com. *Tipos de Datos de Mysql.* [En línea] Febrero de 2003. [Citado el: 8 de Abril de 2011.] <http://www.desarrolloweb.com/articulos/1054.php>.
8. *Tipo de datos Blob.* [En línea] 2009. [Citado el: 8 de Abril de 2011.] <http://www.gxtechnical.com/gxdlsp/pub/genexus/devenv/docum/releasenotes/8.0/blob.htm>.
9. **desarrolloweb.** desarrolloweb.com. *Qué es Perl.* [En línea] Septiembre de 2001. [Citado el: 3 de Diciembre de 2010.] <http://www.desarrolloweb.com/articulos/541.php>.
10. **Alvarez, Miguel Angel.** desarrolloweb.com. *Qué es HTML.* [En línea] 2001. [Citado el: 3 de Diciembre de 2010.] <http://www.desarrolloweb.com/articulos/que-es-html.html>.
11. **Eguíluz Pére, Javier.** LibrosWeb.es. *Introducción a JavaScript. Capítulo 1. Introducción.* . [En línea] 2001. [Citado el: 3 de Diciembre de 2010.] <http://www.librosweb.es/javascript/capitulo1.html>.
12. **Eguíluz Pérez, Javier.** LibrosWeb.es. *Introducción a CSS.* [En línea] 2001. [Citado el: 3 de Diciembre de 2010.] <http://www.librosweb.es/css/>.

13. **Pereira Girón, Marco Antonio.** scribd. *UML(Unified Modeling Language)*. [En línea] 2007. [Citado el: 6 de Diciembre de 2010.] <http://es.scribd.com/doc/2080534/UML>.
14. *Quanta Plus, Editor Web Open Source.* [En línea] Noviembre de 2009. [Citado el: 17 de Marzo de 2011.] <http://www.web-mix.ws/pyme/2009/11/quanta-plus-editor-web-open-source>.
15. **Gómez, Julián.** *Eclipse SDK - Descargar.* [En línea] 2008. [Citado el: 17 de Marzo de 2011.] <http://eclipse-sdk.softonic.com>.
16. **Ortega Retureta, Laura Susana y Roselló Carrazana, Adnier.** Desarrollo del módulo Adquisición del Sistema Integrado de Gestión Bibliotecaria Koha para la Biblioteca Nacional “José Martí. Universidad de las Ciencias Informáticas : s.n., 2010.
17. **Visual Paradigm International Ltd.** freedownloadmanager. *Visual Paradigm for UML (ME)*. [En línea] Marzo de 2007. [Citado el: 14 de Diciembre de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%C3%8D\)\\_14720\\_p](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p).
18. **Warp Networks - Sharing Innovation.** warp. *MySQL, “10 razones para usar MySQL.* [En línea] 2008. [Citado el: 17 de Junio de 2011.] <http://warp.es/mysql/productos/razones>.
19. **Collins-Sussman, Ben, Fitzpatrick, Brian W. y Pilato, C. Michael.** Subversion. *Capítulo 1. Control de versiones con Subversion.* [En línea] 2004. [Citado el: 14 de Diciembre de 2010.] <http://svnbook.red-bean.com/index.es.html>.
20. **El CoDiGo K.** elcodigok. *RapidSVN, un cliente gráfico para repositorios SVN.* [En línea] Octubre de 2008. [Citado el: 17 de Marzo de 2011.] <http://www.elcodigok.com.ar/2008/10/rapidsvn-un-cliente-grafico-para-repositorios-svn>.
21. **ciberaula.** ciberaula. *Una Introducción a Apache.* [En línea] 2010. [Citado el: 17 de Febrero de 2011.] [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro](http://linux.ciberaula.com/articulo/linux_apache_intro).
22. **Van Der Henst S., Christian.** maestrosdelweb. *Qué es el CGI.* [En línea] 21 de Marzo de 2001. [Citado el: 7 de Diciembre de 2010.] <http://www.maestrosdelweb.com/editorial/cgiintro>.
23. **Letelier, Patricio y Penadés, M<sup>a</sup> Carmen.** EVA. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* [En línea] 2011. [Citado el: 5 de Junio de 2011.]

24. **Peñalver, G, Meneses, A y García, S.** EVA. *SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE*. [En línea] 2011. [Citado el: 5 de Junio de 2011.]
25. **Gallego Gómez, Juan Pablo y Galves, Ing. Jorge.** scribd. *FUNDAMENTOS DE LA METODOLOGIA RUP. SOBRE EL PROCESO RACIONAL UNIFICADO*. [En línea] 16 de Septiembre de 2007. [Citado el: 9 de Diciembre de 2010.] <http://www.scribd.com/doc/297224/RUP>.
26. **Universidad Politécnica de Valencia.** EVA. *Rational Unified Process (RUP)*. [En línea] 2010. [Citado el: 5 de Junio de 2011.]
27. EVA. *Fase de Inicio. Disciplina de Modelamiento del negocio*. [En línea] 2011. [Citado el: 24 de Marzo de 2011.]
28. EVA. *Introducción a la Disciplina de Requisitos de RUP*. [En línea] 2010. [Citado el: 24 de Marzo de 2011.]
29. **EcuRed.** EcuRed. *Flujo de Trabajo Análisis y Diseño*. [En línea] 2010. [Citado el: 7 de Abril de 2011.] [http://www.ecured.cu/index.php/Flujo\\_de\\_Trabajo\\_An%C3%A1lisis\\_y\\_Dise%C3%B1o#Modelo\\_del\\_An.C3.A1lisis](http://www.ecured.cu/index.php/Flujo_de_Trabajo_An%C3%A1lisis_y_Dise%C3%B1o#Modelo_del_An.C3.A1lisis).
30. **desarrolloweb.** desarrolloweb. *Usabilidad y arquitectura del software*. [En línea] Septiembre de 2004. [Citado el: 10 de Abril de 2011.] <http://www.desarrolloweb.com/articulos/1622.php>.
31. **Paraiso Linux.** paraisolinux.com. *Arquitectura MVC*. [En línea] Marzo de 2009. [Citado el: 11 de Abril de 2011.] <http://paraisolinux.com/arquitectura-mvc>.
32. **Carrodegas Oliva, Rogelio y León Piñero, Roberlay.** Automatización del módulo de Referencia de la Biblioteca de la Universidad de las Ciencias Informáticas. Universidad de las Ciencias Informáticas : s.n., 2009.
33. **UCI.** EVA. *Conferencia 6: Fase de Construcción*. [En línea] 2010-2011. [Citado el: 27 de Abril de 2011.]
34. **García Peñalvo, Dr. Francisco José y Conde González, Miguel Ángel.** scribd. *Ingeniería del Software Tema 6: Diseño orientado a objetos*. [En línea] Octubre de 2008. [Citado el: 27 de Junio de 2011.] <http://es.scribd.com/doc/49708406/19/Modelo-de-despliegue>.
35. **UCI.** EVA. *Conferencia 7 ISW II*. [En línea] 2010-2011. [Citado el: 27 de Abril de 2011.]

## BIBLIOGRAFÍA

1. **Navarrete, J.** inQnable. [En línea] 2009. [Citado el: 4 de Marzo de 2011.] <<http://www.inqnable.es/temario/especifico/historia-libro-bibliotecas/Evolucion%20historica%20de%20la%20bibliotecas.%20Las%20bibliotecas%20en%20el%20siglo%20XXI.pdf/view>>.
2. **Columbié, Radamés Linares.** Acimed. [En línea] 8 de Septiembre de 2008. [Citado el: 4 de Marzo de 2011.] [http://bvs.sld.cu/revistas/aci/vol8\\_3\\_00/aci09300.htm](http://bvs.sld.cu/revistas/aci/vol8_3_00/aci09300.htm).
3. **El libro y la bibliotecas en los siglos XVIII, XIX y XX.** Islabahia. [En línea] Mayo de 2010. [Citado el: 4 de Marzo de 2011.] [http://www.islabahia.com/arenaycal/2010/171\\_mayo/miguel\\_a\\_fernandez171.asp](http://www.islabahia.com/arenaycal/2010/171_mayo/miguel_a_fernandez171.asp).
4. **Definicion.de.** Definicion.de. [En línea] 2010. [Citado el: 2 de Diciembre de 2010.] <http://definicion.de/administracion-de-recursos/>.
5. **Lugo, Margarita y Hernandez, Adriana.** Evaluacion de recursos digitales: analisis de algunos parametros. [En línea] 2004. <http://www.revista.unam.mx/vol.5/num6/art38/art38.htm>. 1607-60792.
6. **Tecnologias-informacion.** Tecnologias-informacion. [En línea] 2009. [Citado el: 25 de Marzo de 2011.] <http://www.tecnologias-informacion.com/gestionactivosdigitales.html>.
7. **greendata.** greendata. [En línea] 2010. [Citado el: 18 de Marzo de 2011.] <http://www.greendata.es/gestion-de-recursos-electronicos>.
8. **Sarduy Domínguez, MSc. Yanetsys y Urra González, Lic. Pedro.** *Herramientas para la creación de colecciones digitales.* 5, La Habana : s.n., 19 de Mayo de 2006, Acimed, Vol. 14.
9. **Gallo, Patricia Russo.** scribd. [En línea] 2008. [Citado el: 9 de Marzo de 2011.] <http://es.scribd.com/doc/44542992/8A7D3D9Ed01>.
10. **Real Academia Española.** rae. [En línea] 2009. [Citado el: 5 de Junio de 2011.] <http://www.rae.es>.
11. **microgestio.es.** microgestio.es. [En línea] 2008. [Citado el: 18 de Marzo de 2011.] <http://www.microgestio.es/es/gestion-contenidos-digitales.php>.
12. **Rodríguez-Gairín, Josep Manuel y Sulé Duesa, Andreu.** [En línea] 20 de Junio de 2008. [Citado el: 4 de Mayo de 2011.] <http://www.ub.edu/bid/20rodri2.htm>.

13. **desarrolloweb.** desarrolloweb. [En línea] Febrero de 2003. [Citado el: 8 de Abril de 2011.] <http://www.desarrolloweb.com/articulos/1054.php>.
14. [En línea] 2010. [Citado el: 8 de Abril de 2011.] <http://www.gxtechnical.com/gxdlsp/pub/genexus/devenv/docum/releasenotes/8.0/blob.htm>.
15. **mysql-hispano.** mysql-hispano. *Manejo de datos BLOB con PHP y MySQL*. [En línea] 27 de Agosto de 2003. [Citado el: 8 de Abril de 2011.] [www.mysql-hispano.org](http://www.mysql-hispano.org).
16. **Alvarez, Miguel Angel.** desarrolloweb. [En línea] Septiembre de 2001. [Citado el: 3 de Diciembre de 2010.] <http://www.desarrolloweb.com/articulos/541.php>.
17. —. desarrolloweb. *Qué es HTML*. [En línea] 2001. [Citado el: 3 de Diciembre de 2011.] <http://www.desarrolloweb.com/articulos/que-es-html.html>.
18. **Eguíluz Pérez, Javier.** librosweb. *Introducción a JavaScript*. [En línea] 2001. [Citado el: 3 de Diciembre de 2010.] <http://www.librosweb.es/javascript/>.
19. **Cárdenas, Lola y Gracia, Joaquin.** webestilo. *Conceptos basicos*. [En línea] 2001. [Citado el: 3 de Diciembre de 2011.] <http://www.webestilo.com/javascript/js00.phtml>.
20. **Eguíluz Pérez, Javier.** librosweb. *Introducción a CSS*. [En línea] 2001. [Citado el: 3 de Diciembre de 2010.] <http://www.librosweb.es/css/>.
21. **Casasola Romero, Oscar.** programacion.com. *Introducción a UML*. [En línea] 2005. [Citado el: 3 de Diciembre de 2010.] [http://www.programacion.com/articulo/introduccion\\_a\\_uml\\_181](http://www.programacion.com/articulo/introduccion_a_uml_181).
22. **Pereira Girón, Marco Antonio.** scribd. *UML(Unified Modeling Language)*. [En línea] 2007. [Citado el: 6 de Diciembre de 2010.] <http://es.scribd.com/doc/2080534/UML>.
23. *Quanta Plus, Editor Web Open Source*. [En línea] Noviembre de 2009. [Citado el: 17 de Marzo de 2011.] <http://www.web-mix.ws/pyme/2009/11/quanta-plus-editor-open-source>.
24. **Gómez, Julián.** *Eclipse SDK*. [En línea] 2010. [Citado el: 17 de Marzo de 2011.] <http://eclipse-sdk.softonic.com>.
25. **Ortega Retureta, Susana Laura y Roselló Carrazana, Adnier.** *Desarrollo del Módulo Adquisición del Sistema Integrado de Gestión Bibliotecaria Koha para la Biblioteca Nacional José Martí*. Junio : s.n., 2010.

26. **Visual Paradigm International Ltd.** downloadmanager. *Visual Paradigm*. [En línea] 2007. [Citado el: 6 de Diciembre de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%28M%C3%8D%29\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/).
27. **Warp Networks- Sharing Innovation.** warp. [En línea] 2008. [Citado el: 17 de Junio de 2011.] <http://warp.es/mysql/productos/razones>.
28. **El CoDiGo K.** El CoDiGo K. *RapidSVN, un cliente gráfico para repositorios SVN*. [En línea] Octubre de 20Citado el: 17 de Marzo de 2011.] <http://www.elcodigok.com.ar/2008/10/rapidsvn-un-cliente-grafico-para-repositorios-svn>.
29. **Collins-Sussman, Ben, Fitzpatrick, Brian W. y Pilato, C. Michael.** Subversion. *Capítulo 1. Control de versiones con Subversion*. [En línea] 2004. [Citado el: 14 de Diciembre de 2010.] <http://svnbook.red-bean.com/index.es.html>.
30. **ciberaula.** ciberaula. *Una introducción a Apache*. [En línea] 2010. [Citado el: 17 de Febrero de 2011.] [http://www.ciberaula.com/articulo/linux\\_aoache\\_intro](http://www.ciberaula.com/articulo/linux_aoache_intro).
31. **S., Christian Van Der Henst.** maestrosdelweb. *Qué es el CGI*. [En línea] 21 de Marzo de 2001. [Citado el: 7 de Diciembre de 2010.] <http://www.maestrosdelweb.com/editorial/cgiintro>.
32. **Letelier, Patricio y Penadés, María del Carmen.** EVA. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. [En línea] 2011. [Citado el: 5 de Junio de 2011.]
33. **Peñalver, G, Meneses, A y García, S.** EVA. *SXP, Metodología ágil para el desarrollo de software*. [En línea] 2011. [Citado el: 5 de Junio de 2011.]
34. **Gallego Gómez, Juan Pablo y Galves, Ing. Jorge.** scribd. *Fundamentos de la metodología RUP. Sobre el Proceso Racional Unificado*. [En línea] 16 de Septiembre de 2007. [Citado el: 9 de Diciembre de 2010.] <http://www.scribd.com/doc/297224/RUP>.
35. **Universidad Politécnica de Valencia.** EVA. *Rational Unified Process (RUP)*. [En línea] 2010. [Citado el: 5 de Junio de 2011.]
36. EVA. Curso Ingeniería de Software. *Fase de Inicio. Disciplina de Modelamiento del negocio*. [En línea] 2010. [Citado el: 24 de Marzo de 2011.]

37. EVA. Curso Ingeniería de Software I. *Introducción a la Disciplina de Requisitos de RUP.* . [En línea] 2010. [Citado el: 24 de Marzo de 2011.]
38. **EcuRed.** EcuRed. *Flujo de Trabajo Análisis y Diseño.* [En línea] 2010. [Citado el: 7 de Abril de 2011.] [http://www.ecured.cu/index.php/Flujo\\_de\\_Trabajo\\_An%C3%A1lisis\\_y\\_Dise%C3%B1o#Modelo\\_del\\_An.C3.A1lisis](http://www.ecured.cu/index.php/Flujo_de_Trabajo_An%C3%A1lisis_y_Dise%C3%B1o#Modelo_del_An.C3.A1lisis).
39. **desarrolloweb.** desarrolloweb. *Usabilidad y arquitectura del software.* [En línea] Septiembre de 2004. [Citado el: 10 de Abril de 2011.] <http://www.desarrolloweb.com/articulos/1622.php>.
40. **Paraiso Linux.** Paraiso Linux. *Arquitectura MVC.* [En línea] Marzo de 2009. [Citado el: 11 de Abril de 2011.] <http://paraisolinux.com/arquitectura-mvc>.
41. **Carrodegua Oliva, Rogelio y León Piñero, Roberlay.** “Automatización del módulo de Referencia de la Biblioteca de la Universidad de las Ciencias Informáticas.”. Universidad de las Ciencias Informáticas : s.n., 2009.
42. **UCI.** EVA. ISW II. *Conferencia 6: Fase de Construcción.* [En línea] 2011. [Citado el: 27 de Abril de 2011.]
43. **García Peñalvo, Dr. Francisco José y Conde González, Miguel Ángel.** scribd. *Ingeniería del Software. Tema 6: Diseño orientado a objetos.* [En línea] Octubre de 2008. [Citado el: 27 de Junio de 2011.] <http://es.scribd.com/doc/49708406/19/Modelo-de-despliegue>.
44. **UCI.** EVA. *Conferencia 7. ISW II.* [En línea] 2010-2011. [Citado el: 27 de Abril de 2011.]
45. **Arriola Navarrete, Óscar; Butrón Yáñez, Katya.** [En línea] 18 de Marzo de 2010. [Citado el: 2 de Diciembre de 2010.] <http://exagonobibliotecario.blogspot.com/2010/03/sistemas-integrales-para-la.html>.
46. -. Metadatos: concepto y motivación. [En línea] 2010. [Citado el: 7 de junio de 2011.] <http://www.sedic.es/autoformacion/metadatos/tema1.htm>.
47. **Glosario.** glosario.net. Definición de Registro bibliográfico - Diccionario de Términos Bibliotecarios. [En línea] 26 de Abril de 2009. [Citado el: 10 de Junio de 2011.] <http://cultura.glosario.net/terminos-bibliotecarios/registro-bibliogr%E1fico-12427.html>.
48. **Koha-unlp.org.** Koha UNLP. [En línea] 2001. [Citado el: 28 de Junio de 2011.] <http://koha.unlp.edu.ar/>.

49. **Hillmann, Diane.** Guía de uso del Dublin Core. [En línea] 26 de Agosto de 2003. [Citado el: 17 de Marzo de 2010.] <http://dublincore.org/documents/2003/08/26/usageguide/>.