



**SISTEMA DE GESTIÓN DE BIBLIOTECAS  
“Nikolaus Walczuch”  
BAJO SOFTWARE LIBRE**

**Autor: Carlos Luis Moreno Sarmiento**

**Tutor Académico: Prof. Domingo Hernández M.Sc**

**Cotutor: Prof. Dulce Milagro Rivero PhD**

**Proyecto de Grado presentado ante la ilustre Universidad de Los Andes  
como requisito final para optar al título de Ingeniero de Sistemas**

**UNIVERSIDAD DE LOS ANDES  
FACULTAD DE INGENIERIA  
ESCUELA DE SISTEMAS  
Mérida-Venezuela  
2006**



**SISTEMA DE GESTIÓN DE BIBLIOTECAS  
“Nikolaus Walczuch”  
BAJO SOFTWARE LIBRE**

**Autor: Carlos Luis Moreno Sarmiento**

**Tutor Académico: Prof. Domingo Hernández M.Sc**

**Cotutor: Prof. Dulce Milagro Rivero PhD**

**Proyecto de Grado presentado ante la ilustre Universidad de Los Andes  
como requisito final para optar al título de Ingeniero de Sistemas**

**Prof. Herbert Hoeger  
Jurado Principal**

**Prof. Pablo Lischinsky  
Jurado Suplente**

**UNIVERSIDAD DE LOS ANDES  
FACULTAD DE INGENIERIA  
ESCUELA DE SISTEMAS  
Mérida-Venezuela  
2006**

## RESUMEN

Este trabajo de grado lleva como nombre **Sistema de Gestión de Bibliotecas “Nikolaus Walczuch” bajo software libre**, tiene como finalidad la colaboración a Instituciones académicas públicas o privadas que quieran incorporarse a la automatización de sus bibliotecas, a través de una página Web definida en este proyecto.

Todo lo referente a esta página está descrito de manera explícita en este trabajo, el sistema está dividido en tres módulos que permiten automatizar los procesos administrativos y de consulta en bibliotecas, además de el proceso de elaboración de la página Web.

Este sistema está basado en módulos originalmente desarrollados en SERBIULA y el Departamento de Computación de la Escuela de Sistemas de la Facultad de Ingeniería de La Universidad de Los Andes, bajo el sistema SIDULA y SIDULA 2000.

Usando las bondades de software libre se permite la descarga e instalación de estos módulos.

Este proyecto fue desarrollado aplicando el modelo de proceso “Reloj”, y el lenguaje de modelado UML (Unified Modeling Language).

---

**Palabras Claves:** Automatización de bibliotecas, estadísticas, base de datos, servicios bibliotecarios, software libre, gestión de bibliotecas.

# INDICE

---

Resumen	ii
Índice	iii
Índice de figuras	vi
Índice de tablas	viii
<b>Capítulo 1: INTRODUCCIÓN</b>	<b>1</b>
1 Introducción	2
1.1 Antecedentes	4
1.2 Definición del Problema	6
1.3 Objetivos	6
1.3.1 Objetivo General	6
1.3.2 Objetivos Específicos	6
1.4 Método	7
1.5 Alcance del proyecto	8
<b>Capítulo 2: REVISION BIBLIOGRAFICA</b>	
2.1 Sidula	9
2.2 ¿Qué es Serbiula?	10
2.2.1 Serbiula automatiza sus Bibliotecas	11
2.2.2 Misión	12
2.2.3 Objetivo	12
2.2.4 Funciones	12
2.2.5 Organización	13
2.2.5.1 Servicios bibliotecarios existentes en la Universidad de los Andes	16
2.3 Historia del Cidiat	17
2.5 Módulos del Sistema	19
2.5.1 Sistema Automatizado de Adquisiciones Bibliográficas	19
2.5.2 Sistema Automatizado de Recuperación de Información(SARI)	20
2.5.3 Sistema de Edición y Catalogación(EDICLA)	20

2.6	Modelos	21
2.7	El Lenguaje Unificado de Modelado (UML)	21
2.7.1	Una perspectiva general de UML	23
2.7.2	Casos de Uso y Diagramas de Iteración	23
2.7.3	Clases y Diagramas de Implementación	24
2.7.4	Diagramas de Estado	24
2.7.5	Implementando el diseño UML	24
2.7.6	Implementando la aplicación UML	25
2.7.7	Empleo del UML notación	25
2.8	¿Porqué software libre?	26
2.9	Modelo del Proceso “Reloj” para el desarrollo de aplicaciones de software	30
2.9.1	Procesos Gerenciales	31
2.9.2	Proceso de Desarrollo	31
2.9.3	Proceso de Post-desarrollo	32
2.10	Herramientas Utilizadas	34

### **Capítulo 3: MARCO METODOLÓGICO**

3.1	Procesos del Modelo “Reloj”	37
3.1.1	Procesos Gerenciales	37
3.1.2	Procesos de Desarrollo	39
3.1.3	Procesos de Post-desarrollo	39
3.1.4	Análisis del Dominio de Aplicación	39
3.1.4.1	Definición del Dominio	40
3.1.4.2	Identificación y Modelaje de los procesos de dominio	41
3.1.4.2.1	Proceso de selección del Software a usar	43
3.1.4.2.2	Proceso de diseño de la Página Web	43
3.1.4.2.3	Proceso de Inserción de contenidos dentro de cada página	45
3.1.4.3	Identificación y descripción de los Actores	47
3.1.4.4	Identificación y Modelado de las Entidades del dominio	48
3.1.4.5	Identificación y descripción de los Eventos	48
3.1.5	Definición de Requerimientos	49

3.1.5.1	Requerimientos Funcionales	49
3.1.5.2	Requerimientos no Funcionales	55
3.1.5.3	Requerimientos de Interfaz de Usuario	55
3.1.5.4	Requerimientos de Desarrollo y Operación	55
3.1.5.5	Perfil de seguridad y acceso que tienen los usuarios	57
3.1.5.6	Identificación de atributos de Calidad	57
3.1.6	Especificación y análisis de los requerimientos	57
3.1.6.1	Derivación del Modelo de Objetos	61

#### **Capítulo4: IMPLEMENTACIÓN, PRUEBAS Y RESULTADOS**

4	Presentación y etapas del Capitulo 4	65
4.1	Requerimientos de Hardware y Software	67
4.2	Diseño de la Página Web	68
4.3	Creación de la Base de Datos	69
4.4	Proceso de Instalación de los Módulos en mi Máquina	80
4.4.1	Módulo de Adquisiciones	80
4.4.2	Módulo Sari	80
4.4.3	Módulo Win-edicla	81
4.5	Pruebas y Resultados	81
4.6	Empaquetamientos de los Módulos	84
4.6.1	Módulo de adquisiciones	85
4.6.2	Módulo Sari	85
4.6.3	Módulo Win-edicla	85

#### **Capitulo 5: CONCLUSIONES Y RECOMENDACIONES**

5.1	Conclusiones	86
5.2	Recomendaciones	86
	<b>Referencias Bibliográficas</b>	<b>87</b>

## Índice de figuras

---

	Pág.
Figura 2.1 Organigrama de SERBIULA	15
Figura 2.2 Diagrama de proceso “Reloj”	30
Figura 3.1 Esquema de la Fase I	40
Figura 3.2 Diagrama de actividades del proceso general del sistema	43
Figura 3.3 Diagrama de actividades para el diseño y entrega de la página Web.	44
Figura 3.4 Funciones del usuario	49
Figura 3.5 Diagrama de actividades del módulo usuario	52
Figura 3.6 Diagrama de actividades del módulo descarga	53
Figura 3.7 Diagrama de actividades del módulo sugerencias	54
Figura 3.8 Diagrama de caso de uso general del sistema	58
Figura 3.9 Diagrama de caso de uso del usuario	59
Figura 3.10 Diagrama de caso de uso del usuario-administrador	60
Figura 3.11 Diagrama de clase de la base de datos	62
Figura 3.12 Diagrama de Relación de clases	63
Figura 3.13 Diagrama de Despliegue	64
Figura 4.1 Modelo esquemático del diseño de la página	68
Figura 4.2 Interfaz diseñada para la página principal	70
Figura 4.3 Interfaz diseñada para el registro de nuevos usuarios	71
Figura 4.4 Interfaz diseñada para la presentación de manuales	72
Figura 4.5 Pantalla diseñada para la presentación de Descargas	73
Figura 4.6 Pantalla diseñada para la presentación de sugerencias	74
Figura 4.7 Pantalla diseñada para la presentación del módulo adquisiciones	75
Figura 4.8 Pantalla diseñada para la presentación del módulo edicla	75
Figura 4.9 Pantalla diseñada para la presentación del módulo Sari	76
Figura 4.10 Pantalla diseñada para la presentación de los desarrolladores	77
Figura 4.11 Descarga de Módulos	78
Figura 4.12 Usuarios Registrados	79
Figura 4.13 Sugerencias hechas	79
Figura 4.14 Usuario Registrado	82

Figura 4.15	Error Campo vacío	82
Figura 4.16	Error al introducir el password	82
Figura 4.17	Consulta usuarios registrados	83
Figura 4.18	Consulta de descargas	83
Figura 4.19	Consulta sugerencias	84

## Índice de tablas

---

	<b>Pág.</b>	
Tabla 3.1	Tabla de Actores del Sistema	47
Tabla 3.2	Especificaciones del diseño del sistema	56
Tabla 3.3	Casos de uso del Sistema junto con su actor	58
Tabla 4.1	Requerimientos de hardware y software	67
Tabla 4.2	Módulos empaquetados	84

## **CAPITULO 1**

### **INTRODUCCION**

Las demandas actuales y futuras de búsqueda de la información alrededor del mundo, ocupan la atención de los organismos que tienen como una de sus responsabilidades el hallar soluciones adecuadas a la satisfacción de esas demandas. En lo que respecta a las Bibliotecas y Centros de Documentación e Información, están en desarrollo diversas investigaciones orientadas a la utilización más provechosa de los recursos disponibles.

Las bibliotecas y sistemas de información se han visto sometidas a la influencia del rápido desarrollo de los sistemas de comunicación y la creciente posibilidad de presentación de los contenidos en forma digital, adecuando para ello sistemas automatizados en la gestión de dichas bibliotecas.

Avances de importancia se han logrado para el establecimiento de técnicas de optimización y mejor aprovechamiento de estos recursos, partiendo siempre de las demandas de cada usuario.

En un sentido amplio, las bibliotecas son espacios de desarrollo de conocimiento, en el que los usuarios tienen acceso a extensas colecciones de información y a una gama de servicios que les permiten colaborar con otros usuarios y aprovecharlas para realizar actividades de investigación, enseñanza y aprendizaje. Su propósito es poner la información y sus recursos automatizados disponibles para todos en la forma en que cada quien la necesite. Como apoyo a la educación, estas actividades de generación y difusión de conocimientos pueden llegar a tener un impacto determinante sobre el nivel de información obtenida por los estudiantes que como usuarios se benefician de los servicios bibliotecarios.

Es así como en la Universidad de Los Andes se ha venido fortaleciendo durante muchos años, y dando inicio a un Sistema Automatizado de recuperación, adquisición, préstamo y catalogación de la información, que

podiera ser útil y provechoso a otras bibliotecas de la región y en general a otros Centros Académicos del país.

Este trabajo se usará como base conceptual para la automatización de las bibliotecas que comienza con el propósito de promover mayor dinamismo en aquellos procesos que inciden en el tiempo del usuario en el momento de búsqueda de la información así como en los procesos internos de la administración, persiguiendo con esto, dar mayor fluidez al aspecto administrativo, otorgándole a la Biblioteca un carácter moderno, funcional, dinámico y novedoso. Podemos decir que este proceso fortalecerá el crecimiento tecnológico de las instituciones.

El cambio en la forma que se percibe la realidad, se hace palpable al considerar fundamentales para la vida diaria, herramientas de trabajo tales como la Internet. Igualmente, a quienes les ha tocado vivir este tiempo les corresponde ser testigos y actores de los cambios en la forma en que se representa, almacena y aprovecha el conocimiento.

El propósito fundamental de este trabajo de investigación es brindar en forma gratuita herramientas imprescindibles en el apoyo automatizado que las bibliotecas de hoy en día necesitan de manera urgente, aprovechando el uso de software libre disponible en la Internet. Así mismo esperando de los usuarios de este sistema sus aportes y mejoras para el fortalecimiento de este sistema.

## **1.1 Antecedentes.**

A comienzos del siglo XX la Biblioteca de la Universidad se conoce con el nombre de Biblioteca Central teniendo entre sus funciones la organización de las bibliotecas de las diferentes facultades a medida que se fueron creando, así como la formación del personal que trabajaría en ellas.

Las adquisiciones de los materiales se centralizaban a través de la Biblioteca Central, pero los presupuestos para su funcionamiento dependían directamente de las diferentes facultades.

Nace SERBIULA

El 14 de febrero de 1980 se presenta ante el Consejo Universitario de la Universidad de los Andes el "Proyecto de Propuesta para el Funcionamiento de los Servicios Bibliotecarios de La Universidad de los Andes", con el objetivo de establecer y dirigir coordinadamente el Sistema Bibliotecario, así como el de tener un mejor aprovechamiento de las colecciones y de los recursos humanos, presupuestarios y técnicos. Este proyecto se aprueba en Consejo Universitario el 06 de marzo del mismo año. Se integran así los Servicios Bibliotecarios de la Universidad de Los Andes con el nombre de "SERBIULA", bajo la filosofía de descentralización coordinada. Ya en 1977-78 se había iniciado este proceso con la creación de la Biblioteca Integrada de Economía, Ciencias e Ingeniería (BIECI).

La automatización de las bibliotecas comienza con la creación de la BIECI, y del Departamento de Informática, su propósito "...imprimir mayor dinamismo en aquellos procesos que inciden en el tiempo del usuario en el momento de búsqueda de la información así como en los procesos internos de la administración, persiguiendo con esto, dar mayor fluidez al aspecto administrativo, otorgándole a la Biblioteca un carácter moderno, funcional, dinámico y novedoso...".

Este proceso de automatización se inicia por etapas:

- Sistema de control de préstamos.
- Sistema de catálogo de publicaciones periódicas y control de proveedores.
- Sistema de catalogación y recuperación de información (uso de formato Marc).
- Los sistemas automatizados (EDICLA, SARI).

Es así como SERBIULA emprende y desarrolla proyectos que serán la base e inicio de la introducción definitiva de los Servicios Bibliotecarios en el mundo de la teleinformática, acorde con los tiempos actuales, estos son:

- Los sistemas automatizados (EDICLA, SARI)
- Las 14 redes locales de SERBIULA (1977-1978)
- La incorporación masiva de información a la base de datos (1994)
- Indización automática basada en conocimiento

Estos proyectos dieron origen al SIDULA, sistema integrado de gestión de bibliotecas, diseñado desde un principio para ser instalado y utilizado en la Universidad de Los Andes. Es un producto concebido originalmente para manejar la adquisición, catalogación y préstamo de forma centralizada, posteriormente se llegó al concepto de cliente-servidor, aprovechando las bondades de la red de datos de la ULA.

En 1999 se inicio un proceso de reingeniería que llevó a un cambio de enfoque de SIDULA a SIDULA 2000 basados en objetos y en una plataforma de Sistema Operativo Linux.

A principios del 2001 el Departamento de Computación decide construir un nuevo sistema basado en SIDULA pero con un enfoque distinto el cual se basa en la Biblioteca digital virtual para ambientes distribuidos permitiendo este nuevo enfoque descentralizar los procesos y realizar la gestión de las bibliotecas de manera automática vía Web.

## **1.2 Definición del Problema.**

En la actualidad bibliotecas e institutos experimentales carecen de un sistema gestor para los procesos internos de estas, debido al alto costo de los productos comerciales y a su vez el difícil desarrollo de estos sistemas gestores. La Universidad de Los Andes posee un sistema gestor para poner a disposición de bibliotecas e institutos y así contribuir de manera directa al desarrollo de estas.

La entrega de este sistema será bajo la filosofía de software libre, para que todos tengan acceso a las descargas de los módulos del sistema gestor.

## **1.3 Objetivos.**

### **1.3.1 OBJETIVO GENERAL.**

Crear un sitio donde poner a disposición el Sistema Gestor de bibliotecas; para que este sea de dominio público y así pueda ser utilizado por cualquier biblioteca universitaria o institutos experimentales; por tal motivo se debe diseñar e implantar un portal para la descarga del sistema

### **1.3.2 OBJETIVOS ESPECIFICOS.**

Para cumplir con la ejecución del objetivo general, se debe cumplir antes con los siguientes objetivos específicos:

- Búsqueda, recuperación y estudio de cada módulo del Sistema Sidula (Adquisiciones-Sari-Winedicla).
- Instalar, compilar, empaquetar y probar cada uno de los módulos de manera independiente.
- El diseño y elaboración de un portal para la descarga del Sistema
- Elaborar un manual que permita la instalación del Sistema gestor, por un usuario.
- Implementar caso de estudio de la biblioteca del CIDIAT.

## **VENTAJAS.**

- Mayor presencia de la Universidad de Los Andes en el mundo de la información.
- Colaboración permanente de correcciones y mejoras al Sistema Automatizado.
- Desarrollo de nuevos productos. teniendo como ejemplo nuestro Sistema gestor de bibliotecas desarrollado en la Universidad de Los Andes.
- Uso del Software libre, facilidad de acceso, facilidad en la instalación y configuración para la institución que lo requiera.

### **1.4 Método. [1]**

El desarrollo de este trabajo de grado, sigue el método “Reloj” desarrollado por el Profesor Jonás Montilva. Este método esta basado en diferentes grupos de procesos:

- Proceso Gerencial
- Proceso de Desarrollo
- Proceso de Post-desarrollo

Esto significa que el método a seguir posee un enfoque iterativo para el desarrollo de software.

El modelo de proceso reloj posee las características siguientes:

- Puede ser aplicado en el desarrollo de proyectos de software pequeños y medianos.
- Es el resultado de la integración de los modelos de procesos cascada, Modelo V y prototipos, siguiendo el paradigma de orientación por objetos propuesto por Bruegge and Dutoit.
- Está basado en el estándar IEEE 1074 para el desarrollo de procesos de ciclos de vida.

### **1.5 Alcance del Proyecto.**

El Sistema de Gestión de Biblioteca que se implantará, se quiere que este disponible en la red para otras bibliotecas e instituciones académicas, bajo el concepto del software libre y así contribuir de manera directa con el desarrollo del país.

## **CAPITULO 2**

### **REVISION BIBLIOGRAFICA**

La revisión bibliográfica que se presenta en este capítulo, plantea los términos usados de manera general en este trabajo de grado y definiciones muy particulares para la comprensión y desarrollo del mismo. Además se presenta las características de Instituciones misión, visión y departamentos de la Universidad de Los Andes que han participado en el desarrollo de los módulos del Sistema.

La revisión bibliográfica tiene como finalidad recopilar todo aquel material bibliográfico, definiciones y términos afines, que contuviera información sobre el Sistema en estudio, las características de los módulos a utilizar y algunos conceptos y fundamentos importantes sobre tópicos relacionados y vinculados estrechamente con el tema.

#### **2.1 SIDULA**

SIDULA (Sistema de Información y Documentación de la Universidad de Los Andes), sistema integrado de gestión de bibliotecas, desarrollado en lenguaje C, diseñado desde 1980 con la finalidad de ser instalado y utilizado en la Universidad de Los Andes. Es un producto concebido originalmente para manejar la adquisición, catalogación y préstamo de forma centralizada. [2]

El Sistema SIDULA 2000 fue desarrollado en C++ utilizando Sistema Operativo LINUX RedHAD 6.0 con interfaz gráfica basada en páginas Web.[3]

## 2.2 ¿Qué es SERBIULA?

Serbiula (Servicios Bibliotecarios de la Universidad de Los Andes), nace el 14 de febrero de 1980 se presenta ante el Consejo Universitario de la Universidad de Los Andes el *"Proyecto de Propuesta para el Funcionamiento de los Servicios Bibliotecarios de la Universidad de Los Andes"*, con el objetivo de establecer y dirigir coordinadamente el Sistema Bibliotecario, así como el de tener un mejor aprovechamiento de las colecciones y de los recursos humanos, presupuestarios y técnicos. Se integran así los Servicios Bibliotecarios de la Universidad de Los Andes con el nombre de "SERBIULA", bajo la filosofía de descentralización coordinada.

Así se da inicio a la integración de las bibliotecas por áreas del conocimiento, en este caso Ciencia y Tecnología con las colecciones de Economía, Ciencias e Ingeniería y en consecuencia la centralización de algunos Servicios Técnicos para atender las necesidades de estas bibliotecas ahora integradas.

Así comienza la centralización de los Procesos Técnicos y de las Bibliotecas integrantes del proyecto Economía, Ciencias e Ingeniería. Progresivamente, se incorporaron al Departamento de Procesos Técnicos el resto de las bibliotecas componentes del Sistema.

La centralización de las adquisiciones conlleva a la mejor administración de los recursos humanos, técnicos y presupuestarios, ya que los mismos se encuentran dispersos, originando la duplicidad de esfuerzos en horas-hombre en cuanto al tiempo dedicado a las mismas actividades, además de duplicidad de materiales en cada biblioteca. De esta forma las bibliotecas se dedicarán única y exclusivamente a la atención de sus usuarios.

Esta experiencia muestra ventajas positivas respecto al manejo de los recursos humanos, tecnológicos y financieros, dirigiendo esfuerzos hacia la formación del catálogo colectivo, herramienta fundamental para el desarrollo de las colecciones bibliográficas y no-bibliográficas del Sistema Bibliotecario de SERBIULA. [2]

### 2.2.1 SERBIULA automatiza sus Bibliotecas

La automatización de las bibliotecas comienza con la creación de la BIECI (**B**ibliotecas **I**ntegradas de **E**conomía, **C**iencia e **I**ngeniería); actualmente llamada BIACI (**B**ibliotecas **I**ntegradas de **A**rquitectura, **C**iencia e **I**ngeniería), y el Departamento de Informática, su propósito: “imprimir mayor dinamismo en aquellos proceso que inciden en tiempo del usuario en el momento de búsqueda de la información así como en los procesos internos de la administración, persiguiendo con esto, dar mayor fluidez al aspecto administrativo, otorgándole a la Biblioteca un carácter moderno, funcional, dinámico y novedoso”. [2]

Este proceso de automatización se inicia por etapas:

- Sistema de Control de Prestamos.
- Sistema de Catálogo de Publicaciones Periódicas y Control de Proveedores.
- Sistema de Catalogación y Recuperación de Información (uso de formato Marc<sup>1</sup>).

Es así como SERBIULA emprende y desarrolla tres proyectos que serán la base e inicio de la introducción definitiva de los Servicios Bibliotecarios en el mundo de la teleinformática, acorde con los tiempos de esa época, estos son: Los sistemas automatizados, la red local de SERBIULA (1977-1978). La incorporación masiva de información a la base de datos de SERBIULA. (1977).

Estos tres proyectos dieron origen al SIDULA, producto concebido originalmente para manejar la adquisición, catalogación y préstamo de forma centralizada, posteriormente se llego al concepto de cliente-servidor, aprovechando las bondades de la red de los datos de la ULA.

---

<sup>1</sup>Un registro Marc es un registro legible por maquina (Machine-Readable Cataloging). Es un registro bibliográfico, es decir la información que tradicionalmente se presenta en una ficha de biblioteca, donde una computadora en particular puede leer los datos contenidos en un registro catalográfico.

Actualmente SERBIULA desarrolla el proyecto de servicios de información electrónica (SIE), de Biblioteca Digital y de estantería abierta con el objetivo de facilitar y agilizar los servicios que atienden las diferentes demandas de información de la comunidad de usuarios. [2]

### **2.2.2 Misión**

“Eleva los niveles de calidad y efectividad en la generación de servicios y productos de información que constituyen la plataforma de desarrollo de las actividades de docencia, investigación y extensión de la comunidad de usuarios, a través de la optimización permanente de los recursos humanos, tecnológicos de planta física y de presupuesto”. [2]

### **2.2.3 Objetivo**

“Brindar un adecuado y eficiente servicio de consulta y préstamo de material bibliohemerográfico, así como adquirirlo, organizarlo y difundirlo haciendo uso de un sistema teleinformática con tecnología de vanguardia, altamente productivo y competitivo que garantice satisfacer las necesidades de los usuarios con un producto de excelente calidad, en la medida de sus requerimientos”. [2]

### **2.2.4 Funciones [2]**

- SERBIULA desarrolla una serie de funciones encaminadas a conseguir sus objetivos primordiales de apoyo a la formación, la investigación y la docencia.
- Desarrollo de colecciones útiles y de calidad, incluyendo todo tipo de soportes y un gran número de fuentes de información.
- Tratamiento, organización y conservación de la colección.

- Difusión de la colección mediante el préstamo, la consulta, la referencia, los servicios de información bibliográfica o cualquier otro tipo de actividad.
- Formación de los usuarios en el uso de los servicios y fondos bibliotecarios.
- Garantizar el acceso a los fondos a toda la comunidad universitaria.
- Realización de actividades e instrumentos bibliográficos que promuevan el uso del fondo.
- Facilitar el acceso a documentos externos; potenciar actividades de mejora de los servicios.
- Responsabilizarse de la gestión económica y administrativa.
- Mantenimiento de los sistemas automatizados y redes de telecomunicación de que se disponen.
- Mantenimiento de la infraestructura y el equipamiento asignados y adaptación de los mismos según las necesidades.
- Reparto coherente de las cargas de trabajo.
- Administración de las partidas presupuestarias asignadas.
- Establecimiento de criterios y políticas de actuación en la gestión técnica y administrativa.
- Participar en programas y actividades de la propia universidad o ajenos a la misma.
- Contribuir a los objetivos generales de la Universidad y ser útiles para el avance del conocimiento científico y el desarrollo sociocultural.

### **2.2.5 Organización**

Los Servicios bibliotecarios de la Universidad de Los Andes esta conformado por el Vicerrectorado Académico , el cual a su vez esta formado por el Consejo Académico y la Coordinación General que tiene su sede en el Edificio Administrativo, así como también los diferentes servicios bibliotecarios organizado por áreas.

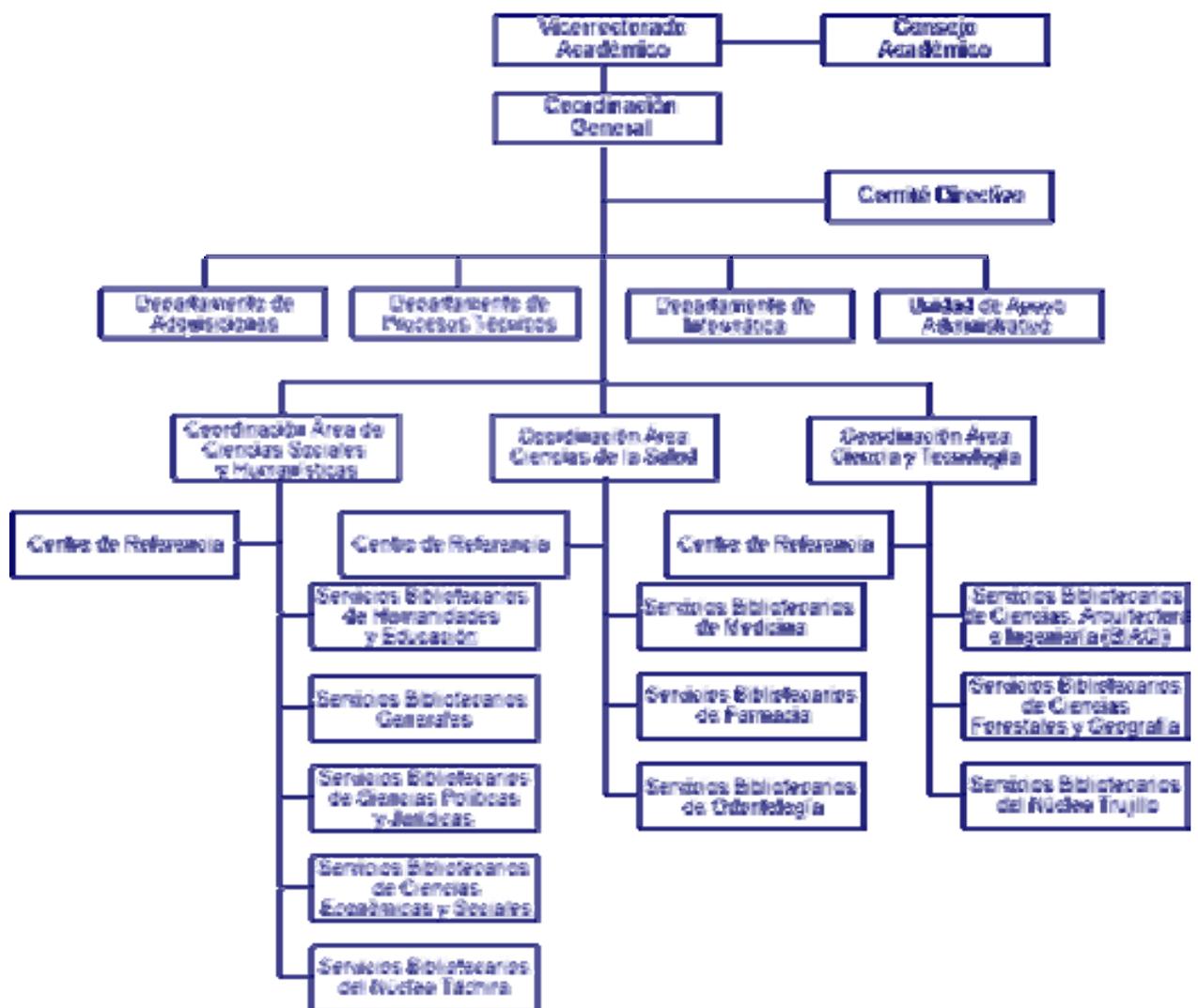
En las Áreas tenemos: Área de Ciencia y Tecnología, Área de Ciencia Sociales y el Área de Ciencias de la Salud, cada uno con su centro de referencia. Estas Áreas están formadas a su vez por diferentes Servicios bibliotecarios. También cuenta con La Unidad Centralizada formada por los Departamentos Centralizados. [2]

**Área de Ciencia y Tecnología:** comprende los Servicios Bibliotecarios de Arquitectura, Economía, Ingeniería, Escuela de Arte, Ciencias, Ciencias Forestales, Geografía. Así como también se realizan compras para el Postgrado de Arquitectura en la ciudad de Mérida y compras para el Núcleo de Trujillo y postgrado del Núcleo de Trujillo.

**Área de Ciencias Sociales:** Formada por la Biblioteca Tulio Febres Cordero, Ciencias Jurídicas y Políticas, Humanidades, Escuela de Idiomas, y Núcleo del Táchira.

**Área de Ciencias de la Salud:** Farmacia, Medicina y Odontología; Hospital Universitario de Los Andes, Hospital Barinas, Hospital Guanare, Hospital Valera, Hospital San Cristóbal, Hospital Valle de la Pascua, Microcopia. Electrónica.

**La Unidad Centralizada:** formada por: Departamento de Adquisiciones, Departamento de Procesos Técnicos, Departamento de Informática y la Unidad de Apoyo administrativo.



**Figura 2.1** Organigrama de SERBIULA.

### 2.2.5.1 Servicios bibliotecarios Existentes en la Universidad de Los Andes.[2]

Los servicios bibliotecarios de La Universidad de Los Andes son doce más el núcleo Táchira y el núcleo Trujillo, que da un total de catorce bibliotecas. Cada

servicio bibliotecario cuenta con una hemeroteca, SERBIULA cuenta además con tres centros de referencia distribuidos por áreas, una mapoteca y la videoteca.

- **Servicios Bibliotecarios**

En cada Facultad, Núcleo o extensión funciona un servicio bibliotecario, donde ofrece atención a toda la comunidad académica.

- **Hemerotecas**

Los Departamentos de Publicaciones Periódicas de cada Servicio Bibliotecario cuentan con políticas de préstamos bibliohemerográficos para los usuarios, haciendo llegar a manos de estos las publicaciones periódicas o revistas adquiridas por el Departamento de Adquisiciones, así como aquellas adquiridas por medio de los mecanismos de canje y donación.

- **Centros de Referencia de los Servicios Bibliotecarios de la Universidad de Los Andes**

Un centro de referencia es una unidad de información que tiene como función básica la de apoyar las actividades de investigación y docencia orientando al usuario en la localización de la información y en el manejo de los recursos existentes. Brinda al usuario atención personalizada para consultar artículos científicos, orientando al usuario en la búsqueda de información accediendo a la consulta de colección impresa.

En la actualidad se cuenta con tres centros de referencia que soportan una colección de consulta en formatos impresos: diccionarios, atlas, enciclopedias, biografías y guías. Además, cuenta con bases de datos en CD nacionales e internacionales y bases de datos en línea. Estos centros son:

- Ciencia y Tecnología.
- Ciencias de la Salud, que cuenta con Guías farmacológicas, Anuarios estadísticos, Censos, entre otros.

- Ciencias Sociales y Humanísticas: cuenta con anuarios, una amplia colección por países de estudio de pregrado y postgrado en las distintas Instituciones educativas del mundo.

- **Mapoteca**

La mapoteca reúne una colección de mapas de variada estirpe, está ubicada en la Facultad de Ciencias Forestales, Geográficas y Ambientales.

- **Videoteca**

La videoteca reúne una colección de videos de material no bibliográfico, esta ubicada en el Edificio Administrativo de la ULA.

## 2.3 HISTORIA DEL CIDIAT[4]

El Cidiat fue creado mediante un acuerdo entre el Gobierno de Venezuela y la Organización de Estados Americanos (OEA), firmado en Washington el 4 de febrero de 1965, y su **Misión** se expresa: “Servir de apoyo en las actividades de postgrado, investigación y extensión desarrolladas por el CIDIAT mediante el suministro de material bibliográfico, hemerográfico, documentos, información referencial de personas, instituciones, cursos y eventos.”

Su **Visión** se manifiesta así: “Brindar servicios de calidad en forma útil, completa y oportuna, buscando utilizar para ello los medios tecnológicos más actualizados que garanticen el uso eficiente de los recursos a disposición.”

La Biblioteca del CIDIAT fue creada al mismo tiempo a sabiendas de que en cualquier organismo de adiestramiento es un servicio imprescindible. Se inicio con acervo bibliográfico de diferentes temas y en la medida en que fue avanzando el Centro con sus cursos, seminarios, talleres y demás actividades, comenzó a ser selectivo con la documentación

bibliográfica que ingresaría a la Biblioteca, de tal manera que cumpliera los requisitos de apoyo a las actividades de enseñanza, investigación y extensión en las áreas de especialidad del CIDIAT, por lo que se estableció que fuera una biblioteca especializada en el campo del desarrollo y la administración de los recursos de aguas y tierras.

En la actualidad el **CIDIAT-ULA** cuenta con una biblioteca que dispone de 10.440 libros, 18.000 revistas científicas (280 títulos), en las diferentes especialidades que enmarcan las áreas prioritarias de actuación del Centro. Así mismo, a través de canje y donaciones, se realiza el intercambio de publicaciones con Instituciones Nacionales e Internacionales.

Además, y con el propósito de cubrir el gran déficit de publicaciones en la materia, en idioma español, se realiza un sustancial esfuerzo por producir documentos resultado de: (i) estudios o investigaciones realizadas en el CIDIAT-ULA por el personal docente y de investigación, y estudiantes de postgrado; (ii) material didáctico preparado por profesores permanentes y visitantes; (iii) estudios, investigaciones, proyectos e informes de diversa índole, producidos fuera del Centro, pero cuya difusión se considere de interés.

## **2.5 Módulos Del Sistema Sidula**

A principios del 2001 el Departamento de Informática de Serbiula, decide construir un nuevo sistema basado en SIDULA pero con un enfoque distinto el cual se basa en la Biblioteca digital virtual para ambientes distribuidos permitiendo este nuevo enfoque descentralizar los procesos y realizar la gestión de las bibliotecas de manera automática vía Web.

De manera similar este trabajo de grado se basa en el Sistema SIDULA y darle una nueva orientación hacia el software libre, empaquetando cada uno de sus módulos y ponerlos a libre disposición de Instituciones académicas vía Web.

- Sistema Automatizado de Adquisiciones bibliográficas.
- Sistema Automatizado de Recuperación de Información. (SARI)
- Sistema de Edición y Catalogación. (EDICLA)

### **2.5.1 Sistema Automatizado de Adquisiciones Bibliográficas**

El Sistema de Adquisiciones permite realizar todas las funciones que se llevan a cabo con la adquisición del material bibliohemerográfico de una manera eficiente. Este sistema atiende las recomendaciones de adquisición de material bibliográfico realizadas por los profesores y/o personal calificado y autorizado, con la finalidad de satisfacer las necesidades de los usuarios de las bibliotecas.

El sistema de adquisiciones controla de manera automática los procesos de solicitud, facturación, pago y reclamo de material bibliohemerográfico, así como también el proceso de recepción e ingreso inicial de la información. El subsistema de adquisición permite configurar las interfaces de interacción con el usuario dependiendo del perfil del mismo y su grado de experticia, de manera que se pueda trabajar con una interfaz de bajo nivel como es la de comando y una de alto nivel basada en Web las cuales ofrecen todas las facilidades para incorporar y recuperar información sobre el material bibliohemerográfico que se encuentre en fase de adquisición o tramitación por parte de los proveedores.

El sistema de adquisiciones tiene por objetivo ofrecer las facilidades para coordinar y supervisar el proceso de adquisición de material bibliohemerográfico y audiovisual solicitado por las unidades de información que conforman el sistema bibliotecario a fin de mantener actualizadas las colecciones y de esta manera evitar duplicidad.[3]

### **2.5.2 Sistema Automatizado de Recuperación de Información (SARI)**

El sistema Automatizado de recuperación de Información permite la búsqueda y recuperación de la información que se encuentra almacenada en diferentes bases de datos que se encuentren distribuidas dentro del campus.

Es un poderoso "motor de búsqueda" que permite, de una manera rápida y eficiente, la recuperación de cualquier documento almacenado en las diferentes bases de datos, mediante el planteamiento de una estrategia de búsqueda, haciendo uso de las palabras claves o descriptores.

Este "motor de búsqueda" ha sido desarrollado por el Departamento de Computación de la Escuela de Ingeniería de Sistemas de la Universidad de Los Andes y está calificado como uno de los más rápidos "motores de búsqueda" a nivel mundial. Este motor posee funcionalidades que permite la elaboración de reportes en distintos formatos (Listas y Fichas), bien sean impresos o por pantalla.[3]

### **2.5.3 Sistema de Edición y Catalogación (EDICLA)**

El Sistema de Edición y Clasificación del Material Bibliohemerográfico y Audiovisual permite el ingreso de la información del material que ingresa a la biblioteca utilizando para ello el Formato Estándar Internacional para Catalogación automatizada e Intercambio de información "MARC".

Provee un editor, con el cual se realiza, de una manera sencilla, el ingreso de la información relacionada con el material bibliohemerográfico y audiovisual que maneja la biblioteca.

Este editor, posee la habilidad de realizar de una manera automática la validación de las Reglas de Catalogación definidas en el estándar "MARC", así como también la asignación automática de la puntuación establecidas por las normas. Este sistema esta basado en una filosofía cliente servidor en la que en el servidor almacena los documentos que forman parte de la biblioteca y el cliente posee una interfaz Windows donde se validan todos los campos del

formato MARC con los que se introducirá la información al servidor una vez que el documento pase por un control de calidad.[3]

## **2.6 Modelos**

Freigenbaum (1992), define los modelos como representaciones esquemáticas de un proceso o conjunto de procesos para describir su secuencia. Estos modelos son utilizados para implementar filosofías y conocer patrones funcionales de operaciones secuenciales; en cambio para David (1998), los modelos son representaciones organizadas a nivel de los procesos para describir la secuencia lógica de las acciones, que fomentan filosofías y fijan patrones funcionales de operaciones secuenciales. Los modelos se pueden utilizar para identificar y estudiar posibles soluciones de los procesos.

## **2.7 El Lenguaje Unificado de Modelado (UML)[5]**

Según Harmon, (1998). UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.

UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso

Harmon (1998). El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos del Sistema.
- 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.
- UML es una consolidación de muchas de las notaciones y conceptos más usadas orientados a objetos.

Empezó como una consolidación del trabajo de Grade Booch, James Rumbaugh, e Ivar Jacobson, creadores de tres de las metodologías orientadas a objetos más populares. UML ofrece notación y semántica estándar UML describe una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos. Previamente, un diseño orientado a objetos podría haber sido modelado con cualquiera de la docena de metodologías populares, causando a los revisores tener que aprender las semánticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí.

### **2.7.1 Una perspectiva general de UML**

Una vez más, UML es una notación, no un método. No describe un proceso para modelar un sistema. No obstante, como UML incluye los diagramas de casos de uso, se le considera estar dotado de una aproximación

al diseño centrada en el problema con los casos de uso. El Diagrama de Caso de Uso da el punto de entrada para analizar los requisitos del sistema, y el problema que se necesita solucionar.

### **2.7.2 Casos de Uso y Diagramas de Interacción**

Un caso de uso se modela para todos los procesos que el sistema debe llevar a cabo. Los procesos se describen dentro del caso de uso por una descripción textual o una secuencia de pasos ejecutados. Los Diagramas de Actividad se pueden usar también para modelar escenarios gráficamente. Una vez que el comportamiento del sistema está captado de esta manera, los casos de uso se examinan y amplían para mostrar qué objetos se interrelacionan para que ocurra este comportamiento. Los Diagramas de Colaboración y de Secuencia se usan para mostrar las relaciones entre los objetos.

### **2.7.3 Clases y Diagramas de Implementación**

Conforme se van encontrando los objetos, pueden ser agrupados por tipo y clasificados en un Diagrama de Clase. Es el diagrama de clase el que se convierte en el diagrama central del análisis del diseño orientado a objetos, y el que muestra la estructura estática del sistema. El diagrama de clase puede ser dividido en capas: aplicación, y datos, las cuales muestran las clases que intervienen con la interfaz de usuario, la lógica del software de la aplicación, y el almacenamiento de datos respectivamente. Los Diagramas de Componentes se usan para agrupar clases en componentes o módulos. La distribución general del hardware del sistema se modela usando el Diagrama de Implementación.

### **2.7.4 Diagramas de Estado**

El comportamiento en tiempo real de cada clase que tiene comportamiento dinámico y significativo, se modela usando un Diagrama de

Estado. El diagrama de actividad puede ser usado también aquí, esta vez como una extensión del diagrama de estado, para mostrar los detalles de las acciones llevadas a cabo por los objetos en respuesta a eventos internos. El diagrama de actividad se puede usar también para representar gráficamente las acciones de métodos de clases.

### **2.7.5 Implementando el diseño UML**

La implementación del sistema trata de traducir información desde múltiples modelos UML en código y estructura de bases de datos. Cuando se modela un sistema grande, es útil fragmentar el sistema (incluyendo los objetos de la interfaz de usuario), su capa de aplicación (incluyendo los objetos de implementación), y su capa de datos (incluyendo la estructura de la base de datos y el acceso a objetos).

### **2.7.6 Implementando la aplicación UML**

El Diagrama de Clase se usa para generar una estructura base del código en el lenguaje escogido. Información de los diagramas de interacción, estado, y actividad, puede ofrecer detalles de la parte procedimental del código de implementación.

### **2.7.7 Empleo del UML Notación.**

A través de la notación UML podemos comunicar y compartir el conocimiento de una arquitectura gracias a la combinación simultánea de cinco perspectivas:

- **Definir.**- Fijar, determinar, decidir, explicar un concepto a través de sus atributos distintivos. Señalar sus límites y dar una idea exacta de lo que es esencial y de lo que es circunstancial.
- **Organizar.**- Establecer unos recursos, disponer un orden de responsabilidades y formalizar unas reglas de relación y actuación; todo ello orientado a conseguir un propósito.

- **Visualizar.**- Representar mediante imágenes y/o símbolos el contenido y la organización de los conceptos que configuran un sistema. Hacer visible su naturaleza y su complejidad.
- **Actuar.**- Pensar y tomar decisiones de manera ágil y sistemática, siguiendo un método; éste a su vez, define el modo de actuar en base a la relación de un conjunto de actores, actividades, entregables y certificaciones posibles en un escenario concreto.
- **Certificar.**- Comprobar de manera fehaciente que un entregable es completo, coherente y usable para el propósito que ha sido creado.

El resultado, es una mayor comprensión y claridad sobre la naturaleza de los objetos, eventos y hechos que tienen consecuencias dentro de un dominio.

## 2.8 ¿Porqué software libre?

(Álvarez, 2001) Durante años se ha prolongado el debate entre los usuarios de sistemas operativos y software en general acerca de si los productos de fuentes abiertas (Open Source Software) son más seguros que el software propietario cuyo código fuente permanece celosamente guardado.

La diferencia entre ambos tipos de programas es tajante: en el primer caso, todos los listados de código fuente están disponibles junto con la distribución del producto, de manera que cualquiera pueda examinarlos y modificarlos si lo juzga conveniente. En este punto es importante realizar la distinción entre software libre y software de fuentes abiertas. Aunque a menudo confundidos, se trata de conceptos distintos.

El software libre es aquel que se distribuye de forma gratuita, sin costo alguno. Pero gratis no significa que venga acompañado de las fuentes.

Por su parte, el software de fuentes abiertas (OSS), además de distribuirse gratuitamente, viene acompañado de su código fuente, lo que permite modificarlo, reutilizar su código en otros programas y redistribuirlo libremente. Se le suele llamar también 'software libre', pero la confusión con la palabra

'gratis' ('free' también en inglés) ha motivado el que normalmente se abandone ese adjetivo en beneficio de 'fuentes abiertas'.

Algunos ejemplos destacados y de sobra conocidos de OSS son los sistemas operativos Linux, FreeBSD y OpenBSD, el lenguaje de programación Perl, los escritorios Gnome y KDE, el navegador Netscape 6, el servidor Web Apache y la base de datos mySQL.

Los defensores del movimiento de fuentes abiertas aducen que el hecho de que investigadores y programadores de todo el mundo puedan examinar las fuentes de los programas acelerará el proceso de detección de agujeros de seguridad y permitirá producir un software más seguro y robusto. La seguridad de un sistema debe fundamentarse en la seguridad de sus algoritmos, protocolos y codificación, no en mantener en secreto su funcionamiento. Cuanto mayor sea el número de gente que pueda revisarlo y durante más años, mayor será la probabilidad de descubrir errores y mejorar el código.

La lista de ejemplos de software que basando su seguridad en la oscuridad han fracasado es francamente extensa: CSS, el sistema de protección de las películas en DVD; los sistemas de cifrado y autenticación de la telefonía móvil GSM; el protocolo de redes privadas virtuales de Microsoft, PPTP; etc.

Por otro lado, los detractores del OSS sostienen que el mismo hecho de que cualquiera pueda examinar el código fuente posibilita a los hackers para que ellos también descubran fallos y los exploten en su propio provecho para atacar al sistema.

La falacia de este argumento reside en la asunción implícita de que los hackers no pueden descubrir fallos en el software comercial de código de fuentes cerradas o no pueden practicar la retroingeniería con él.

Sin embargo, basta con darse una vuelta por sitios como BugTraq o NTBugTraq para comprobar que en sistemas operativos como Windows NT/2000 o Solaris y en productos desarrollados para ellos, todos de fuentes cerradas, se descubren tantos agujeros de seguridad o más que en los de fuentes abiertas. También es verdad que no todos los usuarios disponen de la capacitación técnica ni del tiempo para examinar el kernel de un sistema operativo en busca de fallos o para evaluar la seguridad de un sistema de

seguridad como PGP. Sin embargo, lo verdaderamente fundamental en este movimiento de apertura y cooperación consiste precisamente en el hecho de ofrecer esa posibilidad, esa libertad, en dejar las fuentes disponibles para el escrutinio público.

En esta situación no cabe preguntarse cómo es posible que el software gratuito y abierto continúe prosperando, sino más bien cómo es posible que todavía se siga comprando y utilizando productos y sistemas operativos completamente cerrados y oscuros (u oscurecidos).

“Software Libre” se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias. (libertad 2)
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

(Miguel Vidal-2005) Pese a la retórica de la «defensa del autor», las cuestiones relacionadas con la llamada propiedad intelectual se abordan desde la óptica de maximizar el beneficio empresarial como criterio exclusivo. Es así como se sustenta el modelo propietario, es decir, con la idea incuestionable de que todo programa informático tiene un dueño, generalmente una compañía asociada a su desarrollo o producción. Visto así, parece claro que el modelo industrial vigente requiere cada vez más proteccionismo político, económico, legal y técnico, para garantizar la propiedad privada del software y el beneficio privado, incluso a costa de cercenar innumerables libertades básicas de los usuarios. Este es el discurso al que estamos acostumbrados a escuchar, la defensa de la propiedad intelectual a cualquier precio, por encima de cualquier otro derecho,

y que solo atiende a los intereses privados de los propietarios de los derechos (que rara vez son los propios actores).

El proceso de creación del software libre – contrariamente al software propietario escapa al modelo basado en el marketing, basándose más bien en la noción de utilidad social.

La capacidad de los autores de software libre para recurrir a la comunidad tanto para proponer mejoras como para reforzar el equipo de desarrollo es un factor determinante.

Hay otros retos pendientes, en forma de amenazas, que no deben olvidarse, como es en primer lugar la lucha contra las patentes de programación, las cuales afectan especialmente (aunque no exclusivamente) al uso y desarrollo del software libre. También, es importante defenderse de la cruzada judicial contra el software libre que han intentado de forma insidiosa (y sin base jurídica) corporaciones como SCO o las maniobras menos descaradas (pero igualmente dañinas), en forma de campañas de confusión e intoxicación de la opinión pública que periódicamente promueven empresas como Microsoft. Por último, pero no menos importante, son un reto todas aquellas cuestiones relacionadas con el verdadero trasfondo político del software libre, que va mucho más allá de la informática, y que consiste en promover la libertad y la extensión del procomún, es decir, de los commons o bienes comunes, como base para la difusión de las ideas, del acceso al conocimiento, al florecimiento del arte y de la ciencia, el software crece mejor en libertad y se construye de forma transparente sobre el trabajo hecho con anterioridad. Tal y como sería inconcebible una sociedad libre sustentada en códigos y leyes secretas, es inconcebible una “sociedad de la información” libre sobre código secreto. Una sociedad libre y democrática que merezca tal nombre es aquella que garantiza que sus recursos más importantes permanecen libres y transparentes para la ciudadanía, sean estos libros, leyes o software.

Con todos estos argumentos expuestos anteriormente, vemos y entendemos como hoy en día todo va tomando esta tendencia hacia el “Software Libre” y así

este trabajo de grado va orientado a exponer el Sistema gestor de Bibliotecas “Nikolaus Walczuch” a través de un portal Web.

## 2.9 Modelo del proceso “Reloj” para el desarrollo de aplicaciones de software.[1]

En este trabajo de grado se utilizó el método “reloj”, desarrollado por Jonás Montilva, profesor de la Escuela de Sistemas en la Facultad de Ingeniería en la Universidad de Los Andes. Muestra un método a seguir en el desarrollo de aplicaciones de *software* pequeñas o medianas, y está basado en diferentes grupos de procesos, adaptados del estándar IEEE 1074, entre los que se encuentran: Ingeniería de Métodos, Procesos Gerenciales y de Integración, Proceso de Desarrollo y Proceso de Post-desarrollo [Montilva 2000].

En la figura 2.2 se muestra el diagrama del método “Reloj” a seguir en este proyecto.

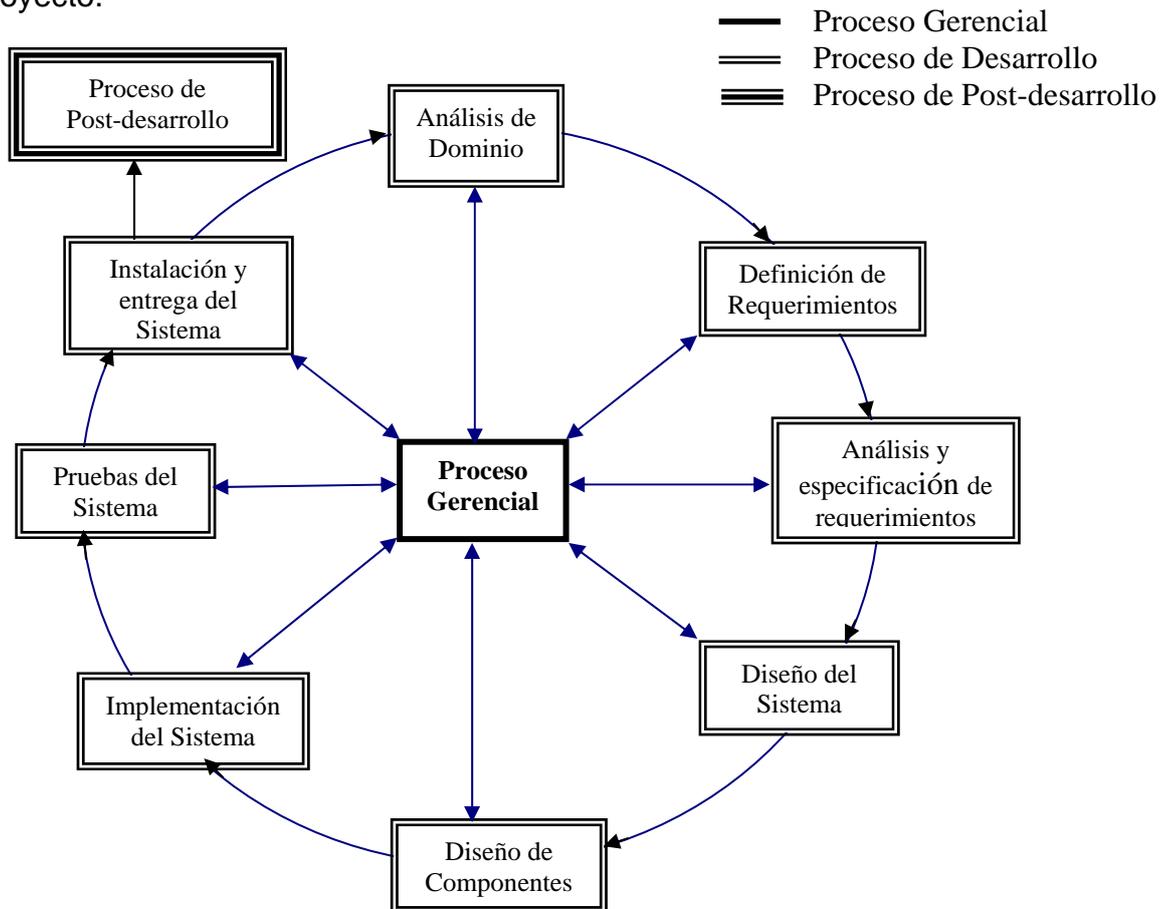


Fig. 2.2 Diagrama del Proceso “Reloj”

En este diagrama se muestra los diferentes procesos que conforman el método “Reloj”, comenzando por el proceso gerencial, y luego avanzando a partir de la fase de “Análisis de Domino” en sentido de las agujas del reloj pasando por las diferentes fases del proceso de desarrollo, pudiendo volver al proceso gerencial y/o a otra fase, hasta alcanzar el punto final del diagrama en el proceso de post-desarrollo.

Debido a que cada proceso tiene características propias, a continuación se describen cada uno de los procesos:

### 2.9.1 Procesos gerenciales

Estos procesos se encargan principalmente de la planificación y control del proyecto a realizar y del equipo de trabajo que lo elaborará, los diferentes procesos son:

- **Administración del proyecto:** consiste en la planificación y control del proyecto, y la organización y dirección del equipo de trabajo.
- **Administración de la calidad del *software*:** consiste en la planificación y aseguramiento de la calidad del *software*.
- **Administración de la configuración del *software*:** consiste en la planificación de la administración y control de la configuración del *software*.
- **Verificación y validación:** consiste en la verificación, validación, revisión y pruebas del *software*.
- **Entrenamiento:** consiste en la planificación y realización del entrenamiento del grupo de trabajo, junto con la producción del material de entrenamiento.
- **Documentación:** consiste en la planificación, producción y entrega de la documentación.

### 2.9.2 Proceso de desarrollo

Este proceso consiste en la realización de diversas actividades o fases técnicas para el desarrollo del *software*, estas fases son:

- **Análisis del dominio de la aplicación:** consiste en comprender el dominio de la aplicación antes de proseguir con las fases de definición y especificación de requerimientos.

- **Descubrimiento y definición de requerimientos:** consiste en descubrir y definir los requerimientos que el sistema debe satisfacer al usuario para luego obtener el prototipo de la interfaz de usuario.
- **Análisis y especificación de requerimientos:** consiste en la elaboración de los requerimientos del usuario para que sea entendido por los diseñadores del sistema.
- **Diseño del sistema:** consiste en traducir los requerimientos en una solución.
- **Diseño de componentes:** consiste en especificar detalladamente el diseño de cada componente y las conexiones presentes en la arquitectura del sistema.
- **Implementación del sistema:** consiste en traducir las especificaciones de diseño en un programa explícito de *software*.
- **Pruebas del sistema:** consiste en realizar pruebas funcionales, de rendimiento y de aceptación.
- **Entrega del sistema:** Consiste en probar la instalación del sistema, se entrena a los usuarios, y se distribuye la documentación.

### 2.9.3 Procesos de Post-desarrollo

Este proceso consiste en la realización del mantenimiento y soporte al sistema implementado.

## 2.10 Herramientas Utilizadas

- **(PHP)** HyperText Preprocessor tiene su origen a principios de 1995 en un conjunto de macros conocido como personal Home Page Tools, su creador fue Rasmus Lerdof. Es un lenguaje de script, que entre sus principales características se pueden destacar su potencia, alto rendimiento y su facilidad de aprendizaje, además de poder añadir dinamismo a las páginas Web, por tal razón, es una herramienta de desarrollo para los programadores Web, ya que proporciona elementos que permiten generar de manera rápida y sencilla sitios Web dinámicos. Es un lenguaje de programación que contiene muchos conceptos de C, Perl y Java. El código PHP está embebido en documentos HTML.[6]

- **PostgreSQL**

PostgreSQL es un sistema manejador de Base de Datos (SMBD) objetos- relacionales. Nace con este nombre en 1996. Originalmente desarrollado en la Universidad de California en Berkeley. Ofrece soporte al lenguaje SQL 92/SQL3, integridad de transacciones, y extensibilidad de tipos de Datos. Es un descendiente de Dominio Público. [7]

- **Base de Datos [8]**

Una base de datos es un conjunto, colección o depósito de datos almacenados en un soporte informático de acceso directo. Los datos deben estar interrelacionados estructurados.

Dada la importancia que tienen en el mundo real las interrelaciones entre los datos, es imprescindible que la base de datos sea capaz de almacenar éstas interrelaciones, al igual que hace con otros elementos, como las entidades y atributos.

La actualización y recuperación en las bases de datos debe realizarse mediante procesos bien determinados, incluidos en un conjunto de programas que se encargan de la gestión de la base de datos y que se denominan sistemas gestores de bases de datos **(S.G.B.D)**; procedimientos que han de

estar diseñados de modo que se mantenga la integridad, seguridad y confidencialidad de la base.

El concepto de base de datos ha ido cambiando y configurándose a lo largo del tiempo, en la actualidad, y de acuerdo con estas características que acabamos de analizar, podemos definir la base de datos como:

“Colección o depósito de datos integrados, con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción, única para cada tipo de datos, han de estar almacenadas junto con los mismos. Los procedimientos de actualización y recuperación comunes y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de los datos.”[8]

- **Base de Datos Relacional[9]**

En una computadora existen diferentes formas de almacenar información. Esto da lugar a distintos modelos de organización de la base de datos: jerárquico, red, relacional y orientada a objeto.

Los sistemas relacionales son importantes porque ofrecen muchos tipos de procesos de datos, como: simplicidad y generalidad, facilidad de uso para el usuario final, períodos cortos de aprendizaje y las consultas de información se especifican de forma sencilla.

Las tablas son un medio de representar la información de una forma más compacta y es posible acceder a la información contenida en dos o más tablas.

Las bases de datos relacionales están constituidas por una o más tablas que contienen los datos ordenados de una forma organizada. Cumplen las siguientes leyes básicas:

- Generalmente, contendrán muchas tablas.
- Una tabla solo contiene un número fijo de campos.
- El nombre de los campos de una tabla es distinto.
- Cada registro de la tabla es único.
- El orden de los registros y de los campos no está determinado.
- Para cada campo existe un conjunto de valores posibles.

- **Campo**

Unidad básica de una base de datos. Generalmente un campo puede ser, por ejemplo datos de una persona. Los nombres de los campos, no pueden empezar con espacios en blanco y caracteres especiales. La descripción de un campo, permite aclarar la información referida a los nombres del campo. El tipo de campo, permite especificar el tipo de dato que cargaremos en dicho campo, esta puede ser:

- Texto: para introducir cadenas de caracteres hasta un máximo de 255.
- Memo: para introducir texto extenso hasta 65.535 caracteres.
- Numérico: para introducir números.
- Fecha-hora: para introducir datos en formato fecha y hora.
- Moneda: para introducir datos en formato número y con el signo monetario
- Lógico: Este tipo de campo es solo para contenidos del tipo Si-No, verdadero o falso, etc.
- Hipervínculo: podemos definir un enlace a un página Web.

- **Campo Clave**

Es el campo que permite identificar y localizar un registro de manera ágil y organizada

- **Registro**

Es el conjunto de información referida a una misma persona u objeto.

Un Sistema de base de datos incluye dos (2) tipos de programas:

- **El software de propósito general**, para la gestión de la base de datos, comúnmente llamado Sistema Gestor de Bases de Datos (S.G.B.D), el cual maneja todas las solicitudes de acceso a la base de datos formuladas por los usuarios y los programas de aplicación.

- **El software de aplicación**, que usa las facilidades del S.G.B.D para manipular la base de datos con el fin de llevar a cabo una función específica en la gestión de la empresa (por ejemplo: la gestión de almacenar). Puede estar desarrollado en un lenguaje de programación estándar, tal como C++, o en un lenguaje propio de los S.G.B.D denominados lenguajes de cuarta generación (4GL).

## **CAPITULO 3**

### **MARCO METODOLÓGICO**

#### **Selección de un Método existente para la Creación y Desarrollo de la Página Web.**

En este capítulo se enumeran las actividades desarrolladas para el logro del objetivo a través del método “Reloj” ya expuesto en el capítulo anterior y posteriormente se desarrollan cada una de estas actividades.

#### **3.1 Modelo del Proceso “Reloj”**

A continuación pasamos a describir de manera detallada los procesos basados en el método “Reloj”, proceso gerencial, proceso de desarrollo y proceso de post-desarrollo.

##### **3.1.1 Procesos gerenciales**

Se determina el equipo de trabajo, administración del proyecto, administración de la calidad del software, administración de la configuración del software, verificación y validación para probar el sistema con la finalidad de verificar la integridad de los datos y los requerimientos exigidos.

- **Administración del Proyecto:**

Consiste en la planificación, el control, la organización y dirección del proyecto. En este proyecto de grado el grupo de trabajo está formado por el Prof. Domingo Hernández (Tutor Académico), Prof. Dulce Rivero (cotutor), Ing. René Izarra y la Ing. Tania Rodríguez en la administración y la calidad del software y, en la parte de diseño del sistema, Carlos Moreno tesista de este proyecto.

- **Administración de la Calidad del Software:**

Consiste en la planificación y aseguramiento de la calidad del software. Selección del software a usar y requerimientos mínimos del hardware para montar la plataforma.

Revisión del sistema que se quiere montar, para ver si cumple con las sugerencias presentadas, comprobar los requisitos del software y hardware necesarios para montar el sistema, probar cada uno de los módulos por separado para comprobar su funcionamiento.

- **Administración de la configuración del Software:**

Revisar que el software que se propone, corresponda a las características exigidas en el sistema; una vez revisado los requisitos de software, comprobar si es posible ejecutar cada módulo sobre el nuevo sistema operativo.

En cuanto a los requerimientos del software, se escogió inicialmente Linux-Fedora para finalmente probar también con Ubuntu versión 5.10 donde qued el sistema definitivamente, es hacer notar que el sistema funcionó con ambos sistemas operativos luego de corregir algunos errores, la razón de cambio argumentada para cambiar el sistema operativo es simplemente probar con por lo menos dos (2) versiones distintas como son Fedora y Ubuntu.

- **Verificación y Validación:**

Probar en su totalidad y en conjunto el funcionamiento de los tres módulos propuestos para el sistema gestor de bibliotecas y de esa manera hacer la comprobación sobre el sistema operativo que se montó.

- **Entrenamiento:**

Sobre el adiestramiento a usuarios que vayan a descargar el software para su institución, se elaborará pasos de instalación a seguir en la configuración e instalación del software, que estarán disponibles en la página Web.

- **Documentación:**

Documentar desde la instalación misma del sistema operativo, hasta la instalación completa de cada uno de los módulos.

Elaborar la página haciendo los vínculos pertinentes que permitan: la suscripción al sistema y genere una clave al usuario, además de permitirle la descarga de los módulos por separado, elaborar los manuales de ayuda para la descarga del sistema y que lleve la estadística de quien lo use y el módulo que descargo.

### **3.1.2 Procesos de desarrollo**

El proceso de desarrollo comienza con el análisis del dominio de la aplicación, identificación y análisis del problema, identificación de los procesos del negocio involucrados, identificación de unidades organizacionales involucradas en la ejecución de los procesos, así como la definición de requerimientos, requerimientos funcionales, y no funcionales, el diseño del sistema como parte final.

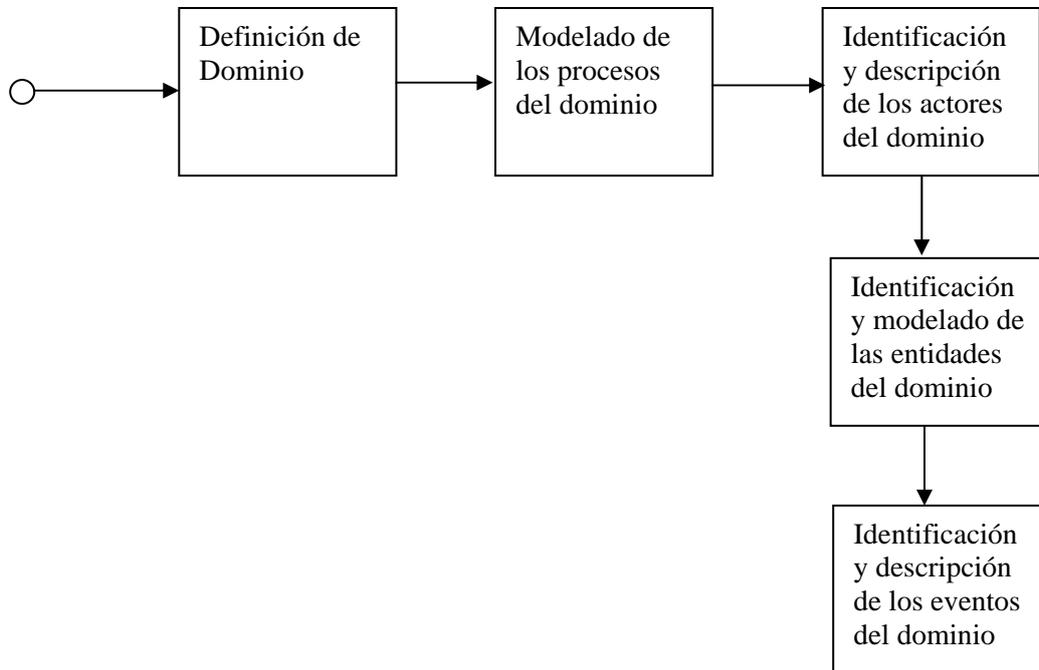
### **3.1.3 Procesos de post-desarrollo**

Una vez implantado el sistema. En este proceso corresponde comprobar el correcto funcionamiento del sistema para detectar posibles errores en el manejo del mismo, prestar operación de soporte y mantenimiento, el cual corresponderá al Departamento de Computación de la Facultad de Ingeniería, Escuela de Sistemas de la Universidad de los Andes.

### **3.1.4 Análisis del Dominio de Aplicación**

En esta fase se estudia el sistema organizacional, es decir, las actividades y los procesos que debemos seguir para el logro de los objetivos planteados en el capítulo 1, se definen los actores que intervienen en el desarrollo de las actividades del sistema, el dominio de aplicaciones de esas actividades y se

modela la estructura funcional, describiendo las funciones, procesos y tareas que se ejecuten. Esta fase culmina con una descripción detallada del modelo del dominio de la aplicación (Ver Figura 3.1)



**Figura 3.1 Esquema de la fase I**

#### **3.1.4.1 Definición del dominio del sistema.**

- Identificación y Análisis del problema:

Surge la necesidad de la construcción de una página Web que contenga un sistema gestor de bibliotecas que permita la automatización de estas y este disponible a usuarios e instituciones académicas públicas o privadas, bajo el enfoque de software libre.

Se da entonces así, inicio a la selección de los módulos que estarían disponibles en ese sistema gestor, estos módulos serán: módulo de adquisiciones, módulo de catalogación y módulo de recuperación de la información.

- Identificación de las necesidades de los clientes:

La necesidad de un usuario o de una institución que administre una biblioteca y requiera de automatizarla, disponga de un recurso más en la web que le permita cumplir su objetivo.

- **Funciones que realiza cada módulo del Sistema**

- El módulo Automatizado de Adquisiciones, realiza las funciones relacionadas con la Adquisición de material bibliográfico y no bibliográfico. Esta basado en el proyecto SIDULA. (Sistema de Información y Documentación de la Universidad de Los Andes).

Atiende las solicitudes de adquisición de material realizadas por los profesores y/o personal calificado y autorizado con ambiente Intranet/Internet. Controlando los procesos de solicitud, facturación, pagos de material bibliográfico, así como también el proceso de recepción e ingreso inicial de la información.

- El módulo Automatizado de Catalogación, este módulo está basado en una filosofía cliente-servidor en la que en el servidor almacena los documentos que forman parte de la biblioteca y el cliente posee una interfaz Windows donde se validan todos los campos del formato Marc, con la que se introduce la información al servidor.

- El módulo Automatizado de Recuperación de Información, permite la búsqueda y recuperación de la información que se encuentra en las diferentes bases de datos dentro de la Universidad de los Andes.

### **3.1.4.2 Identificación y Modelaje de los Procesos del Dominio.**

Los Procesos involucrados en el Sistema, serán detallados debidamente para luego presentar un diagrama general de actividades del sistema.

El sistema permitirá a través de una página web, el libre acceso a usuarios con conocimiento del sistema operativo Linux o administradores encargados de una biblioteca. Deberá registrarse en el sistema como requisito único para la descarga de los módulos de forma independiente, podrá ver el funcionamiento de los módulos a través de un demo existente en la página Web.

En reunión con el tutor y los colaboradores del sistema se tomaron las siguientes sugerencias para la elaboración de la página Web:

La página principal debe contener el nombre del sistema que indique el propósito general (Sistema Gestor de Bibliotecas “Nikolaus Walczuch” bajo software libre), además de páginas secundarias para el registro de los usuarios, manuales de instalación de cada módulo, explicación detallada sobre el uso de cada módulo, descarga de los módulos, sugerencias que pueda aportar el usuario una vez descargue el módulo, desarrolladores y colaboradores, que han intervenido en la elaboración de cada módulo y un enlace web a la Universidad de los Andes (Institución donde se desarrolló el sistema).

De acuerdo a lo anterior se determinaron los siguientes procesos:

- Proceso de selección del software a usar.
- Proceso de diseño.
- Proceso de Inserción de contenidos, en cada página (principal y secundarias).

En la figura 3.2 se detalla un diagrama de actividades donde se describe el comportamiento general del sistema.

- Diagrama de Actividades:

Es un diagrama de flujo (en paralelo) del proceso multipropósito que se usa para modelar el comportamiento del Sistema.

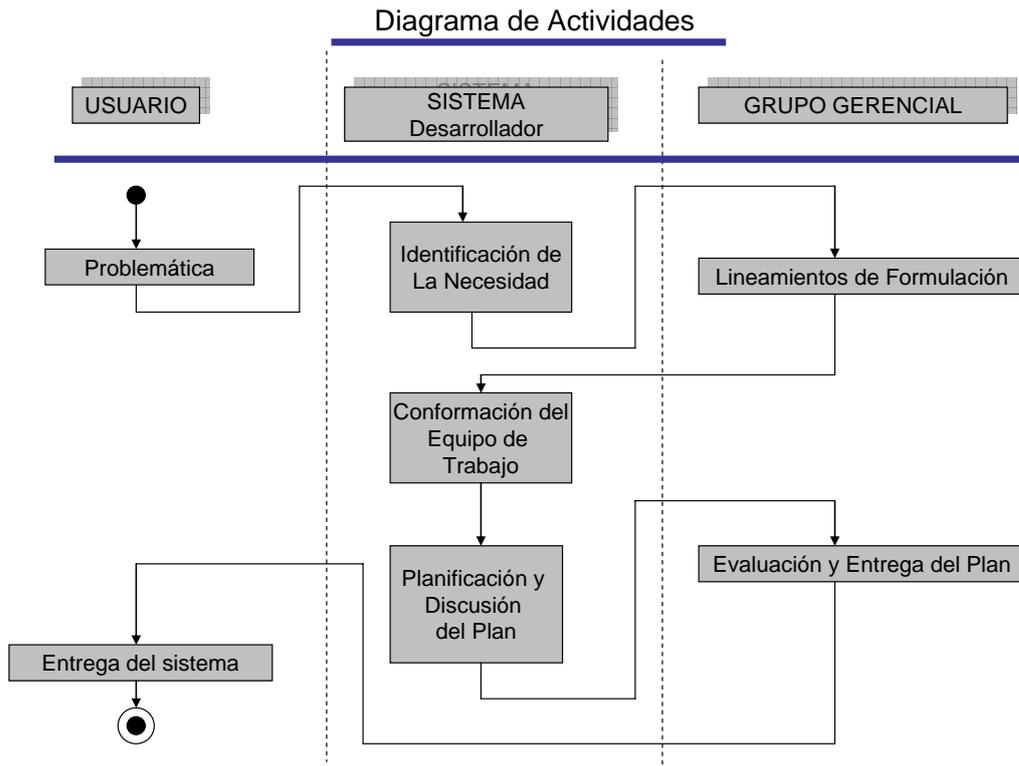


Figura 3.2 Flujo de Tareas Definidas por Grupos

#### 3.1.4.2.1 Proceso de Selección del Software a Usar

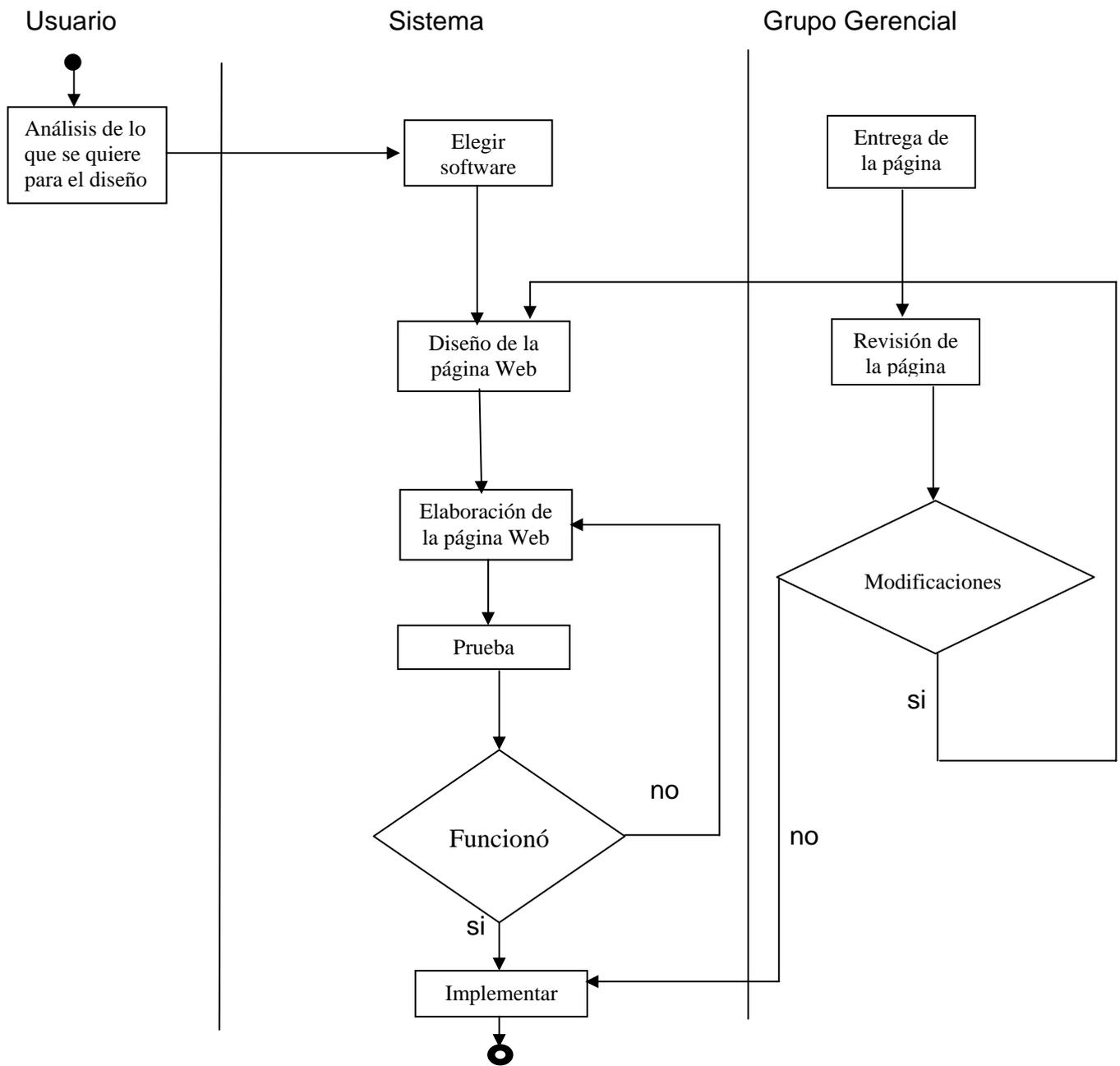
En este proceso se debe definir el software a usar, se eligió un software con código interno Html de macromedia Dreamweaver para el desarrollo del portal, acompañado de photoshop para el tratamiento de imágenes.

Para la configuración de las páginas dinámicas se uso:

- PHP4
- Postgres 7.4

#### 3.1.4.2.2 Proceso de Diseño de la Página Web

El flujo de actividades que se quiere mostrar, será a través de un diagrama de actividades mostrado en la figura 3.3



**Figura 3.3** Diagrama de actividades para el diseño y entrega de la página Web

Este proceso define el diseño de la página Web y se divide como sigue:

- Una página principal titulada con el nombre del sistema y el logo de la Universidad de Los Andes, con enlaces a páginas secundarias tales como:

- Registro de usuarios.
- Textos explicativos de cada uno de los módulos, con enlaces y anclajes para “ver más” en detalle.
- Manuales de instalación, necesarios para la configuración del computador donde se vaya a instalar.
- Descarga de los módulos.
- Enlace a los demos de cada uno de los módulos.
- Sugerencia o aporte al sistema por parte del usuario que descargue cualquiera de los módulos.
- Enlace a una página donde se encuentran el nombre de los principales desarrolladores y colaboradores del sistema.
- Enlace a una página de administración, donde me permita la consulta a la base de datos.
- Enlace a la Universidad de los Andes (ULA).

#### **3.1.4.2.3 Proceso de Inserción de contenidos dentro de cada Página**

Una vez finalizado y teniendo el diseño de la página Web, se paso a la inserción de texto o contenido en la página principal: en la bienvenida al sitio se le indica el propósito general del sistema, en las siguientes páginas señalaremos los propósitos de cada modulo y las descargas de estos.

- **Registrar Usuarios:** el registro de nuevos usuarios está definido por campos como: nombre, apellido, país, institución donde labora o desee instalar el sistema, su correo electrónico, contraseña con que el usuario se registre, otro campo de repita contraseña, debe definir si es administrador del sistema o no, y un último campo para colocar un comentario.
- **Explicación de Funcionamiento de los módulos:** dentro de este enlace aparecen los siguientes módulos: Adquisiciones, Edicla y Sari con su respectivo texto explicativo, además de un enlace adicional que nos lleva a otra página para poder “ver más” en detalle el propósito de cada módulo.

- **Manuales:** en esta parte, nos lleva hacia un manual o pasos de instalación para cada módulo, especificando el software requerido para tal instalación.
- **Descarga los Módulos:** al hacer clic en este enlace, nos lleva a una página donde indica que debo escribir el correo electrónico y la clave o password para aquellos usuarios que ya se han registrado; dentro de la misma página le indica si es un usuario nuevo “regístrate aquí”, que te lleva a la página de registro para nuevos usuarios.
- **Desarrolladores y Colaboradores:** es una página con un pequeño texto sobre los creadores y colaboradores, que han trabajado en el desarrollo de cada uno de los módulos que se presentan en el sistema, partiendo por supuesto de la persona que inicio la automatización de la biblioteca de la Universidad de los Andes, como fue el Profesor “Nikolaus Walczuch”.
- **Sugerencias:** este enlace lleva a una página que presenta dos campos: el usuario (identificado por su correo electrónico), y un campo para su sugerencia o aporte que quiera proponer para el mejoramiento del sistema o de un módulo en particular.
- **Administración:** la parte de administración va a permitir al administrador como tal, la consulta a través de tres tablas en tres páginas diferentes, de la manera siguiente:  
la tabla del usuario definida por nombre, apellido, institución, país y el correo electrónico; la segunda tabla para la descarga, definida por fecha y hora, nombre, apellido, institución y módulo que se descargó; y una última tabla indicando la fecha, nombre y apellido de quien hizo la sugerencia, para ver tal sugerencia basta con pararse sobre el nombre de quien hizo la sugerencia y se despliega un recuadro con la sugerencia o comentario hecha por el usuario.

- **ULA:** este enlace lleva a la pagina Web de La Universidad de Los Andes institución donde ha sido desarrollado el sistema y donde se desarrollaron estos módulos, durante años, que a partir de ahora estarán en la red bajo el enfoque de software libre, para aquellas instituciones que deseen acompañarnos en esta tarea, como lo es, el automatizar los servicios de las bibliotecas y así contribuir a la mejora de nuestras instituciones.

### 3.1.4.3 Identificación y Descripción de los Actores

En esta sección se especifican los diferentes actores o usuarios que interactúan con el sistema.

Se identificó tres tipos de usuarios:

<b>ACTORES</b>	
<b>1.- Administrador</b>	Es el usuario encargado de consultar la base de datos, con la función también de poder eliminar datos.
<b>2.- Usuario Registrado</b>	Es el usuario que ingresa a la página Web y se registra, con la finalidad de descargar algún módulo. Este usuario una vez registrado, tiene la opción de hacer o no sugerencias al sistema.
<b>3.- Usuario no Registrado</b>	Es el usuario que ingresa al sistema solo recorre la página, puede ver los demo, enlace a la ULA, manuales de instalación, etc., pero no puede hacer descarga de módulos ni sugerencias al sistema.

**Tabla 3.1 Actores del Sistema**

#### 3.1.4.4 Identificación y Modelado de las Entidades del Dominio

En este proceso resultaron las siguientes entidades:

**Usuarios:** Son los encargados de interactuar con el sistema, representados por las personas y sus datos.

**Áreas:** Representan las áreas accesibles dentro de la página.

**Servicios:** Representa los diferentes Servicios que ofrece el sitio para quien hace el ingreso al sistema.

**Solicitudes:** Están representadas las descargas que el usuario hace del sistema.

#### 3.1.4.5 Identificación y Descripción de Eventos

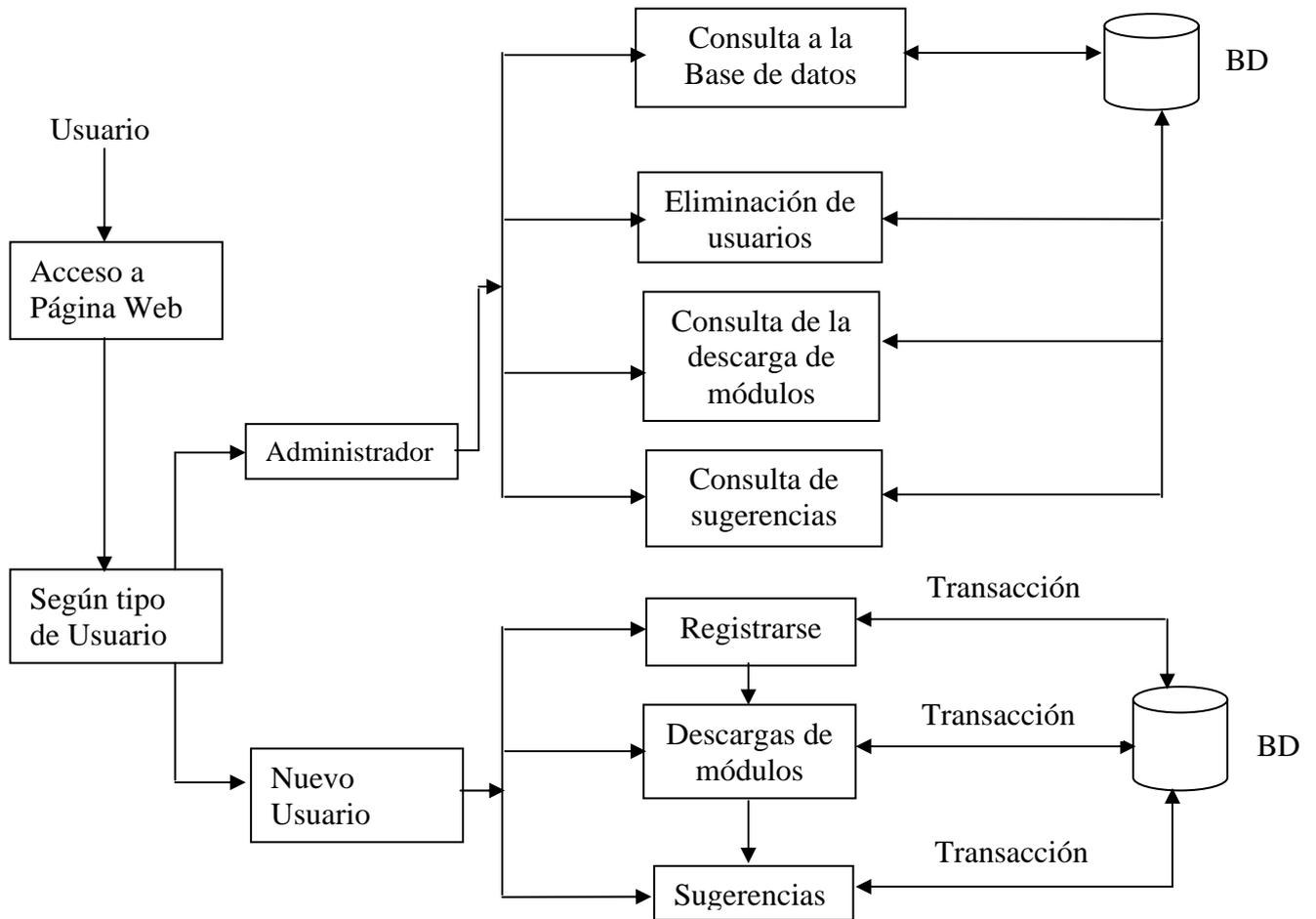
En la Figura 3.3 se puede apreciar las actividades que los clientes pueden ejecutar en el sistema.

El usuario deberá ingresar a la página Web como primer paso, dependiendo del tipo de usuario entrará a la parte que corresponda o a los procesos en sí. En la parte de registrar usuarios lo hacen los actores que ingresan y quieren registrarse (usuarios nuevos). En la parte de Administración el usuario entrará con su palabra clave para consultar la base de datos, y así llevar una estadística del sistema. Estos harán uso de las funciones administrativas para llevar a cabo el proceso de control de solicitud o descarga de los módulos (Control de usuarios que se suscriben, Control de usuarios que han hecho descargas, Control de usuarios que han hecho sugerencias, Control de eliminación de usuarios).

Al usuario nuevo hacer su registro, se le enviará a su cuenta de correos la palabra clave para poder tener acceso a las descargas de los módulos del sistema.

Una vez hecho todo el recorrido del sistema en cuanto a: registrarse, descargar módulos y hacer sugerencias, se estará llenando una base de datos de la persona que se registró, el módulo que descargó y la persona que hizo sugerencias al sistema.

En la figura 3.4 se especifican las funciones del usuario:



**Figura 3.4** Funciones del usuario

### 3.1.5 Definición de Requerimientos

Para esta fase se debe especificar los requerimientos, que son las necesidades que deben ser satisfechas por el sistema, partiendo de dos tipos de requerimientos: los requerimientos **funcionales** y **no funcionales**.

#### 3.1.5.1 Requerimientos Funcionales

Como lo anticipa su nombre, son las funciones que debe realizar el sistema en cuanto a la manipulación de datos por parte de los usuarios (en este caso el

usuario administrador) y poderle brindar a los demás usuarios las funciones para lo cual el sistema fue creado.

Las funciones agrupadas según su tipo:

- Inserción y captura de datos
- Eliminación de usuarios
- Consulta a la Base de Datos
- Descarga de módulos

### Inserción y Captura de Datos

Los requerimientos que se describen en esta fase son los necesarios para capturar los datos del sistema:

- Captura e inserta en la base de datos, los datos del usuario que se registre. Estos datos son los siguientes: nombre, apellido, país, institución, correo, contraseña y tipo de usuario.

### Eliminación de usuarios

Los requerimientos en esta parte describe la eliminación de usuarios, solo por parte del administrador del sistema:

- El usuario administrador puede consultar a la base de datos y listar los usuarios registrados, para elegir que borrar o eliminar de la base de datos.

### Consulta a la Base de Datos

Los requerimientos en esta sección son necesarios para llevar el control estadístico del sistema:

- se captura en la base de datos quien se registra en el sistema
- se captura en la base de datos quien hace comentarios del sistema
- se captura en la base de datos que módulo fue descargado

### Descarga de Módulos

Los requerimientos en esta parte, cumple con una de las funciones del sistema que es permitir descargas:

- Valida si el usuario está registrado permite la descarga

### **Requerimientos funcionales para el control de usuarios**

Permiten al usuario encargado de tal función hacer consultas de registros de usuarios, con la finalidad de llevar un control de todos los usuarios que harán uso del sistema.

Para registrarse en el sistema se deben almacenar los datos siguientes del usuario: nombres y apellidos de usuario, país, institución para la cual desea bajar los módulos, función que ejecutará en el sistema, dirección de correo electrónico, palabra clave o contraseña.

### **Requerimientos funcionales en Servicios**

Se desea un control que realice las operaciones de actualización, consulta e inserción.

El sistema debe permitir almacenar los siguientes datos:

- o En cuanto al Registro de nuevos usuarios (inserción), los campos que debe llenar el usuario para su identificación en el sistema.
- o En cuanto al servicio que se ofrece: el nombre del módulo que desea descargar, pasos para su instalación, demo de cada módulo y explicación en detalle del módulo.
- o En cuanto al control de consultas, para el administrador: quien se suscribe al sistema, que módulo es descargado y quien hace sugerencias al sistema.

Para el control de la estadística del sistema se generan tres módulos, para mostrar consultas en una base de datos: modulo de usuario, módulo de descarga y módulo de sugerencias.

### **Módulo de Usuario**

Este módulo está definido dentro de una tabla con los siguientes campos: nombre, apellido, institución, país y la dirección del correo electrónico.(fig. 3.5)

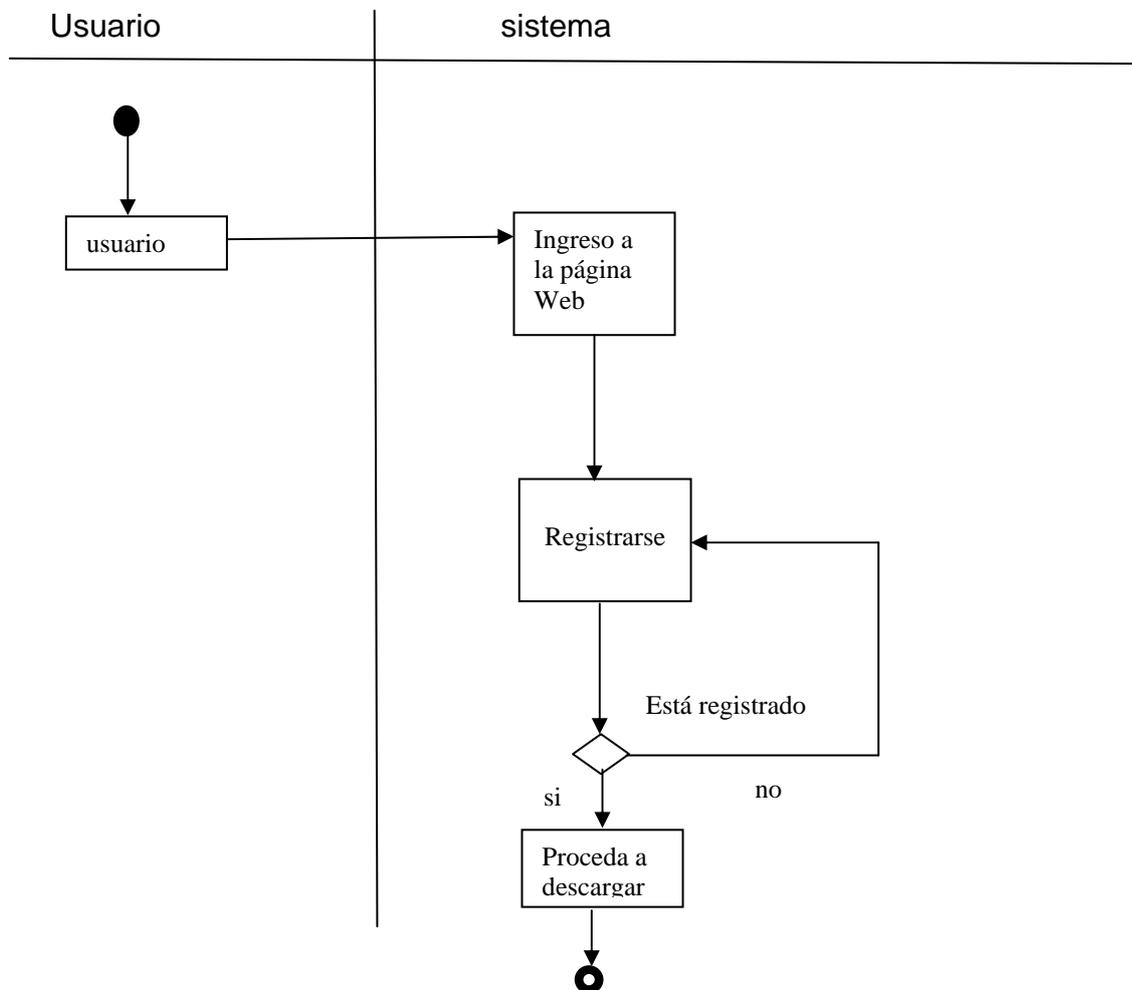
### Módulo de Descarga

Este módulo está igualmente definido con un tabla, como un control para llevar la estadística y comprende los siguientes campos: fecha, nombre, apellido, institución y módulo que descarga.(fig. 3.6)

### Módulo de Sugerencias

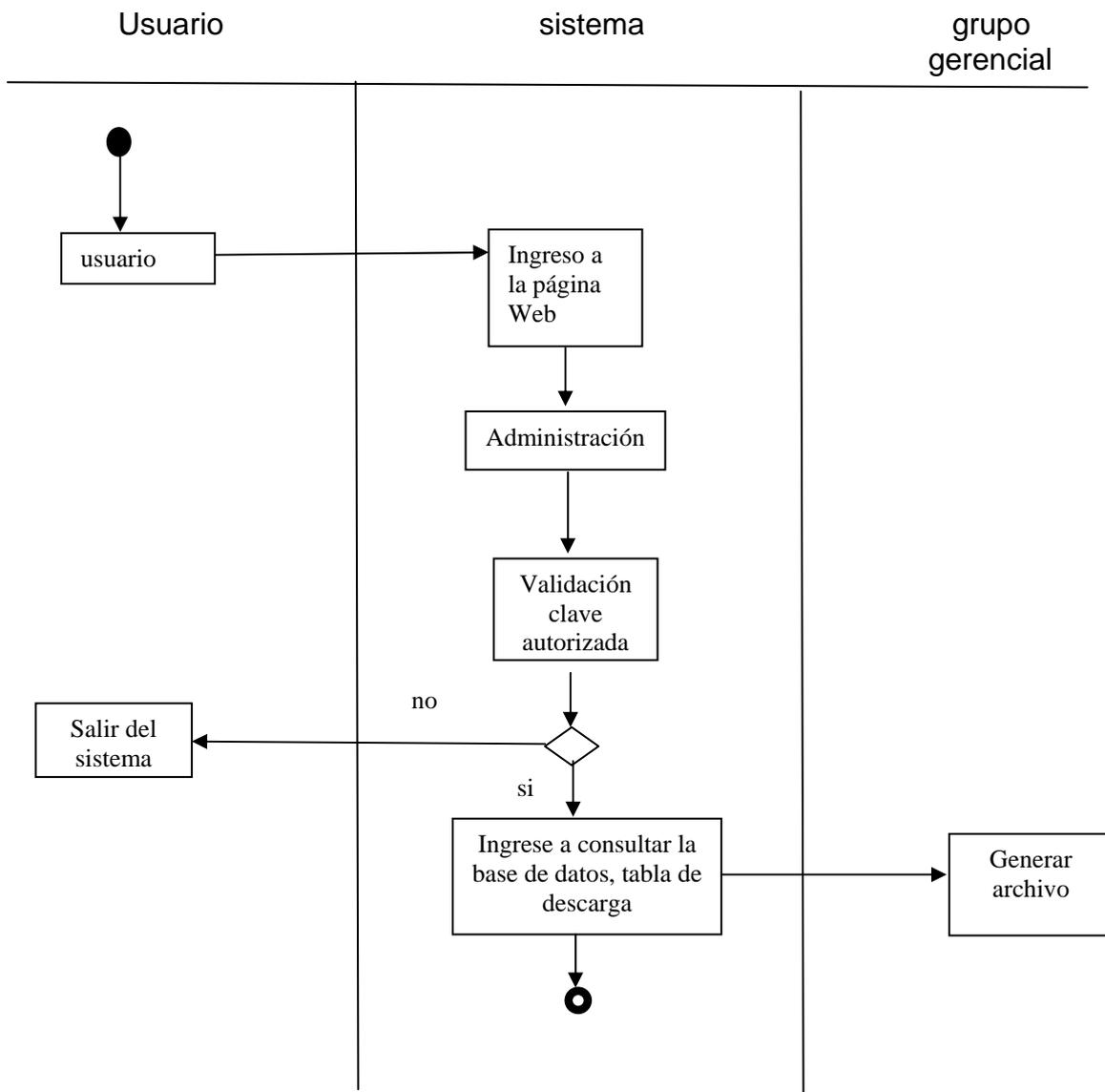
Este módulo comprende los siguientes campos: fecha, nombre y apellido.(fig.3.7)

Diagrama de actividades para el módulo de usuario nuevo que desee registrarse:



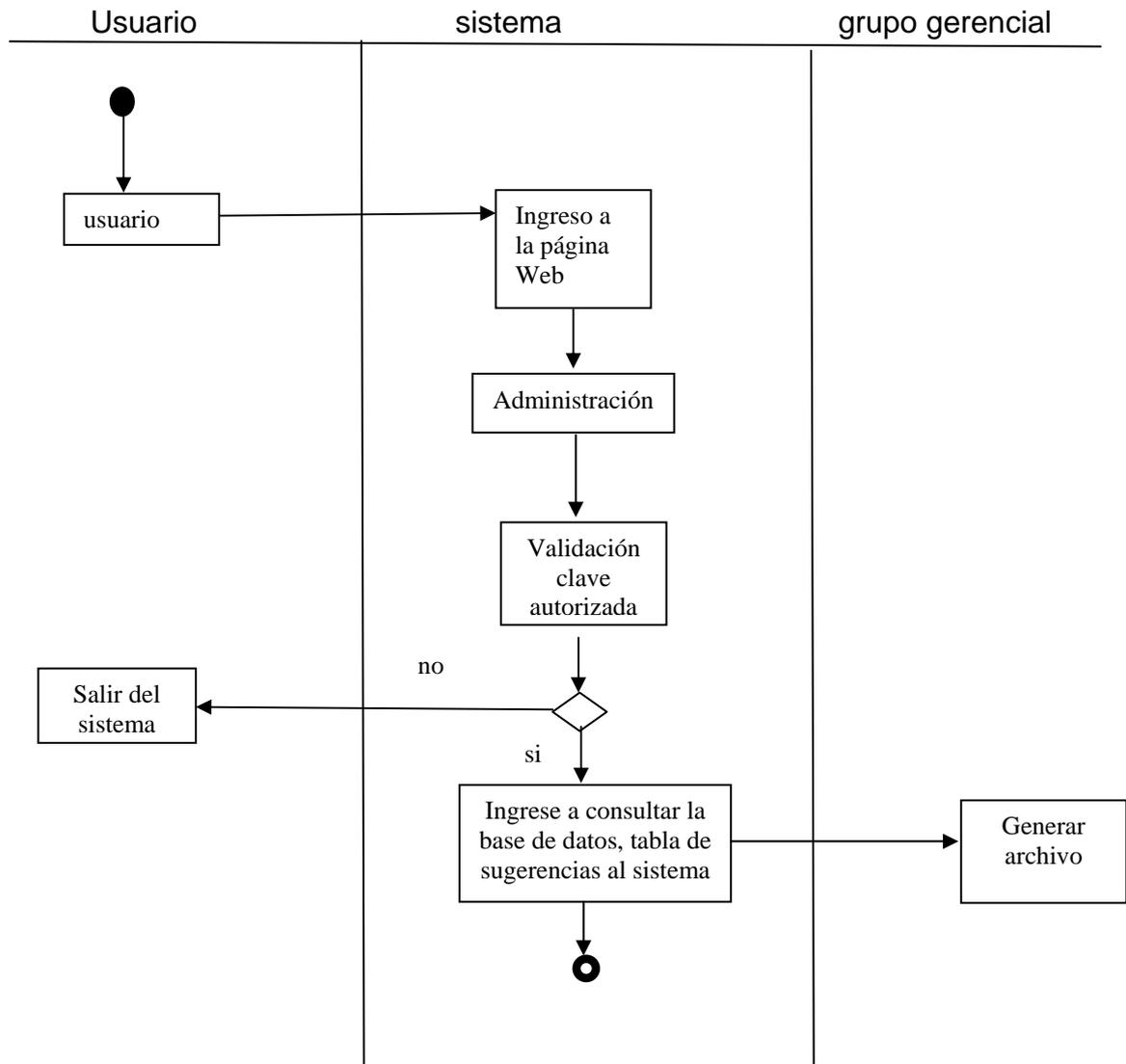
**Figura 3.5** Diagrama de actividades del módulo usuario

Diagrama de actividades para el módulo de descarga, para el control del sistema por parte del usuario administrador:



**Figura 3.6** Diagrama de actividades del módulo descarga

Diagrama de actividades para el módulo de sugerencias, para el control del sistema por parte del usuario administrador:



**Figura 3.7** Diagrama de actividades del módulo sugerencias

### **3.1.5.2 Requerimientos no funcionales**

Los requerimientos no funcionales corresponden a los aspectos del sistema que no cumplen una función específica, pero contribuyen a la interacción entre el sistema y los actores.

### **3.1.5.3 Requerimientos de Interfaz de Usuario**

La interfaz es el medio de comunicación entre el usuario y el sistema. Se busca diseñar y crear una interfaz de usuario de ventana, con marcos interactivos que permitan la navegación por la aplicación sin mayor complicación, lo más sencilla posible, fácil de usar, de entender y aprender.

El formato general debe contener colores y texturas que no cansen al usuario.

Antes de enviarse la información al sistema en la interfaz se deben validar que la información este completa, que se corresponda con el tipo de dato que va a almacenar la Base de Datos, que el usuario tenga permiso para esa operación y no debe interactuar con el mismo si esto no se cumple.

### **3.1.5.4 Requerimientos de Desarrollo y Operación**

El Diseño del sistema de Gestión de bibliotecas, fue realizado usando las siguientes herramientas mostradas en la tabla 3.2:

Herramienta de diseño	Descripción
<b>Photoshop</b>	<p><b>Photoshop</b> presenta un entorno completo para diseñadores y grafistas profesionales en el que podrá crear sofisticadas imágenes para impresión, Internet, dispositivos inalámbricos y otros medios. Además, Photoshop proporciona soluciones para la edición profesional de imágenes. Con la aplicación de herramientas Web integradas.</p>
<b>DreamweaverMX</b>	<p>Gran parte de la interfaz será realizada con esta herramienta ya que esta es una de las herramientas que ayuda a crear páginas Web de gran calidad, fácil de usar entre otras bondades.</p> <p>Esta Herramienta genera el código HTML, puesto que es un editor de Código HTML. Junto a esta herramienta se utiliza lenguaje java script para algunas validaciones.</p>
<b>PHP4</b>	<p>Con esta herramienta se hace la mayor parte de la programación de este sistema. Es la que abre la puerta de comunicación entre la base de datos y la interfaz.</p>
<b>PostgreSQL</b>	<p>Este es el sistema manejador de Base de Datos que se empleará para la implantación de La base de datos del Sistema.</p>

**Tabla 3.2** Especificaciones del diseño del sistema

### **3.1.5.5 Perfil de seguridad y acceso que tienen los Usuarios** (Dependiendo del tipo de usuario)

En la actualidad cualquier usuario nuevo inserta, y solo el usuario administrador consulta a la base de datos, se quiere:

- Los perfiles de cada quien para la Web, dependiendo del rol del usuario en el sistema.
- Si alguien quiere eliminar o consultar (Es con clave).
- Registrarlo en una base de datos.
- Interfaz dinámica (Aparece para cada quien) en función de los parámetros.
- La seguridad de los datos. (solo permitido para la persona autorizada).

### **3.1.5.6 Identificación de atributos de calidad**

Especifican un conjunto de condiciones que determinan la calidad del sistema.

- Dispone de mecanismo de control de usuarios, para la selección de distintos tipos de acceso.
- El tiempo de respuesta es mínimo para la inserción de registros en la base de datos.
- Incorporación de un manual (sencillo) para pasos de instalación y descarga de los módulos ofertantes.

### **3.1.6 Especificación y análisis de los requerimientos**

Se debe expresar los requerimientos del usuario de manera formal o técnica para que sea entendida por los diseñadores del sistema.

## Refinamiento de los diagramas de caso de Uso

Se presenta a continuación la descripción de actores del sistema y sus casos de uso en la tabla 3.3

Actores	Caso de Uso
Usuario nuevo	Registrarse
Usuario registrado	Descargar / hacer sugerencias
Usuario administrador	Consultas a la BD

**Tabla 3.3** Casos de Uso del Sistema, junto con su actor

Se presentan a continuación caso de uso del sistema en general:

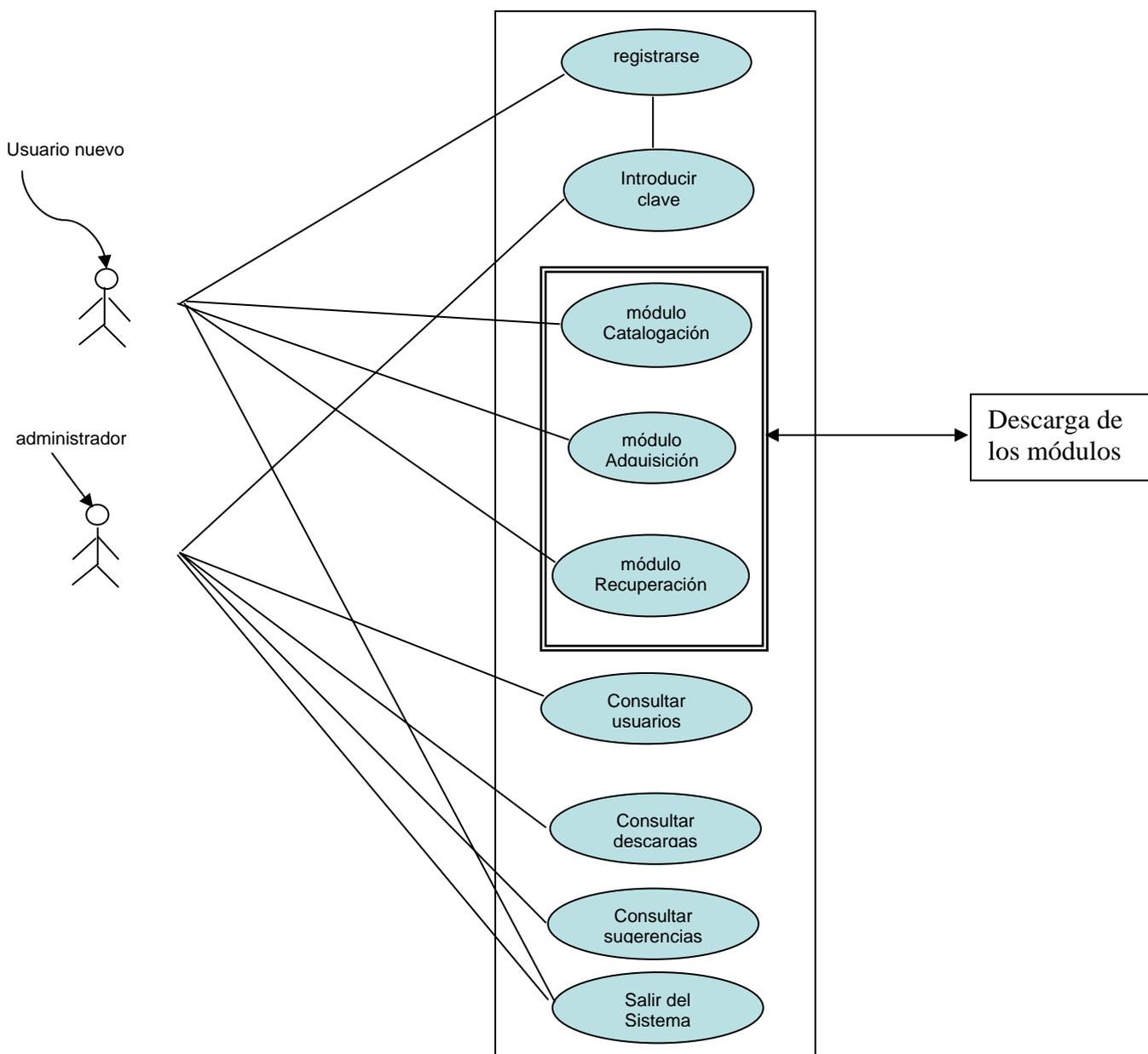
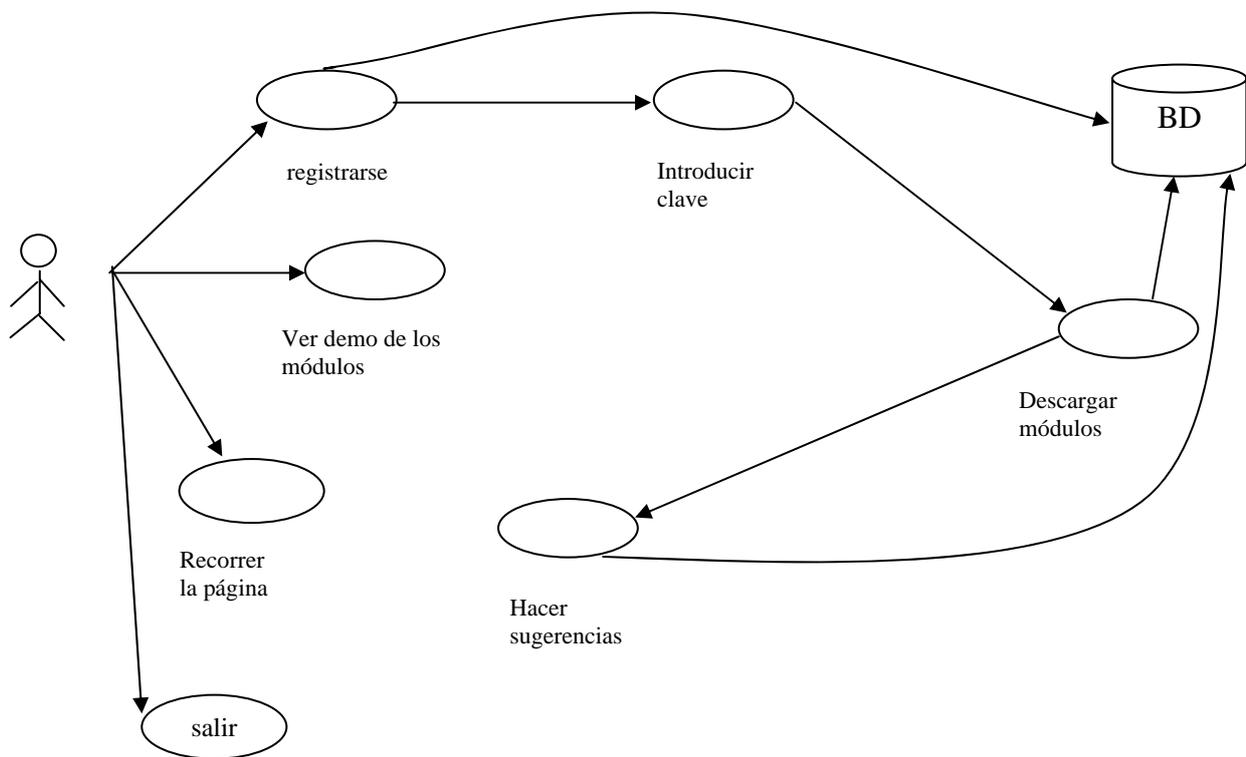


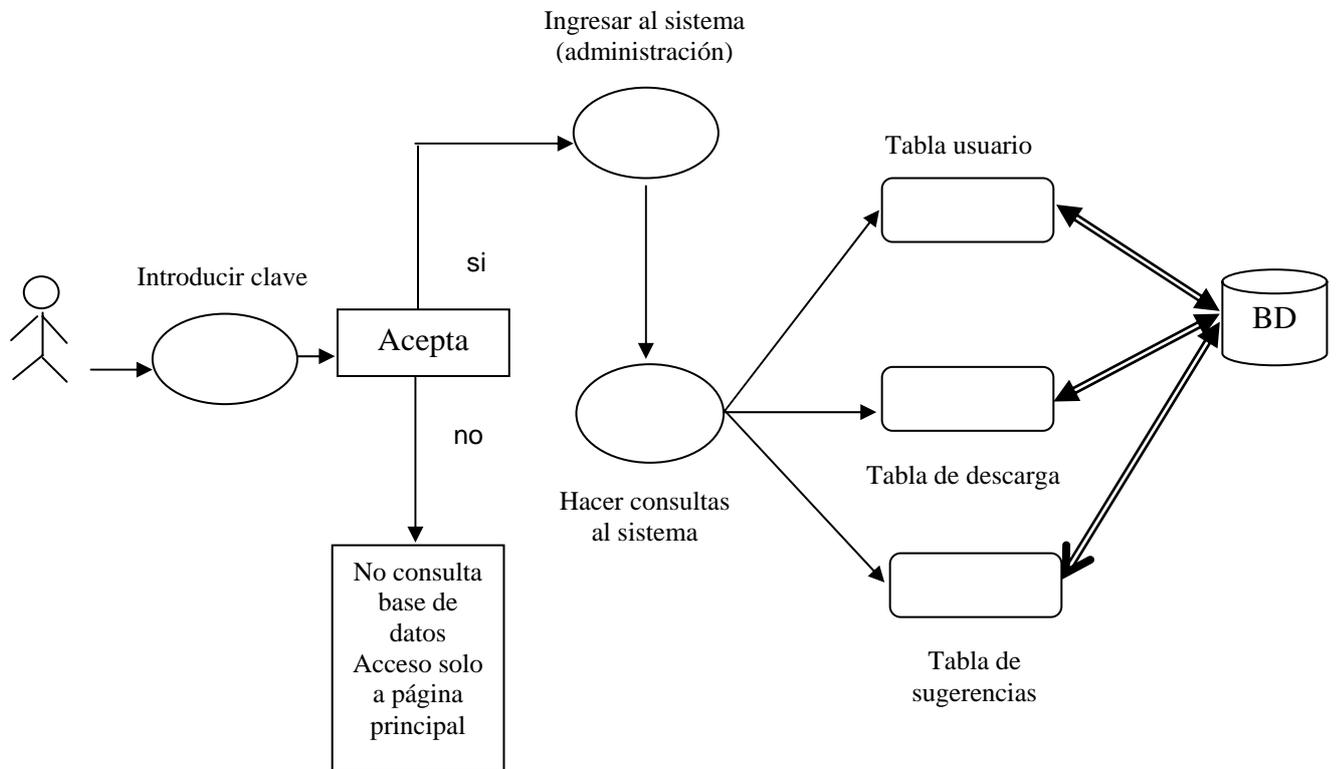
Figura 3.8 Diagrama de caso de uso general del sistema

Desglose del caso de uso del nuevo usuario y usuario registrado, El registro, las descargas y las sugerencias alimentan la base de datos:



**Figura 3.9** Diagrama de caso de uso del usuario

Desglose del caso de uso del usuario-administrador o persona autorizada, hace consultas a la base de datos y genera un archivo para el control estadístico:



**Figura 3.10** Diagrama de caso de uso del usuario-administrador

El desglose de cada caso de uso anterior se describe de la siguiente manera:

**Caso de Uso:** descarga de módulos.

**Actor:** nuevo usuario.

**Descripción:** El manejador de Bibliotecas luego de registrarse en el sistema con sus datos: nombre, apellido, país, institución y su dirección de correo, el sistema le envía a su cuenta de correo la palabra clave y así ingresa al sistema de descargas selecciona el módulo que desea descargar y el sistema en respuesta le despliega el nombre del módulo, le muestra texto explicativo sobre el propósito del módulo, le indica los pasos de instalación y el software necesario para su instalación.

**Caso de Uso:** Consultas a la Base de Datos.

**Actor:** Usuario administrador

**Descripción:** En este caso el administrador del sistema o la persona autorizada, selecciona el área que desea consultar y el sistema en respuesta le despliega una tabla con los campos antes descritos: consultar usuarios, descarga o sugerencias.

### **3.1.6.1 Derivación del modelo de Objetos**

Los diagramas de objetos modelan las instancias de los elementos contenidos en los diagramas de clases

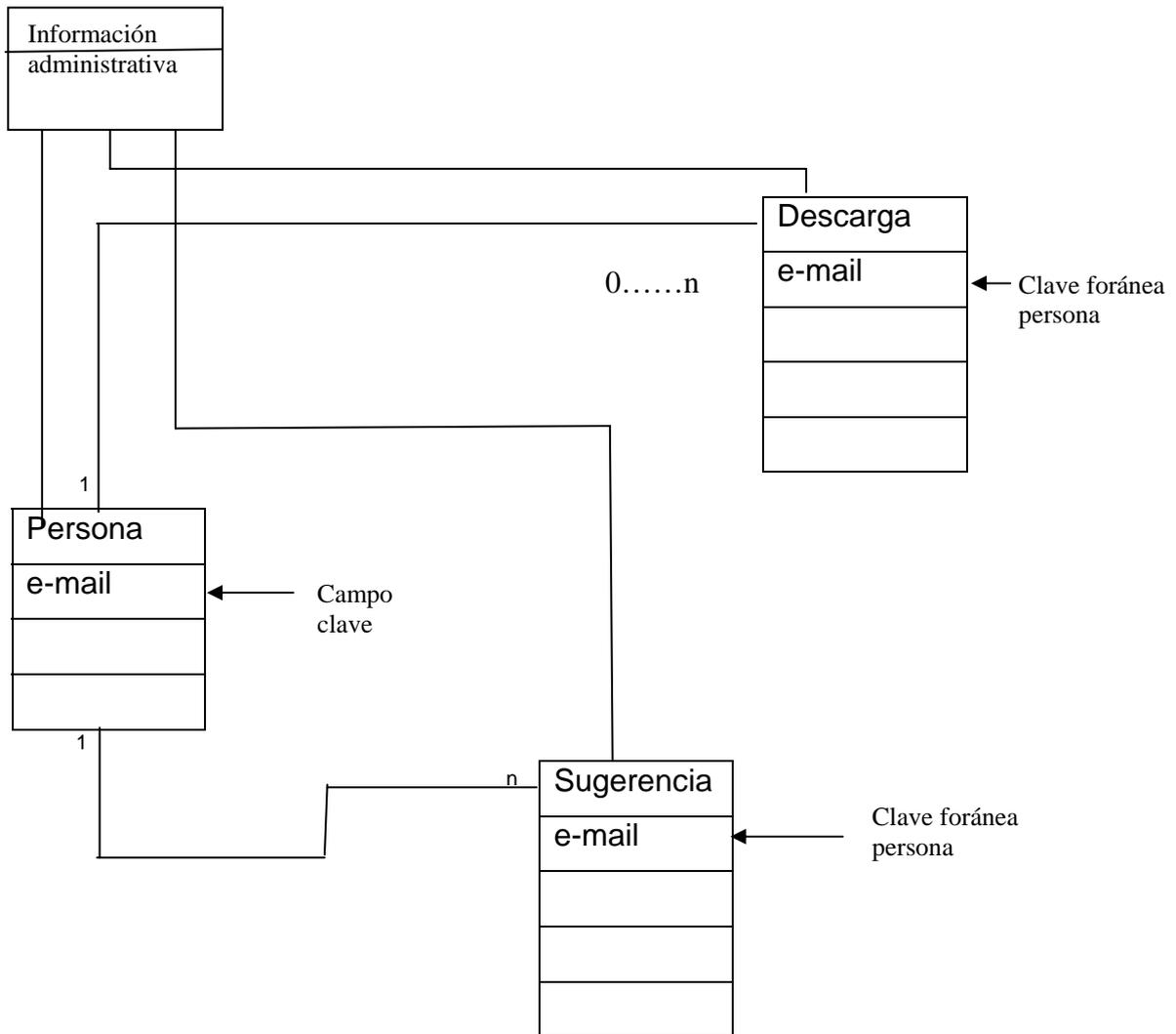
#### **Diagramas de Clases**

Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones.

Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema.

El primer modelo de objeto o de inicio fue originado en el diseño inicial, partiendo de la necesidad de la creación de la base de datos para la clase administrativa.

Partiendo de estos conceptos y en con apoyo del diagrama anterior, este generó el siguiente diagrama de clases; las clases obtenidas son: personas, descarga y sugerencias.

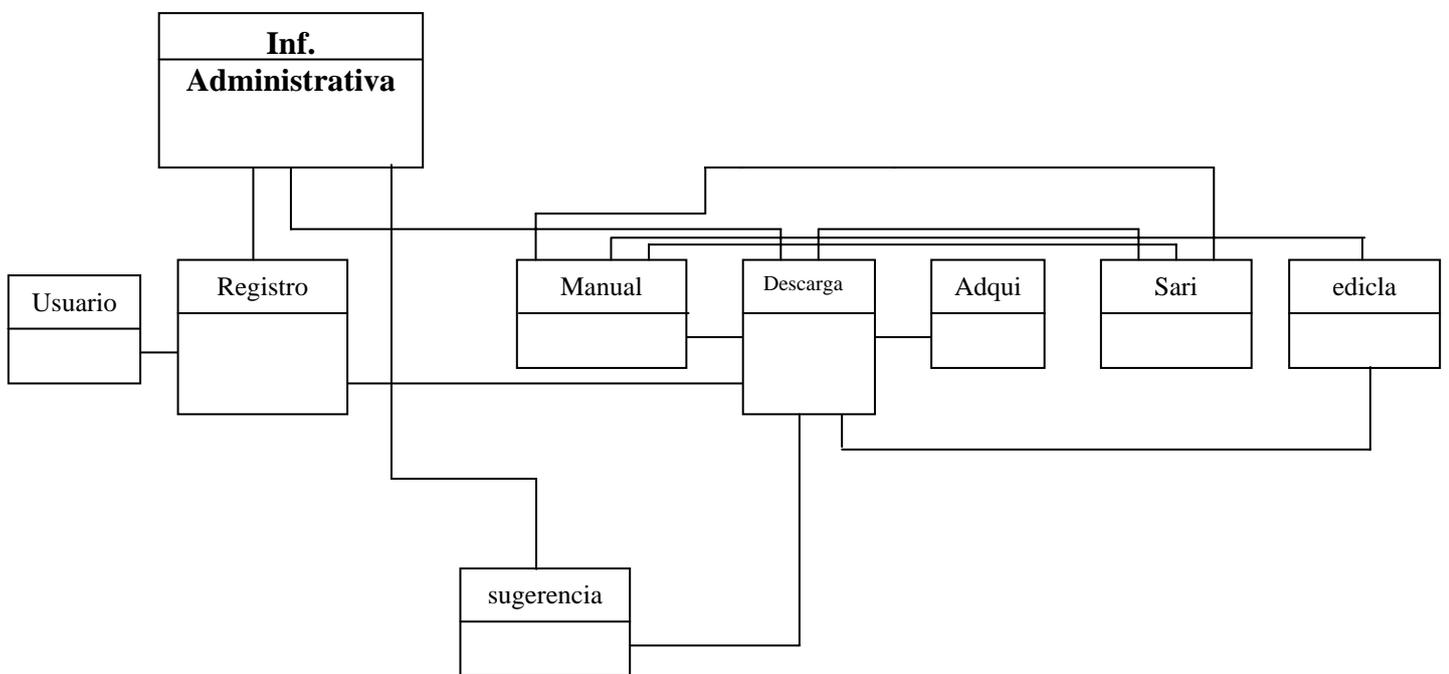


**Figura 3.11** Diagrama de clase de la base de datos

### Diagrama de clase relacional

La clase de información administrativa es la relación de el registro de usuario, la descarga que el hace y la sugerencia que pueda hacer o no, pasan a la base de datos. Ver figura 3.12

Estos diagramas no contienen los atributos ni métodos de las clases, pues solo enfatizan las relaciones entre las clases.

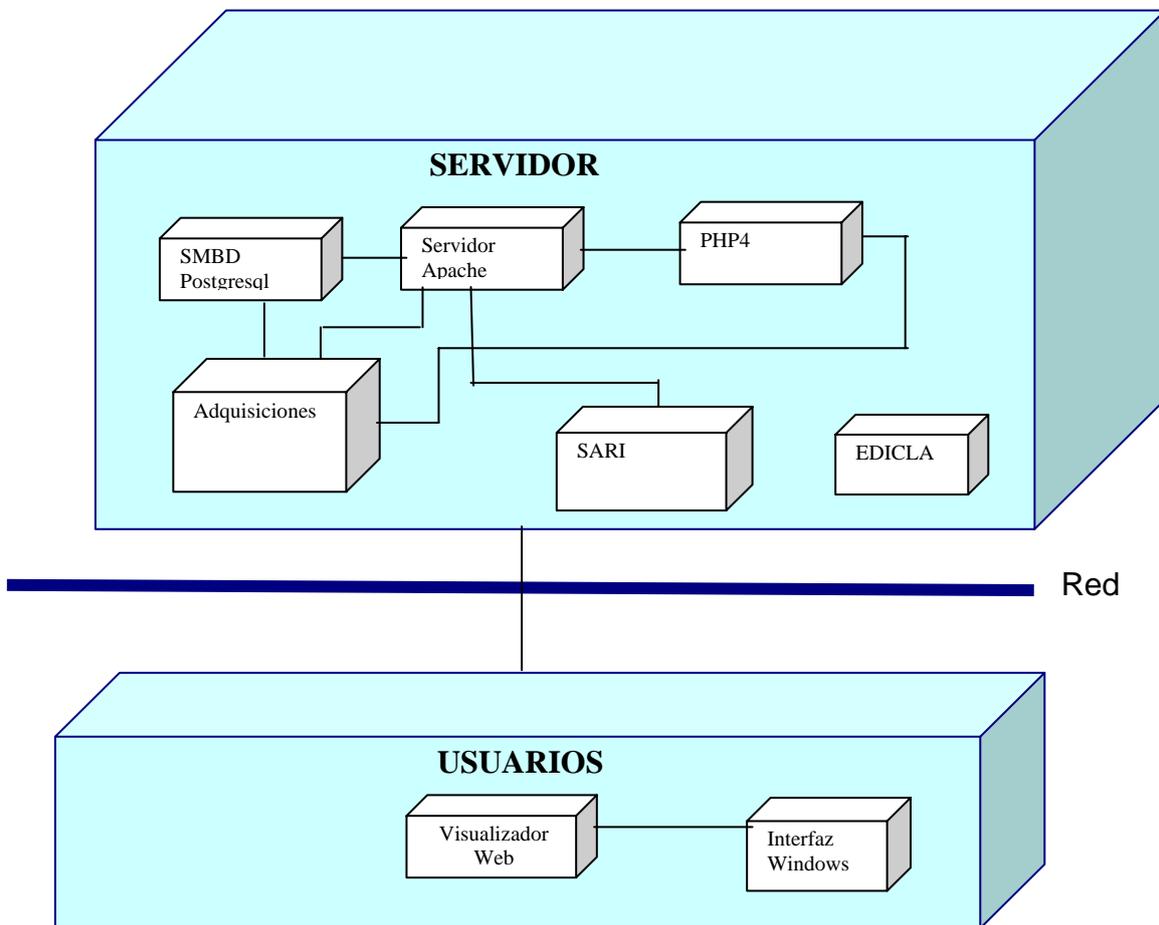


**Figura 3.12** Diagrama de relación de clases

### Diagrama de Despliegue

En este diagrama se precisa el reparto de los procesos. En la figura 3.13 se presenta el diagrama de despliegue del sistema donde los nodos representan los computadores (Servidor y usuarios). En el nodo servidor están los módulos, que son el sistema en si, el lenguaje Php, el servidor apache y el sistema manejador de base de datos postgre. El sistema necesita del lenguaje Php y el Servidor Apache para interactuar con la Base de Datos.

En el nodo Usuarios están los componentes Visualizador, Red e interfaz Windows para el módulo Edicla.



**Figura 3.13** Diagrama de Despliegue.

## CAPITULO 4

### IMPLEMENTACIÓN, PRUEBAS Y RESULTADOS

En este capítulo se especificará todo el desarrollo que se hizo para la implementación del sistema gestor de bibliotecas “Nikolaus Walczuch” bajo software libre,

Este sistema se implementó utilizando plataforma Linux(Ubuntu), soportado por un sistema manejador base de datos (SMBD) Postgresql, basado también en páginas Web elaboradas en DreamweaverMX, como interfaz al usuario.

El desarrollo de este trabajo de grado fue realizado en el laboratorio “Nikolaus Walczuch”, del departamento de computación de la escuela de Ingeniería de sistemas de La Universidad de Los Andes. Este proceso de desarrollo comprendió dos etapas.

En la primera etapa se realizaron los siguientes pasos:

- **Selección del Hardware**, escogencia de una computadora con buenas características para la instalación del sistema, estas características se especificarán mas adelante.
  
- **Instalación del sistema operativo escogido**; software libre Linux/Ubúntu(5.10), este proceso ocurre como cualquier instalación de software, el propio software guía la instalación y da la opción de instalar y configurar o no algunos paquetes propios del software, esta instalación es relativamente fácil para cualquier usuario con conocimientos medios o básicos de computación.

Para el funcionamiento correcto del sistema se deben instalar los siguientes paquetes:

- servidor apache o apache 2
- Php4, es importante la instalación de este software, ya que todos los módulos fueron programados en Php4.
- Postgresql (7.4), este software se usó como sistema manejador de base de datos(SMBD).

- **Agrupar módulos**, Se comenzó con la revisión del Sistema Sidula 2000 existente en la Universidad de Los Andes, para comprobar el funcionamiento de cada uno de los módulos, módulos estos que se van a agrupar y montar en el nuevo sistema. Estos módulos son:
  - Módulo de Adquisiciones
  - Módulo de recuperación de la información (Sari)
  - Módulo de Catalogación, solo para revistas(win-edicla) con interfaz para el usuario con Windows.
- **Empaquetar y Comprimir los módulos**, luego de el montaje de cada módulo en el nuevo sistema y probar su funcionamiento para luego proceder a la corrección de errores y poner a funcionar cada uno de los módulos, se procedió a efectuar el empaquetamiento y a comprimir cada módulo, se debe decir que los módulos empaquetados y comprimidos son: Adquisiciones como “adqui.tgz” y sari como “sari.tgz”, ya que win-edicla es una interfaz bajo Windows.
- **Definir directorios**, para el funcionamiento de algunos de los módulos es imprescindible definir los directorios y rutas donde el usuario final debe hacer el des-empaquetamiento y descompresión de los archivos .tgz anteriormente mencionados, para su correcto funcionamiento.
- **Elaboración de manuales**, en cuanto a este punto se debe definir la configuración del software que el usuario final necesite para su correcto funcionamiento, además de indicar de forma precisa los pasos de instalación de cada uno de los módulos.

La segunda etapa esta definida por dos procesos:

- **La elaboración de una página Web**, donde poder descargar el sistema y a través de esta el usuario final pueda tener acceso a la descarga de estos módulos.
- **Adecuar la primera etapa en un caso específico**, por ejemplo, la biblioteca del Centro Interamericano de Desarrollo e Investigación Ambiental y Territorial (CIDIAT), bajando un módulo; como prueba de que el Sistema funciona.

#### 4.1 Requerimientos de Hardware y Software

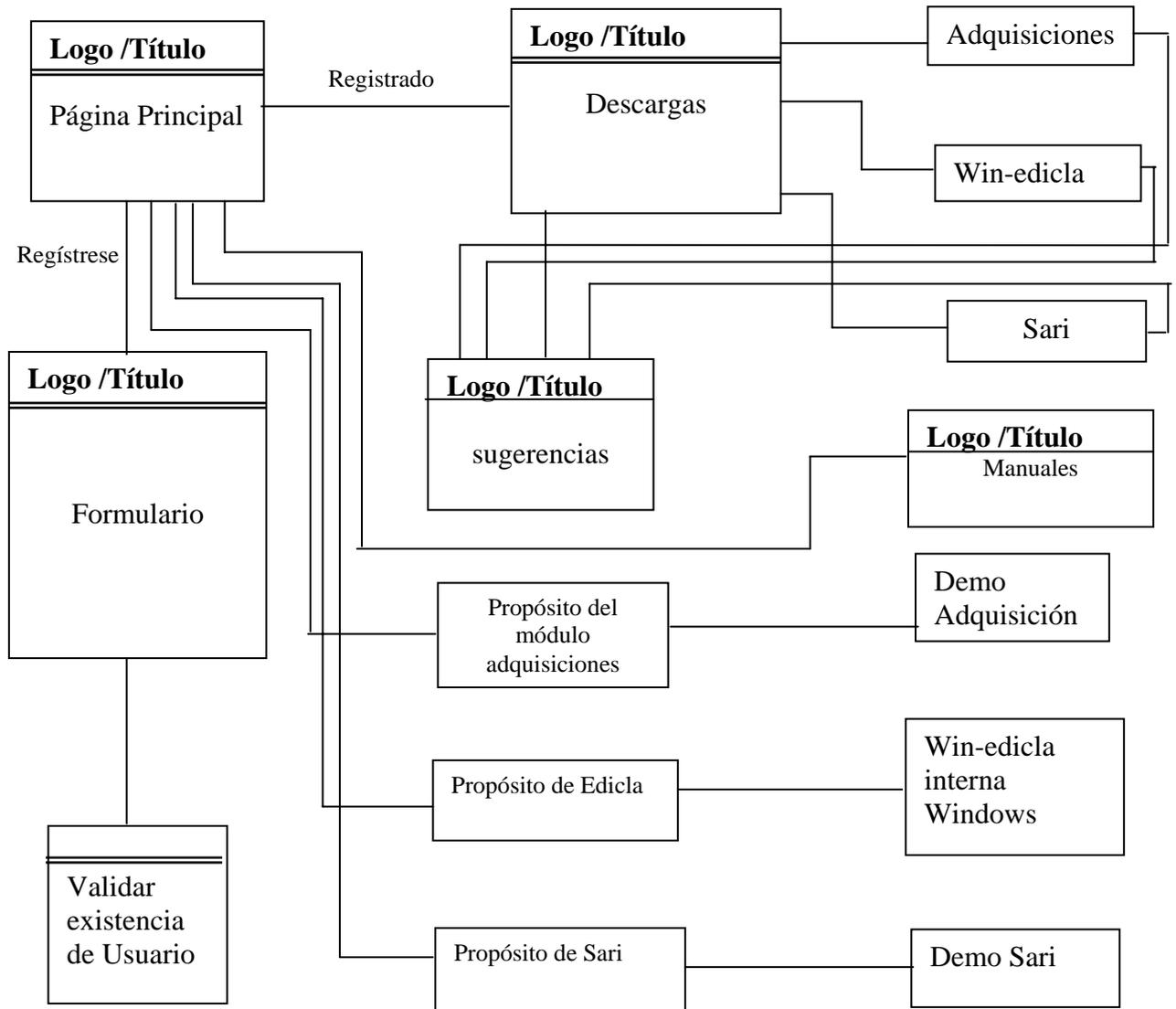
En la siguiente tabla se especifica el hardware y software usados en la computadora donde se instaló el sistema y la página Web, así como se menciona también los requerimientos mínimos que debe tener el equipo del usuario final que desee instalar el sistema.

<b>SERVIDOR</b>	Memoria Ram 256 Mb.
	Disco duro de 40 Gb
	Sist. Operativo Linux-Ubuntu
	Php4
	Postgresql (SMBD)
	Servidor apache
<b>USUARIO</b>	<b>Requisitos mínimos</b>
	Procesador Pentium 2
	Disco Duro de 20 Gb (crecimiento a futuro)
	Memoria Ram 64 Mb
	Sist. Operativo Linux (no especifica)
	PHP4
	Postgres
	Servidor Apache

**Tabla 4.1** Requerimientos de hardware y software

## 4.2 Diseño de la Página Web

En esta sección se muestran ventanas, marcos y formularios diseñados en esta etapa.



**Figura 4.1** Modelo esquemático del diseño de la página

### 4.3 Creación de la Base de Datos

La creación de la base de datos llamada sidula parte de las tablas descritas en el capítulo anterior como diagramas de clase; la tabla persona, la tabla descarga y la tabla sugerencia con sus respectivos campos.

- Creación de la Primera Tabla (Persona):

Create table persona (nombre text, apellido text, pais text, institución text, correo text, administrador boolean, primary key(correo));

- Creación de la segunda Tabla (Descarga):

Create table descarga (correo text, referentes persona(correo), modulo integer, fecha date);

- Creación de la tercera Tabla (Sugerencia)

Create table sugerencia( correo text, referentes persona(correo), modulo integer, fecha date);

El uso de php va a permitir la conexión con la base de datos sidula.

## Página principal del Sistema Gestor

En esta sección se describe de manera detallada los distintos enlaces de la página Web del sistema en su totalidad.

El encabezado de la página muestra, en su parte superior pestañas activas a enlaces, tales como: Principal (inactivo) por estar ubicado en ella, descarga de módulos(activo), manuales de instalación(activo), Sugerencias(activo), ULA(activo), desarrolladores y colaboradores. El logo de la Universidad de los Andes y el Título del sistema en este proyecto de grado.

Mas abajo en la parte izquierda: registrar usuarios, enlace hacia los manuales de instalación y el nombre de cada módulo con un enlace hacia un texto explicativo de cada uno de los módulos, indicando el propósito para cada modulo.

En la parte central y superior de la página se muestra el propósito general del sistema, los cómo de instalación y un enlace hacia las descargas de los módulos.(Ver figura 4.2)

Principal Descarga Módulos Manuales de Instalación Sugerencias ULA Desarrolladores y Colaboradores

 **Sistema Gestor de Biblioteca**  
**Nikolaus Walczuch**  
Bajo software libre

**Registrar Usuarios**

**Manuales**

**Explicacion de funcionamiento de los Módulos**

**Adquisiciones Edicla SARI**

**Descarga Los Modulos ya!!!**  **Clic Aquí**

**Bienvenidos**  
Proposito general:

**Cómo de Instalacion**  
instalación bajo plataforma Linux

Este Portal ofrece un Sistema Integral de gestión de bibliotecas a sus administradores, permitiendoles la descarga de tres(3) módulos como son: Sistema automatizado de recuperación de información (SARI), Módulo de Adquisiciones bibliohemerográficas y módulo win-edicla(referente a Catalogación). Enmarcado bajo Software Libre (GNU) y diseñado por el Dpto. de Computación de la Facultad de Ingeniería, Escuela de Sistemas Universidad de Los Andes. Bajo la tutoría del Profesor (M.Sc.) Domingo Hernández.

Además permitiendole al usuario la actualización y automatización de sus bibliotecas le permite interactuar con este sitio web ,a través de sus sugerencias y aportes para este sistema.

Figura 4.2 Interfaz diseñada para la Pág. principal

**Enlace a Registrar Usuarios:** el mismo marco superior y una tabla con los campos señalados, campos estos que deben ser llenados por el usuario, excepto el comentario que es una opción.

Una vez que el usuario pulse el botón aceptar, el sistema validará los campos del formulario y le enviará su clave a la cuenta de correos. (Ver figura 4.3)



## Registro de Nuevos Usuarios

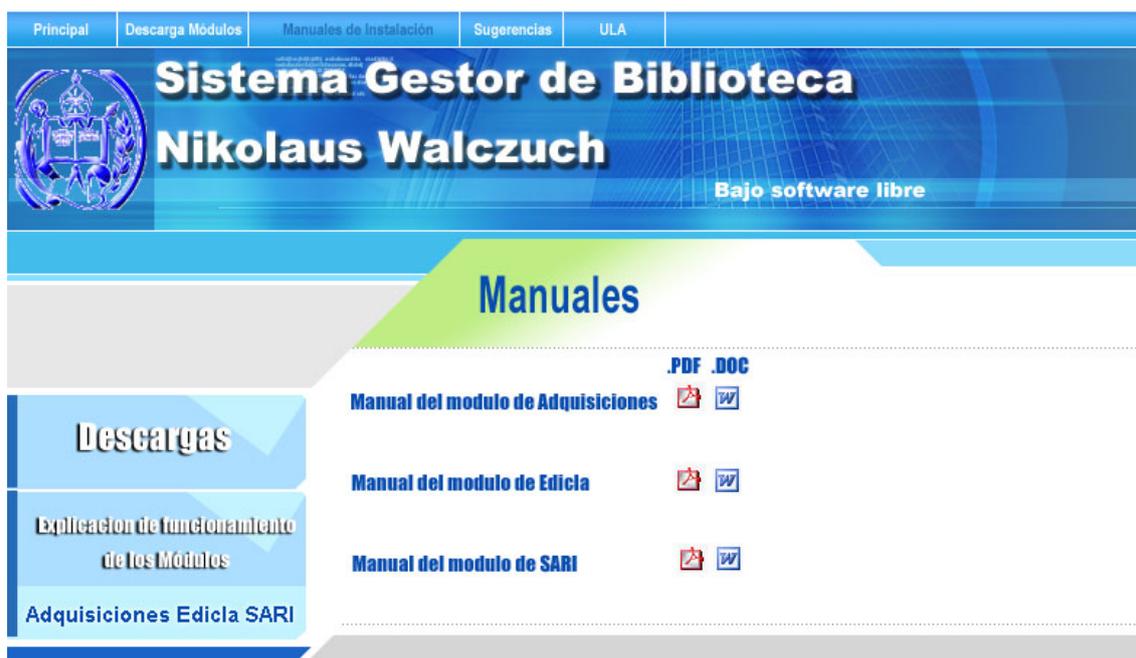
---

Nombre	<input type="text"/>
Apellido	<input type="text"/>
País	<input type="text"/>
Institucion	<input type="text"/>
e-mail	<input type="text"/>
Contraseña	<input type="password"/>
Repita-contraseña	<input type="password"/>
Administrador de biblioteca	<input type="checkbox"/> si <input type="checkbox"/> no
Comentario	<input type="text"/>

---

**Figura 4.3** Interfaz diseñada para el registro de nuevos usuarios

**Ventana para Manuales:** Este enlace para ver los manuales o pasos de instalación, permiten presentar la información necesaria para la instalación de cada modulo como el software que se necesita para su correcto funcionamiento, la información es presentada en formato .pdf y formato .doc.(Ver figura 4.4)



**Figura 4.4** Interfaz diseñada para la presentación de manuales

**Ventana para Descargas:** Una vez que el usuario pulse el botón para efectuar la descarga le mostrará una ventana donde le solicita su e-mail o dirección de correo electrónico y su password o palabra clave, solo para aquellos usuarios que estén registrados y así permitirle la descarga del módulo que desee instalar. Además en la misma pantalla le ofrece la opción de registrarse si aun no lo ha hecho. Estos archivos estan empaquetados y comprimidos, por lo tanto su formato es .tgz (Ver figura 4.5)



**Figura 4.5** Pantalla diseñada para la presentación de Descargas

**Ventana para Sugerencias:** Una vez abierta esta ventana permite al usuario ingresar su cuenta de correo en el campo “usuario” y su sugerencia si así lo desea, al pulsar aceptar la sugerencia se suma a la base de datos, ya descrita.(Ver figura 4.6)

## Sugerencias

Usuario

Sugerencia

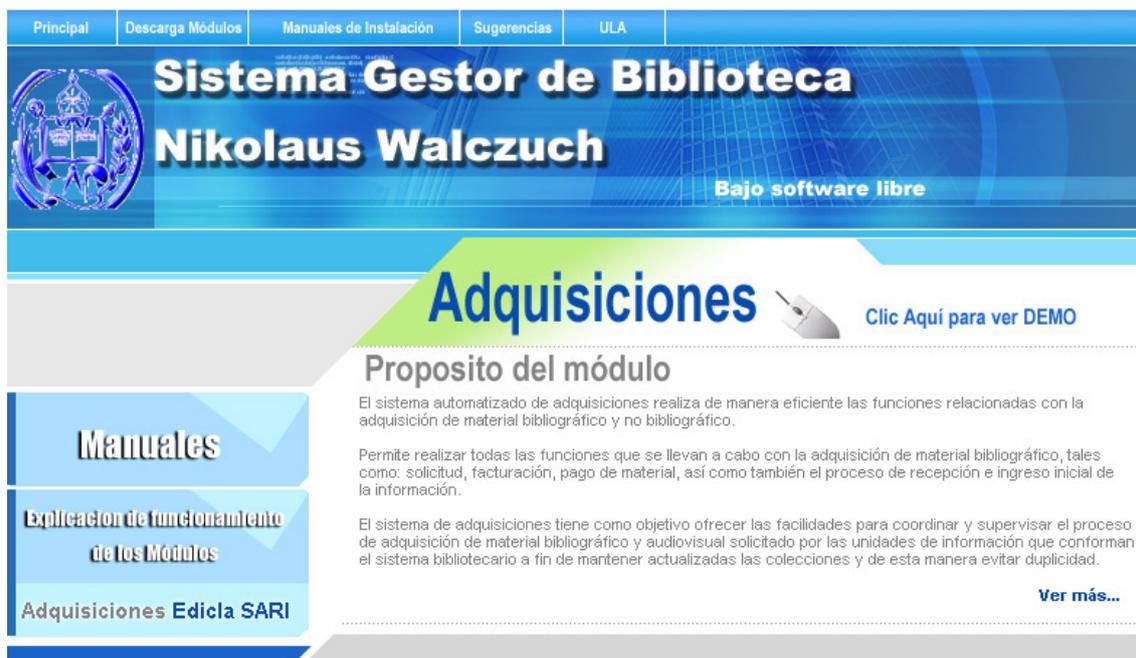
---

[ir a la página principal](#)

Autor: Carlos Moreno Sarmiento

**Figura 4.6** Pantalla diseñada para la presentación de Sugerencias

**Ventana del Módulo Adquisiciones:** Esta ventana permite presentar el propósito del módulo de adquisiciones, un “ver más” para conocer mas a fondo sobre el módulo y a su vez ver un demo sobre su funcionamiento.(ver figura 4.7)



**Figura 4.7** Pantalla diseñada para la presentación del módulo Adquisiciones

**Ventana del Módulo Edicla:** Esta ventana permite presentar el propósito del módulo de win-edicla, un “ver más” para conocer mas a fondo sobre el módulo y para ver un demo el usuario debe hacer una interfaz desde Windows.(ver figura 4.8)



**Figura 4.8** Pantalla diseñada para la presentación del módulo Edicla

**Ventana del Módulo Sari:** Esta ventana permite presentar el propósito del módulo de Sari, un “ver más” para conocer mas a fondo sobre el módulo y a su vez ver un demo sobre su funcionamiento.(ver figura 4.9)



**Figura 4.9** Pantalla diseñada para la presentación del módulo Sari

**Ventana Desarrolladores y Colaboradores:** Esta ventana tiene como objetivo hacer un reconocimiento público a Profesores, Estudiantes, Empleados y demás personas que han participado en la creación y mejoras de cada módulo. Esta dedicación a la automatización de las bibliotecas viene contribuyendo de manera directa al desarrollo y prestigio de nuestra institución, a ellos pues mi humilde contribución al mencionarles en este proyecto de grado.(Ver figura 4.10)



## Desarrolladores y Colaboradores

Los desarrolladores y colaboradores que se listan son las personas que han trabajado por muchos años en la construcción de cada uno de los módulos: Edición y Catalogación, Adquisiciones, Préstamo y recuperación de la información. El profesor NIKOLAUS WALCZUCH comenzó en los años 80 con todo este interesante proceso de automatizar las bibliotecas, es por ello que este sistema integrado de gestión de bibliotecas lleva su nombre. acompañado por profesores, tesisistas y empleados de la Universidad de los Andes y demás personas que han hecho aportes y mejoras a los módulos. A todos ellos muchas gracias.

Nikolaus	Walczuch
Isabel	Besembel
Milagros	Riveros
Domingo	Hernández
Flor	Narciso
Edgar	Chacón
Miguel	Ruíz
Herbert	Hoeger
Pablo	Lischinsky
Gabriel	Lischinsky
Fabiola	Rosales
José	Majarrés
Taniana	Rodríguez
Alcides	Ortega
René	Izarra
Alejandra	Briceño
Manuel	Calistri
Carlos	Araujo
Mary	Otero
Marlene	Sosa
Paolo	Perpetuini
Carlos	Gonzalez
Mercedes	Narciso
Nestor	Marcano
Richard	García
Elizabeth	García

[ir a la página principal](#)

**Figura 4.10** Pantalla diseñada para la presentación de los desarrolladores

**Ventanas para Administración:** Estas ventanas son uso exclusivo del administrador del sistema o la persona autorizada, hace consultas en cuanto a: quien se ha registrado, que modulo ha descargado y si hizo o no sugerencias al sistema.(Ver figuras 4.11 – 4.12 y 4.13)

Principal Descarga Módulos Manuales de Instalación Sugerencias ULA **Administrador**

**Sistema Gestor de Biblioteca Nikolaus Walczuch**  
Bajo software libre

**Administrador**

Descargas  
Usuarios  
Sugerencias

En esta página el administrador tiene acceso a consultar la base de datos, tanto para las descargas, como los usuarios y las sugerencias, y así generar un archivo para el control estadístico.

**Sistema Gestor de Biblioteca Nikolaus Walczuch**  
Bajo software libre

## Administración de Descargas

Institución	Módulo	Fecha/Hora	Nombre	Apellido

ir a la página del Administrador

**Figura 4.11** Descarga de módulos



## Administración de Usuarios

Nombre	Apellido	País	Institución	e-mail

[ir a la página del Administrador](#)

**Figura 4.12** Usuarios Registrados



## Administración de Sugerencias

Nombre	Apellido	Fecha

[ir a la página del Administrador](#)

**Figura 4.13** Sugerencias hechas

## 4.4 Proceso de instalación de los Módulos en mi Máquina:

**4.4.1 Módulo Adquisiciones:** Los pasos iniciales de instalación de este módulo comienza con el descomprimir el directorio `adqui.tgz` en el directorio raíz (/) y como usuario root

1. Las páginas de adquisición se encuentran en `/var/www/ServiAqui/`.
2. Abrir una consola y allí se creó un usuario de postgres con nombre: `tesista`, de la siguiente manera:

```
postgres@maquina:# createuser tesista
```

3. Descomprimir el archivo `BDAquisicionesFinal` en cualquier directorio.
4. Cambiarse al directorio `BDAquisicionesFinal`.
5. Crear una base de datos con nombre `Adquisiciones3` de la siguiente manera: `postgres@maquina:# createdb Adquisiciones3`.
6. Luego entrar a la línea de comando de postgres, con el comando `psql Adquisiciones3`, allí correr todos los scripts que estén en el directorio `BDAquisicionesFinal` con el comando siguiente: `\i` para hacer las respectivas pruebas de este módulo.

Una vez descomprimido el directorio, se corrió el módulo usando el navegador de la siguiente manera:

<http://nombredelservidor/ServiAqui/index.php> en este caso el servidor asignado es: <http://maquina/ServiAqui/index.php>

**4.4.2 Módulo Sari:** Para la instalación de este módulo en su máquina debe entrar como usuario root y luego seguir los siguientes pasos:

Descomprima el archivo `sari.tgz` como usuario root en la raíz (/) de la siguiente manera: `tar xzvf sari.tgz`

Una vez descomprimido el archivo `sari.tgz` en la raíz el modulo ya esta instalado, haga una prueba desde su navegador y coloque:

<http://localhost/html/Sari> el modulo debe abrirse

Verifique los siguientes pasos:

1. Para que funcione sari correctamente las páginas deben estar en: /var/www/html/sari además el cgi-bin.
2. En el caso de Ubuntu (sistema operativo instalado para el proyecto) el archivo cgi-bin del apache2 por defecto se encuentra en: /usr/lib/cgi-bin ; en esta ruta debe estar alojado el directorio **Sari**, contiene el directorio de las páginas dinámicas.
3. Para que el módulo sari funcione correctamente, debe contener en la ruta /usr/lib la siguiente librería: libstdc++-lib6.2-2.so.3

**4.4.3 Módulo Win-Edicla:** Para este módulo en especial, se debe estar como usuario root y en la ruta siguiente:

1.- /home/sidula/bases/demonio, dentro del demonio está el archivo daemonedicla, este archivo hay que levantarlo de la siguiente manera: ./daemonedicla &.

2.- Se necesita la siguiente librería para que corra el módulo:

libstdc++-libc6. 1-1.so.2

3.- En winedicla-reving hay un archivo conf.ini dentro del directorio winedicla, allí se debe colocar la dirección correcta del servidor que tenga la base de datos.

## 4.5 Pruebas y Resultados

En esta última etapa se realiza un recorrido por la página para comprobar su correcto funcionamiento. Se mostrará algunas imágenes para comprobar el funcionamiento.

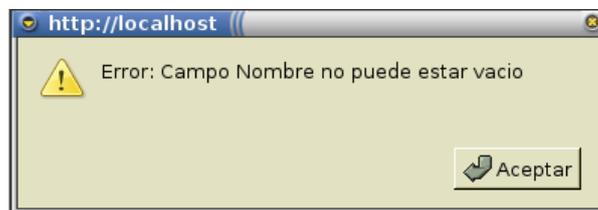
En la figura 4.14 se muestra el mensaje que se le envía al usuario cuando ya se ha registrado correctamente:

## Usuario Carlos Registrado

Se ha enviado un correo a [carlos@ula.ve](mailto:carlos@ula.ve)

**Figura 4.14** Usuario registrado

En la figura 4.15 se muestra el mensaje que se le envía al usuario cuando en el formulario ha dejado un campo vacío



**Figura 4.15** Error campo vacío

En la figura 4.16 se muestra el mensaje que se le envía al usuario cuando ha introducido el usuario o la palabra clave errados

## Clave o password Errado

**Figura 4.16** Error al introducir el password

En la figura 4.17 se muestra la ventana para el control de la estadística, en este caso: los usuarios que se registran en el sistema, mostrando los campos anteriormente descritos. Esta ventana es uso exclusivo del administrador. Se muestra un ejemplo con la inserción de un usuario.



**Sistema Gestor de Biblioteca Nikolaus Walczuch**  
Bajo software libre

Nombre ↓	Apellido	Institucion	País	Correo
Carlos	Moreno	ULA	Venezuela	carlos@ula.ve

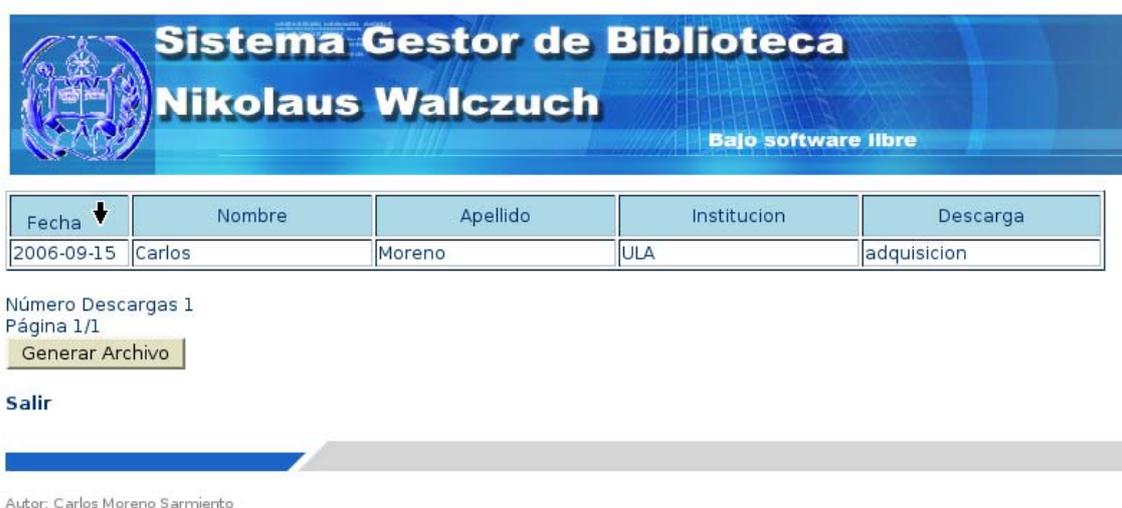
Número Usuarios 1  
Página 1/1  
Generar Archivo

Salir

Autor: Carlos Moreno Sarmiento

**Figura 4.17** Consulta usuarios registrados

En la figura 4.18 se muestra la ventana para el control de la estadística, en este caso: los usuarios que se registran en el sistema y hacen una descarga. Esta ventana es uso exclusivo del administrador. Se muestra un ejemplo con la descarga de un módulo.



**Sistema Gestor de Biblioteca Nikolaus Walczuch**  
Bajo software libre

Fecha ↓	Nombre	Apellido	Institucion	Descarga
2006-09-15	Carlos	Moreno	ULA	adquisicion

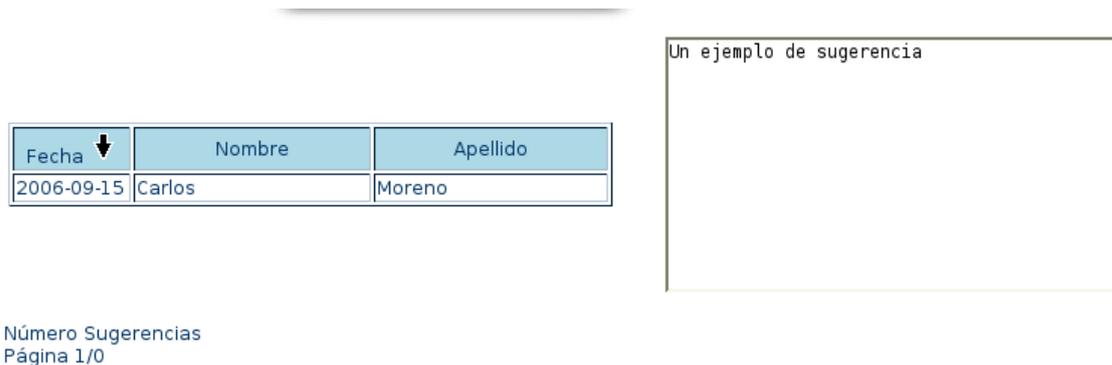
Número Descargas 1  
Página 1/1  
Generar Archivo

Salir

Autor: Carlos Moreno Sarmiento

**Figura 4.18** Consulta descargas

En la figura 4.19 se muestra la ventana para el control de la estadística, en este caso: los usuarios que se registran en el sistema y hacen una sugerencia. Esta ventana es uso exclusivo del administrador. Se muestra un ejemplo con una sugerencia hecha. Para ver la sugerencia en esta ventana basta pasar con el cursor o el puntero por encima del nombre del usuario, así se despliega la sugerencia hecha.



**Figura 4.19** Consulta sugerencias

#### 4.6 Empaquetamientos de los Módulos

Los módulos empaquetados y comprimidos se presentan en la siguiente tabla, con su respectivo nombre:

Módulo	Nombre
Adquisiciones	adqui.tgz
Sari	sari.tgz
Edicla	edicla

**Tabla 4.2** Módulos empaquetados

Lo referente a los empaquetamientos queda definidos estos nombres para los dos primeros módulos, ya que para el edicla es un demonio que debe instalar y usar interfaz Windows. A continuación se describe como y donde debe

descargar cada módulo, pasos que el usuario final debe tener en cuenta para la descarga y conocer donde desempaquetar cada módulo.

#### **4.6.1** Módulo de Adquisiciones:

El empaquetamiento se hizo desde el usuario root:

```
tar czvf adquisi.tgz /var/www/ServiAqui/ var/www/BDAquisicionesFinal/dump.sql
```

el archivo empaquetado está en root y se llama adquisi.tgz, los usuarios deben desempaquetar desde el directorio raíz y como usuario root.

#### **4.6.2** Módulo Sari: De igual manera el empaquetamiento se hizo desde el usuario root:

```
tar czvf sari.tgz /var/www/html/sari/ /usr/lib/cgi-bin/Sari/ /home/sidula/bases/  
/usr/lib/libstdc++-libc6.2-2.so.3
```

#### **4.6.3** Módulo Win-edicla:

El empaquetamiento se hizo desde el usuario root:

```
tar czvf /home/sidula/bases/demonio/ /usr/lib/libstdc++-libc6. 1-1.so.2
```

## **CAPITULO 5**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1 Conclusiones:**

- Permitirá que los usuarios utilicen estos recursos, además de poder contribuir con las mejoras al sistema.
- Permitirá la actualización de las bibliotecas en sus procesos técnicos internos.
- Permitirá disminuir tiempo en la búsqueda de la información en las bibliotecas.
- Permitirá la distribución de este sistema a otras bibliotecas e instituciones del mundo, dando a conocer más la Universidad de Los Andes (ULA), creador del sistema.

#### **5.2 Recomendaciones:**

- El seguir trabajando en cada uno de los módulos, tanto profesores expertos, como tesis de la escuela.
- El abandono o el archivar todos estos módulos, sería de gran pérdida para la Universidad de Los andes, ya que se han invertido valiosos recursos, humanos, de tiempo y recursos económicos.
- El poner esta página Web a disposición del público, ya que podemos esperar mejoras a los módulos desde estos usuarios.

## REFERENCIAS BIBLIOGRAFICAS

[1] **JONAS MOLTILVA** (2001). "Metodología del reloj", Mérida. ULA.

[3] **HERNANDEZ DOMINGO Y RODRIGUEZ TANIANA**, SIRICO un sistema eficiente para la creación de bibliotecas digitales en ambientes distribuido sobre Internet.

[5] **GRADY BOOCH, JAMES RUMBAUGH Y IVAR JACOBSON**. El lenguaje Unificado de Modelado. Pearson Wesley Iberoamericana.

[9] **ELMASRI RAMEZ y NAVATHE AHAMKAT** (1997). Sistemas de Bases de Datos. E. U. A: Adisón Wesley Iberoamericana, S.A.

**GIL FCO, TEJEDOR JORGE, YAGUE AGUSTIN, VILLAVERDE SANTIAGO y GUTIEREZ ABRAHAM** (2001). Creación de sitios Web con PHP 4. España: McGrawHill.

**DAVID M. KROENKE**, Procesamiento de base de datos

Referencias a Internet:

[2] <http://www.serbiula.ve>

[4] <http://www.cidiat.ing.ula.ve>

[6] <http://www.php.net>

[7] <http://postgresql.org>

[8] <http://usuarios.lycos.es/cursosgbd/ud2.htm>

<http://usuarios.lycos.es/oopere/uml.htm>

<http://www.uoc.edu/activitats/docbcn/esp/docbcn.html> (software libre)

<http://perso.wanadoo.es/manel3/chuletas/I-postgresql>